Semantic Mapping using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle

Örebro Studies in Technology 30

Martin Persson

Semantic Mapping using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle

Title: Semantic Mapping using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle

Publisher: Örebro University 2008
www.publications.oru.se

Editor: Maria Alsbjer
maria.alsbjer@oru.se

Printer: Intellecta DocuSys, V Frölunda 04/2008

ISSN 1650-8580
ISBN 978-91-7668-593-8

Abstract

Persson, Martin (2008). Semantic Mapping using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle. Örebro Studies in Technology 30, 170 pp.

In this thesis, semantic mapping is understood to be the process of putting a tag or label on objects or regions in a map. This label should be interpretable by and have a meaning for a human. The use of semantic information has several application areas in mobile robotics. The largest area is in human-robot interaction where the semantics is necessary for a common understanding between robot and human of the operational environment. Other areas include localization through connection of human spatial concepts to particular locations, improving 3D models of indoor and outdoor environments, and model validation.

This thesis investigates the extraction of semantic information for mobile robots in outdoor environments and the use of semantic information to link ground-level occupancy maps and aerial images. The thesis concentrates on three related issues: i) recognition of human spatial concepts in a scene, ii) the ability to incorporate semantic knowledge in a map, and iii) the ability to connect information collected by a mobile robot with information extracted from an aerial image.

The first issue deals with a vision-based virtual sensor for classification of views (images). The images are fed into a set of learned virtual sensors, where each virtual sensor is trained for classification of a particular type of human spatial concept. The virtual sensors are evaluated with images from both ordinary cameras and an omni-directional camera, showing robust properties that can cope with variations such as changing season.

In the second part a probabilistic semantic map is computed based on an occupancy grid map and the output from a virtual sensor. A local semantic map is built around the robot for each position where images have been acquired. This map is a grid map augmented with semantic information in the form of probabilities that the occupied grid cells belong to a particular class. The local

maps are fused into a global probabilistic semantic map covering the area along the trajectory of the mobile robot.

In the third part information extracted from an aerial image is used to improve the mapping process. Region and object boundaries taken from the probabilistic semantic map are used to initialize segmentation of the aerial image. Algorithms for both local segmentation related to the borders and global segmentation of the entire aerial image, exemplified with the two classes ground and buildings, are presented. Ground-level semantic information allows focusing of the segmentation of the aerial image to desired classes and generation of a semantic map that covers a larger area than can be built using only the onboard sensors.

Keywords: semantic mapping, aerial image, mobile robot, supervised learning, semi-supervised learning.

Acknowledgements

If it had not been for Stefan Forslund this thesis would never have been written. When Stefan, my former superior at Saab, gets an idea he believes in, he usually finds a way to see it through. He pursued our management to start financing my Ph.D. studies. I am therefore deeply indebted to you Stefan, for believing in me and making all of this possible.

I would like to thank my two supervisors, Achim Lilienthal and Tom Duckett, for their guidance and encouragement throughout this research project. You both have the valuable experience needed to pinpoint where performed work can be improved in order to reach a higher standard.

Most of the data used in this work have been collected using the mobile robot Tjorven, the Learning Systems Lab's most valuable partner. Of the members of the Learning Systems Lab, I would particularly like to thank Henrik Andreasson, Christoffer Valgren, and Martin Magnusson for helping with data collection and keeping Tjorven up and running. Special thanks to: Henrik, for support with Tjorven and Player, and for reading this thesis; Christoffer, for providing implementations of the flood fill algorithm and the transformation of omni-images to planar images; and Pär Buschka, who knew everything worth knowing about Rasmus, the outdoor mobile robot I first used.

The stay at AASS, Centre of Applied Autonomous Sensor Systems, has been both educational and pleasant. Present and former members of AASS, you'll always be on my mind.

This work could not have been performed without access to aerial images. My appreciation to Jan Eriksson at Örebro Community Planning Office, and Lena Wahlgren at the Karlskoga ditto, for providing the high quality aerial images used in this research project. And thanks to Håkan Wissman for the implementation of the coordinate transformations that connected the GPS positions to the aerial images. The financial support from FMV (Swedish Defence Material Administration), Explora Futurum and Graduate School of Modelling and Simulation is gratefully acknowledged. I would also like to express gratitude to my employer, Saab, for supporting my part-time Ph.D. studies.

Finally, to my beloved family, who has coped with a distracted husband and father for the last years, thanks for all your love and support.

Contents

Part I
Preliminaries

Chapter 1
Introduction

Mobile robots are often unmanned ground vehicles that can be either autonomous, semi-autonomous or teleoperated. The most common way to allow autonomous robots to navigate efficiently is to let the robot use a map as the internal representation of the environment. A lot of research has focused on map building of unknown environments using the mobile robot's onboard sensors. Most of this research has been devoted to robots that operate in planar indoor environments. Outdoor environments are more challenging for the map building process. It cannot any longer be assumed that the ground is flat, the environment contains larger moving objects such as cars and the operating area has a larger scale that put higher demands on both mapping and localization algorithms.

This thesis presents work on how a mobile robot can increase its awareness of the surroundings in an outdoor environment. This is done by building semantic maps, where connected regions in the map are annotated with names of the semantic class that they belong to. In this process a vision-based virtual sensor is used for the classification. It is also shown how semantic information can be used to extract information from aerial images and use this to extend the map beyond the range of the onboard sensors.

There are a wide range of application areas making use of semantic information in mobile robotics. The most obvious area is human robot interaction where a semantic understanding is necessary for a common understanding between human and robot of the operational environment. Other areas include the use of semantics as the link between sensor data collected by a mobile robot and data collected by other means and the use of semantics for execution monitoring, used to find problems in the execution of a plan.

1.1 Motivation

Occupancy maps can be seen as the standard low level map in mobile robot applications. These maps often include three types of areas:

1. Free areas - areas where the robot with a high probability can operate (if the area is large enough).
2. Occupied areas - areas where the robot with a high probability cannot be located. In indoor environments occupied areas typically represent walls and furniture.
3. Unexplored areas - areas where the status is unknown to the robot.

Occupancy maps are used for planning and navigating in an environment. The map can be used for localization and path planning, i.e., the mobile robot can determine how to go from A to B in an optimal way. The robot can also use the map to decide how the area shall be further explored in order to reduce the extent of unknown areas.

A semantic map brings a new dimension of knowledge into the map. With a semantic map the robot not only knows that it is close to an object, but also knows what type of object it faces. With semantic information in the map, the abstraction level of operating the robot can be changed. Instead of ordering the robot to go to a coordinate in the map the robot can be ordered to go to the entrance of a building. To illustrate the benefits of the ability to extract semantic knowledge and of the use of semantic mapping, a number of different situations in outdoor environments are given in the following, where semantic knowledge can support a mobile robot or similar systems:

Follow a route description Humans often use verbal route descriptions when explaining the way for someone that will visit a location for the first time. If the robot has the possibility to understand its surroundings it could follow the same type of descriptions. A route description could for instance be:

1. Follow the road straight ahead.
2. Pass two buildings on the right side.
3. Stop at the road crossing.

Make a route description Conversely to the previous example, a robot that travels from A to B using absolute navigation could also produce route descriptions for humans. Stored information can then be used to automatically produce descriptions for tourists, business travellers, etc.

Localization using GIS When the robot can build maps that not only outline objects, but also labels the object types, navigation using GIS (Geographical Information Systems) such as city maps is facilitated. If the robot can distinguish, for example, buildings from other large objects (trees, lorries and ground formations) the correlation between the building information in the robot's map and in a city map may be established as long as

the initial pose estimation is good enough. For the case where only one building has been found this "good enough" is related to the inter-house distances and for the case where several buildings have been mapped the initial pose estimation can be even less restricted.

Navigation using GPS and aerial image Consider a mobile robot that should go from position A to position B, where the positions are known in global coordinates. If the robot is equipped with a GPS (Global Positioning System) it can navigate from A toward B. What it cannot foresee are possible obstacles in the form of rough terrain, large buildings, etc., and it is therefore not possible to plan the shortest traversable path to B. Now assume that the robot has access to an aerial image and that it has the ability to recognise certain types of objects, such as buildings, trucks and roads, with the onboard sensors. The robot can then build a semantic map of its vicinity, correlate this with estimated buildings and roads in the aerial image and start planning the path to take. As more buildings are detected, the segmentation of the aerial image improves and the final path to the goal can be determined.

Assistance for the visually impaired The technique of a virtual sensor that uses vision to understand objects in the environment could be used in an assistance system for blind people. With a small wearable camera and an audio interface the system can report on objects detected in the environment, e.g.:

1. Bus stop to the left.
2. Approaching a grey building.
3. Entrance straight ahead.

This case clearly indicates the benefit of using high-level (semantic) information, since the alternative where the environment is only described in terms of objects with no labels is less useful.

Search and Surveillance Consider a robot that should be used in an urban area that is restricted for persons to enter and that the robot has no access to any a priori information. Depending on the task the robot needs to understand the environment and be able to detect human spatial concepts that are of interest for an operator. This can, for example, be to search for injured people or to find signs of intruders like broken windows. Extracting information with a vision system the robot can report the locations of different objects and send photos of them back to the operator. This gives the operator the possibility to mark interesting locations in the images for further investigations or to give new commands based on the visual information.

From the above situations three desired "skills" related to semantic information can be noted:

1. The ability to recognise certain types of objects in a scene and by that relating these objects to human spatial concepts,
2. the ability to incorporate semantic knowledge in a map, and
3. the ability to connect information collected by a mobile robot with information extracted in an aerial image.

1.2 Objectives

The main objective of the work presented in this thesis is to propose a framework for semantic mapping in outdoor environments, a framework that can interpret information from vision systems, fuse the information with other sensor modalities and use this to build semantic maps. The performance of the proposed techniques is demonstrated in experiments with data collected by mobile robots in an outdoor environment. The work is structured according to the three "skills" discussed in the previous section. It was decided to use machine learning for the recognition part in order to have a generic system that can adapt to different environments by a training process.

A mobile robot shall by use of onboard sensors and possible additional information include semantic information in a map that is updated by the robot. For the work with this thesis the main information source was selected to be vision sensors. Vision sensors have a number of attractive properties, including:

- They are often available at low cost,
- they are passive, resulting in decreased probability of interference with other sensors,
- they can produce data with rich information (both high resolution and colour information), and
- they can acquire the data quickly.

There are also some drawbacks, especially in comparison to laser range scanners or radar; standard cameras do not allow to measure range directly, indirect range measurement have low accuracy, and standard cameras are sensitive to brightness, mix of direct and indirect light, weather conditions, etc.

Another objective of the work presented in this thesis is to develop algorithms that allow to automatically include information from aerial images in the mapping process. With the growing access to high quality aerial images, e.g., from Google Earth and Microsoft's Virtual Earth, it becomes an attractive

opportunity for mobile systems to use such images in planning and navigation. Extracting accurate information from monocular aerial images is not a trivial task. Usually digital elevation models are needed in order to separate, e.g., buildings from driveways. An alternative method that can replace digital elevation models by combining the aerial image with data from a mobile robot is suggested and evaluated. The objective is to extract information that can be useful in tasks such as planning and exploration.

The work presented in this thesis is concentrated on extraction of semantic information and on semantic map building. It is assumed that techniques for navigation, planning, etc., are available. The experiments were performed using manually controlled mobile robots and the paths were chosen by a human. The evaluated algorithms are implemented in Matlab [The MathWorks] for evaluation and currently work off-line.

1.3 System Overview

The system presented in this thesis consists of three modules that were designed to be applied in a sequential order. The modules can be exchanged or extended separately if new requirements arise or if information can be gathered in alternative ways.

The first module is a virtual sensor for classification of views. In our case the views are images and together with the robot pose and the orientation of the sensor this module points out the directions toward selected human spatial concepts. Two different vision sensors have been used; an ordinary camera mounted on a pan-and-tilt-head (PT-head) and an omni-directional camera giving a 360°-view of the surroundings in one single shot. Each omni-image was transformed to a number of planar images, dividing the 360°-view into smaller portions. The images are fed into learned virtual sensors, where each virtual sensor is trained for classification of a certain type of human spatial concept.

The second module computes a semantic map based on an occupancy grid map and the knowledge about the objects in the environment, in our case the output from *Module 1*, the virtual sensor. A local map is built for each robot position where images have been acquired. The local maps are then fused into a global probabilistic semantic map. These operations assume that the robot is able to determine its pose (position and orientation) in the map.

The third module uses information extracted from an aerial image in the mapping process. Region and object boundaries in the form of line segments taken from the probabilistic semantic map (*Module 2*) are used to initialize local segmentation of the aerial image. An example is given with the class buildings, where wall estimates constitute the object boundaries. These wall estimates are matched with edges found in the aerial image. Segmentation of the aerial image is based on the matched lines. The results from the local segmentation are used to train colour models which are further used for global

segmentation of the aerial image. In this way the robot acquires information about the surroundings it has not yet visited. The global segmentation is exemplified with two classes, buildings and ground.

With these three modules, the three "skills" listed at the end of Section 1.1 are addressed.

1.4 Main Contributions

The main contributions of the work presented in this thesis are:

- Definition and evaluation of a learned virtual sensor based on a generic feature set. Together with the pose from the mobile robot this can be used to point out different human spatial concepts.

  [Publications 6 and 7]

- A method to build probabilistic semantic maps that handles the uncertainty of the classification with the virtual sensor.

  [Publication 5]

- Introduction of ground-based semantic information as an alternative to the use of elevation data in detection of buildings in aerial images.

  [Publications 1, 2, 3, and 4]

- The use of aerial images in mobile robot mapping to extend the view of the onboard sensors to, e.g., be able to "see" around the corner.

  [Publications 1, 2, 3, and 4]

1.5 Thesis Outline

The presentation of the work is divided into two parts where the first part (Chapters 3 - 5) covers ground-based semantic mapping and extraction of semantic information, i.e., *Module 1* and *2*. The second part (Chapters 6 - 8) is based on work that includes aerial images. In detail, the thesis is organized as follows:

Chapter 2 describes the experimental equipment used, consisting of two mobile robots and two handheld digital cameras.

Chapter 3 gives an overview of works that have been published in the area of semantic mapping and works about mobile robot applications in which semantic information is utilized in a number of different ways.

Chapter 4 describes the virtual sensor (*Module 1*). Two classification methods, AdaBoost and Bayes classifier, are compared for diverse sets of images of buildings and nonbuildings. Virtual sensors for windows and trucks are learned and an example where the output from the virtual sensor is combined with the mobile robot pose to point out the direction to buildings is given.

Chapter 5 shows how the information from the virtual sensor can be used to label connected regions in an occupancy grid map and in this way create a probabilistic semantic map (*Module 2*).

Chapter 6 describes systems for automatic detection of buildings in aerial images and specifically points out problems with monocular images.

Chapter 7 presents a method to overcome problems in detection of buildings in monocular aerial images and at the same time to improve the limited sensor range of the mobile robot. It is shown how the probabilistic semantic map described in Chapter 5 can be used to control the segmentation of the aerial image in order to detect buildings (first part of *Module 3*).

Chapter 8 extends the work in Chapter 7 by adding a global segmentation step of the aerial image in order to obtain estimates of both building outlines and driveable areas. With this information exploration in unknown areas can be reduced and path planning facilitated (second part of *Module 3*).

Chapter 9 summarizes the thesis, discusses the limitations of the system and gives proposals for future work.

The appendices contain a list of abbreviations and explanations of the notation used in the thesis (Appendix A), and give details on some of the implementations (Appendix B).

1.6 Publications

A large extent of the work presented in this thesis has been previously reported in the following publications:

1. Martin Persson, Tom Duckett and Achim Lilienthal, "Fusion of Aerial Images and Sensor Data from a Ground Vehicle for Improved Semantic Mapping", accepted for publication in *Robotics and Autonomous Systems*, Elsevier, 2008

2. Martin Persson, Tom Duckett and Achim Lilienthal, "Improved Mapping and Image Segmentation by Using Semantic Information to Link Aerial Images and Ground-Level Information", In *Recent Progress in Robotics; Viable Robotic Service to Human*, Springer-Verlag, Lecture Notes in Control and Information Sciences, Vol. 370, December 2007, pp. 157–169

3. Martin Persson, Tom Duckett and Achim Lilienthal, "Fusion of Aerial Images and Sensor Data from a Ground Vehicle for Improved Semantic Mapping", In *IROS 2007 Workshop: From Sensors to Human Spatial Concepts*, November 2, 2007, San Diego, USA, pp. 17–24

4. Martin Persson, Tom Duckett and Achim Lilienthal, "Improved Mapping and Image Segmentation by Using Semantic Information to Link Aerial Images and Ground-Level Information", In *Proceedings of the 13th International Conference on Advanced Robotics (ICAR)*, August 21–24, 2007, Jeju, Korea, pp. 924–929

5. Martin Persson, Tom Duckett, Christoffer Valgren and Achim Lilienthal, "Probabilistic Semantic Mapping with a Virtual Sensor for Building/Nature Detection", In *Proceedings of the 7th IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, June 21–24, 2007, Jacksonville, FL, USA, pp. 236–242

6. Martin Persson, Tom Duckett and Achim Lilienthal, "Virtual Sensor for Human Concepts – Building Detection by an Outdoor Mobile Robot", In *Robotics and Autonomous Systems*, Elsevier, 55:5, May 31, 2007, pp. 383–390

7. Martin Persson, Tom Duckett and Achim Lilienthal, "Virtual Sensor for Human Concepts – Building Detection by an Outdoor Mobile Robot", In *IROS 2006 Workshop: From Sensors to Human Spatial Concepts - Geometric Approaches and Appearance-Based Approaches*, October 10, 2006, Beijing, China, pp. 21-26

8. Martin Persson and Tom Duckett, "Automatic Building Detection for Mobile Robot Mapping", In *Book of Abstracts of Third Swedish Workshop on Autonomous Robotics*, FOI 2005, Stockholm, September 1–2, 2005, pp. 36–37

9. Martin Persson, Mats Sandvall and Tom Duckett, "Automatic Building Detection from Aerial Images for Mobile Robot Mapping", In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Espoo, Finland, June 27–30, 2005, pp. 273–278

Chapter 2
Experimental Equipment

This chapter contains descriptions of the equipment used in the experiments presented in Chapters 4, 5, 7, and 8. First, navigation sensors for mobile robots are discussed, followed by descriptions of the robots, Tjorven and Rasmus. Then, the two handheld cameras used to take images for training and evaluation of the virtual sensor are introduced, and details about the aerial images used are presented.

2.1 Navigation Sensors for Mobile Robots

During the collection of data with our mobile robots, the robots were manually controlled. Thus, the navigation sensors onboard the robots were not needed in this phase. However, when the data were processed, localization was important. It was used for building the occupancy grid maps and for registration of the position and orientation of the robot.

GPS

In the Global Positioning System (GPS) triangulation of signals sent from satellites with known positions and at known times is used to calculate positions in a global coordinate system. GPS system errors, such as orbit, timing and atmospheric errors, limit the accuracy that can be achieved to approximately 10-15 metres for a standard receiver [El-Rabbany, 2002].

GPS receivers placed in the vicinity of each other often show the same errors. This fact is exploited in differential GPS (DGPS). A GPS receiver is then placed at a known location and the current error can be calculated. This is then transmitted to mobile GPS receivers via radio, and the error can in this way be significantly reduced. The method gives an accuracy of 1-5 m at distances of up to a few hundred kilometres.

Other methods for improving navigation accuracy include RTK GPS (real-time kinematics GPS), and differential corrections offered as commercial services; these are used, e.g., by agriculture vehicles [García-Alegre et al., 2001].

A quality measure of the position estimate is indirectly available from the GPS receiver. The number of satellites used in the calculation is one measure. At minimum three are needed for a 2D-position (latitude/longitude), and four are needed to also calculate an altitude value. Depending on the relative positions of the satellites, the accuracy may vary considerably. This is reported in three parameters: position, horizontal and vertical dilution of precision (PDOP, HDOP, and VDOP).

Odometry

Odometry consists of proprioceptive (self-measurement) sensors that measure the movement of robot wheels using wheel encoders. The encoder information can be used to compute a position estimate. The error in position estimates from this type of sensor accumulates as the robot moves and estimates are usually useless for long runs. Errors are due to factors such as slippery surfaces and unequal wheel diameters.

IMU, Compass and Inclinometers

Inertial Measurement Unit (IMU) compass, and inclinometers are complementary navigation sensors. An IMU usually consists of three accelerometers, measuring the unit's acceleration in an orthogonal frame, and angular rate gyros that measure the rotation rates around the same axes. From these a relative 6D pose can be calculated. A compass delivers absolute values of the robot heading, and inclinometers measure pitch and roll angles of the robot.

Integrated Navigation

The sensors described above are often used in integrated navigation systems due to their complementary strengths and weaknesses. GPS is the only sensor that directly gives a global position. Due to the properties of GPS, it is usually combined with sensors that are accurate for short distance motion or do not drift over time. Odometry needs calibration but it can be quite accurate over short distances, and it is not affected by time. Inertial measurements on the other hand suffer from drift directly related to the integration time. Combined, these sensors can constitute a navigation system, giving geo-referenced positions with high accuracy as long as the GPS delivers reliable position estimates.

GPS is best suited for open areas or in the air, where it continuously has a number of GPS-satellites in view. In urban terrain, where satellites may be shadowed by buildings, problems due to reflections or multi-path signals arise, especially close to large objects. This has been noted by, e.g., Ohno *et al.* in their work that addressed fusion of DGPS and odometry [Ohno et al., 2004]. The problem with multi-path signals is impossible to detect using the usual quality measures, such as the number of satellites in sight or the position dilution

of precision, and can therefore introduce severe position errors. One way to overcome the problem is to use complementary sensors, e.g., laser range scanners [Kim et al., 2007], since these are suitable for navigation in urban regions. Still, the system needs a correct initial position in order to be able to detect the presence of multi-path signals.

2.2 Mobile Robot Tjorven

This section describes the mobile robot Tjorven, used in most of the experiments presented in this thesis. Tjorven is a Pioneer P3-AT from ActivMedia, equipped with differential GPS, a laser range scanner, cameras and odometry. The robot is equipped with two different types of cameras; an ordinary camera mounted on a pan-tilt-head together with the laser, and an omni-directional digital camera. The onboard computer runs Player[1], tailored for the used sensor configuration, which handles data collection. The robot is depicted in Figure 2.1 with markings of the used sensors and equipment.

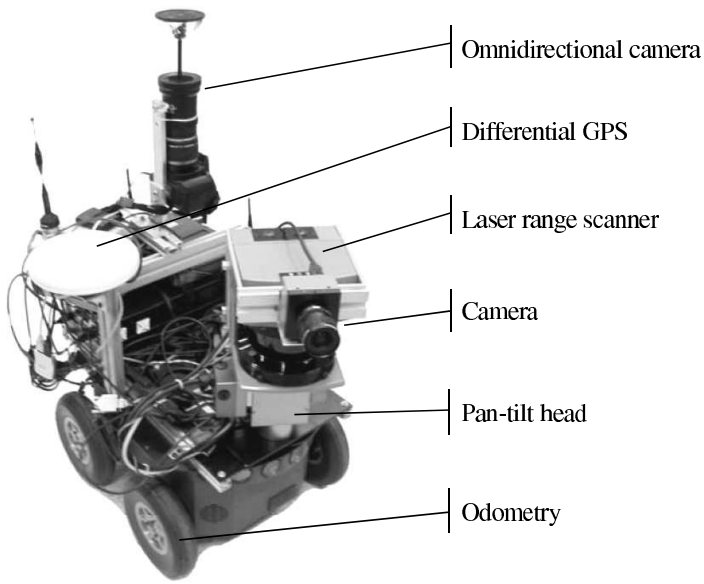


Figure 2.1: The mobile robot Tjorven.

[1]Player is a robot server released under the GNU General Public License. Information about the project can be found at http://playerstage.sourceforge.net/.

Laser range scanner

The laser range scanner is a SICK[2] LMS 200 mounted on the pan-tilt-head. It has a maximum scanning range of 80 m, a field of view of $180°$, a selectable angular resolution of $0.25°$, $0.5°$, or $1°$ ($1°$ was used in the experiments), and a range resolution of 10 mm. A complete scan takes in the order of 10 ms, and scans are usually stored at 20 Hz in our experiments.

GPS

The used differential GPS from NovAtel[3], a ProPak-G2*Plus*, consists of one GPS receiver, which is called the base station, placed at a fixed position and one portable GPS receiver, called the rover station, which is mounted on the robot. These two GPS receivers are connected via a wireless serial modem. The imprecision of the system is around 0.2 m (standard deviation) in good conditions. GPS data are stored at 1 Hz.

Odometry

The odometry measures the rotation of one wheel on the left side of the robot and one wheel on the right side of the robot. Using this it captures both translational and rotational motion. Measurements from odometry are stored at 10 Hz.

Pan-Tilt Head

The rotary pan-tilt head, a PowerCube 70 from Amtec[4], allows for rotational motion around two axes. In the horizontal plane it can rotate about three quarters of a revolution, limited by the physical configuration of the robot components, and the head can tilt approximately $\pm 60°$.

Planar Camera

The planar camera is mounted on the laser range scanner giving it the same movability as the laser. The camera is a DFK 41F02 manufactured by ImagingSource[5]. It is a FireWire camera with a colour CCD sensor with 1280×960 pixel resolution.

Omni-directional Camera

The omni-directional camera gives a $360°$ view of the surroundings in one single shot. The camera itself is a standard consumer-grade SLR digital camera, 8

[2] www.sick.com
[3] www.novatel.com
[4] Amtec-robotics is now integrated in Schunk GmbH, www.schunk.com
[5] www.theimagingsource.com

megapixel Canon EOS350D[6]. On top of the lens, a curved mirror from 0-360.com[7] is mounted.

2.3 Mobile Robot Rasmus

Rasmus is an outdoor mobile robot, an ATRV JR from iRobot. The robot is equipped with a laser scanner, a stereo vision sensor and navigation sensors.

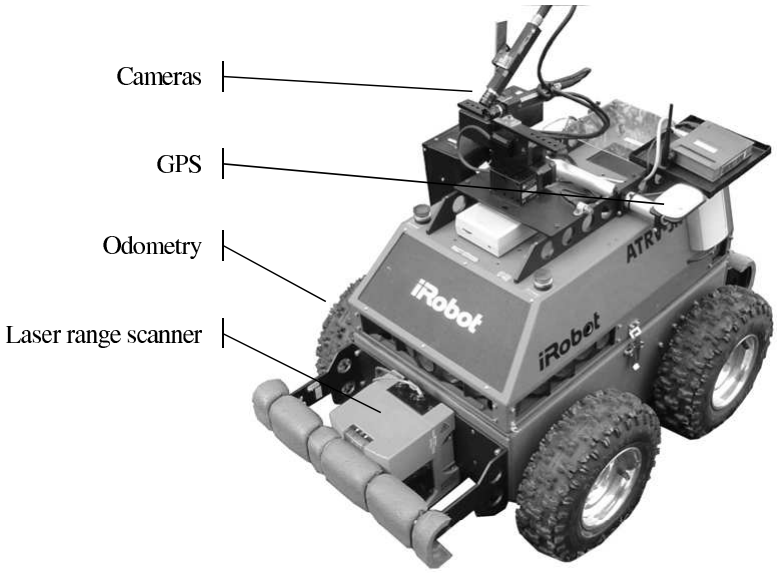


Figure 2.2: The mobile robot Rasmus.

Camera

The cameras on Rasmus are analogue camera modules XC-999, manufactured by Sony[8]. They have a 1/2 inch CCD colour sensor with 768×494 pixel resolution.

[6] www.canon.com
[7] www.0-360.com
[8] www.sony.com

Laser range scanner

The mobile robot is equipped with a fixed 2D SICK laser range scanner of type LMS 200; the specifications are the same as the ones for the laser range scanner on Tjorven, see Section 2.2.

GPS-receiver

The GPS-receiver on Rasmus is an Ashtech G12 GPS[9]. The update rate for position computation is selectable between 10 Hz and 20 Hz [Ashtech, 2000]. At 20 Hz the calculation is limited to eight satellites.

Inertial Measurement Unit

The inertial measurement unit is a Crossbow[10] IMU400CA-200. It consists of 3 accelerometers and 3 angular rate sensors; see the manual [Crossbow, 2001] for further information.

Compass

The compass unit is a KVH C100. It measures heading with a resolution of 0.1°, and it has an update rate of 10 Hz [KVH, 1998].

Odometry

The robot uses two wheel encoders that give the rotation of one left and one right wheel. The output is presented as a linear value representing forward distance and a rotation value representing the robot rotation around its vertical axis. The resolution is below 0.1 mm.

2.4 Handheld Cameras

Two handheld digital cameras have been used to collect images for training the virtual sensor. The first is a 5 megapixel Sony DSC-P92 digital camera with autofocus. It has an optical zoom of 38 to 114 mm (measured as for 35 mm film photography).

The second camera is built into a SonyEricsson K750i mobile phone. This camera is also equipped with autofocus, and the image size is 2 megapixels. The fixed focal length is 4.8 mm (equivalent to 40 mm when measured as for 35 mm film photography).

[9] www.magellangps.com
[10] www.xbow.com

Both cameras store images in JPEG-format, and the finest settings (highest resolution and quality) have been used for the collection of images. The cameras are depicted in Figure 2.3.

Figure 2.3: The used digital cameras. The one on the right is the Sony DSC-P92, and the one on the left is the SonyEricsson K750i.

2.5 Aerial Images

The aerial images used in this project are colour images taken from altitudes of 2300 m to 4600 m. The images were taken during summer, in clear weather. The pixel size is 0.5 m or lower (images with higher resolution were converted to 0.5 m). The images are stored in uncompressed TIFF-format.

Part II

Ground-Based Semantic Mapping

Chapter 3
Semantic Mapping

This chapter presents the state-of-the-art in semantic mapping for mobile robots. The focus is on outdoor semantic mapping, even though it is not restricted to outdoor environments. It was, however, difficult to find relevant literature in this subject since the number of publications on semantic mapping is still quite low. Most of the relevant publications relate to mapping of indoor environments and only a few consider the problem that the robot itself extracts the semantic labels for the map. The content of this chapter is therefore broader than the topic of this thesis in order to capture immediate works that can have an influence on research in semantic mapping.

In this thesis, semantic mapping is understood to be the process of putting a tag or label on objects or regions in a map. This label should be interpretable by and have a meaning for a human. In mobile robotics this can also be described as a transformation of sensor readings to a human spatial concept. Alternative interpretations of semantics in this area exist. For instance semantics can, when extracted by a robot, have a meaning for the robot but be hard for humans to interpret [Téllez and Angulo, 2007].

This chapter is organised as follows. Section 3.1 gives a short overview on different types of maps used in mobile robotics. Extraction of semantic information in indoor environments is discussed in Section 3.2. This includes object detection, space classification and systems where semantic maps form a layer in a hierarchical representation of the environment. Examples from outdoor mapping are presented in Section 3.3. As a motivation for using semantic information, Section 3.4 gives examples of how semantic information is used in different applications. The chapter concludes with a summary in Section 3.5.

3.1 Mobile Robot Mapping

A map is a representation of an area, a restricted part of the world. Maps used in mobile robotics can be divided into three groups; metric maps, topological maps and hybrid maps, where the latter are a combination of the first two

types. To give a short overview, this section briefly presents these maps. For a more comprehensive survey on mapping for mobile robots see e.g. [Thrun, 2002], and for hybrid maps see [Buschka, 2006].

3.1.1 Metric Maps

A metric map is a map where distances can be measured, distances that relate to the real world. Metric maps build by a mobile robot can be divided into grid maps and feature-based maps [Jensfelt, 2001].

Grid Map Grid maps are probably the most common environment representation used for indoor mobile robots. The value of a grid cell in a metric grid map represents a measure of occupancy of that specific cell and gives information whether the cell has been explored or not [Moravec and Elfes, 1985]. A grid map containing metric information is well suited for path planning. Static objects that are observed several times are usually given higher values than dynamic objects that appear at different locations. The main drawbacks of grid maps are that they are space consuming and that they provide a poor interface to most symbolic problem solvers [Thrun, 1998].

Feature-based Map Feature-based maps represent features or landmarks that can be distinguished by the mobile robot. Examples of commonly used features are edges, planes and corners [Chong and Kleeman, 1997]. Feature-based maps are not used in the work presented in this thesis, but some of the referred works presented in, e.g., Section 3.2 use this type of map.

Topographic Map In a topographic map the elevation of the Earth's surface is shown by contour lines. This type of map also often includes symbols that represent features like different types of terrain, cultural landscapes and urban areas with streets and buildings. Topographic maps are seldom used directly in mobile robotics but they can be used in the creation of schematic maps [Freksa et al., 2000]. The schematic map can be used both for path planning of a mobile robot and as the reference model of the environment during navigation by the mobile robot.

3.1.2 Topological Maps

Topological maps are represented as graphs with nodes and arcs (also called edges) where the nodes represent distinct spatial locations and the arcs describe how the nodes are connected. This allows efficient planning and typically results in lower computational and memory requirements [Thrun, 1998]. Topological maps can be built from metric maps where the nodes can be found by use of, e.g., Voronoi diagrams [v. Zwynsvoorde et al., 2000].

3.1.3 Hybrid Maps

Hybrid maps are a solution to overcome the shortcomings from using only one specific type of map by combining different type of maps. Most common is the combination of metric and topological maps. In the context of this thesis, the combination of metric maps and semantic information is more relevant.

3.2 Indoor Semantic Mapping

In this section works related to indoor semantic mapping are presented. First, works on object labelling are reported. Second, scientific work on how to classify different areas is described, e.g., where space in an indoor environment is labelled as "kitchen" and "office". Finally, hierarchical map constructions that include semantic maps are presented.

3.2.1 Object Labelling

Finding doors and gateways is essential for mobile robots in order to navigate in indoor environments and consequently the largest group of publications addresses door and gateway recognition. In the following, short descriptions of a selection of works where objects are found and classified are given.

In Anguelov's work [Anguelov et al., 2002] movable objects, detected from mapping the environment at different times, are learned. Similar objects can then be detected by the mobile robot in new environments without seeing the object move. The approach is model-based, where the objects are detected using a 2D laser range scanner. This gives the contour and size of the object, which in turn is compared with templates. To detect a correct contour the objects need to be separated from other objects. In the experiments a small number of objects (a sofa, a box and two robots) are used.

Another type of object that often moves is a door. A door can be in a state from closed to fully open. This fact has been explored in order to detect doors [Anguelov et al., 2004]. The authors assume rectilinear walls and perform consecutive mappings of a corridor using a laser range scanner. The map building process detects when an already mapped door has moved. This door is then used to train a model of the door colour as seen by an omni-directional camera. The vision system uses the colour model to find more doors of the same colour (only one colour is modelled at a time).

Another step toward semantic representations of environments is taken by Limketkai *et al*. They present a method to classify 2D laser range finder readings into three classes: *walls*, *doors*, and *others* [Limketkai et al., 2005]. A Markov chain Monte Carlo method is used to learn model parameters and the results are metric maps with object labels. The objects are aggregated from primitive line segments. A wall can, for example, be a number of aligned lines. The doors are assumed to be indented and the size of the indentation is learned.

Indoor environments often contain planar surfaces that are parallel or orthogonal with respect to each other. Extracting planes from 3D laser range data has been used to achieve semantic scene interpretations of indoor environments as floors, walls, roofs, and open doors [Nüchter et al., 2003, Weingarten and Siegwart, 2006]. Nüchter *et al.* use a semantic net with relationships such as parallel, orthogonal, under etc. The planes are classified using these relationships, for example `floor is parallel to roof` and `floor is under roof`. The semantic information is used to merge neighboring planes, which in turn leads to refined 3D models with reduced jitter in floors, walls and ceilings. In a similar work the semantic information was used to improve scan matching using a fast variant of Iterative Closest Point (ICP) [Besl and McKay, 1992] by performing individual matching of the different classes, e.g., points belonging to `floor` in one scan are matched with `floor`-points in the following scan [Nüchter et al., 2005].

Beeson *et al.* use extended Voronoi graphs to autonomously detect *places* in a global metric map [Beeson et al., 2005]. Their work is based on detection of gateways and path fragments in the map. These two concepts are used to detect places. According to their definition, a place is found when there are not exactly two gateways and one path. For instance, a dead end is a place since it has one gateway and one path, and an intersection is also a place since it has more than one path. This type of place detection can be used in topological map building.

In vision based SLAM (Simultaneous Localization and Mapping) different kinds of visual landmarks are used. A semantic approach is to learn and label objects by their appearance using SIFT (Scale-Invariant Feature Transform) features. In order to handle different views of the landmarks, 3D-object models can be built based on a number of views of the object [Jeong et al., 2006]. Integrating the semantic map in SLAM eliminates the need for a specific anchoring technique that connects positions in the map (landmarks) and their associated semantics. Instead, the SIFT features directly constitute the link between the learned objects and objects registered as landmarks in the semantic map. In the work presented by Jeong *et al.* experiments are performed with 5 different objects that are manually labelled and pre-stored in an object feature database.

Ekvall *et al.* also use SIFT features to recognise objects in combination with SLAM [Ekvall et al., 2006]. Training is performed by showing the interesting objects to a mobile robot equipped with a vision system. An object is extract by background subtraction. Semantic information in the form of Receptive Field Cooccurence Histograms (RFCH) and SIFT features of the object are extracted. RFCH is used to locate potential objects and SIFT features are used for verification of the object in zoomed-in images. When the robot performs SLAM and detects an object, a reference to the object is placed in the map and in this way the robot can return and find a requested object. If the object has moved, a search for the object is performed.

3.2.2 Space Labelling

The previous subsection gave examples of object labelling and locating objects in a map. In this subsection, the focus is on classification of areas in the map. Several works on classification of indoor space have been reported. They can be divided into two classes. The first type that is reported here distinguishes between gateways, rooms, and corridors. The second type of methods also classify what type of room the robot has entered, e.g., "kitchen" or "living room".

An example of the first type is the virtual sensor for detection of room and corridor transitions presented in [Buschka and Saffiotti, 2002]. The virtual sensor makes use of sonar sensors and both indicates the transitions between different rooms and calculates a set of parameters characterising the individual rooms. Each room can be seen as a node in a topological structure. The set of parameters includes the width and length of the room and is calculated using the central 2^{nd} order moments resulting in a virtual sensor that is relatively stable to changes of the furniture in the room.

When service robots act in a domestic environment it is important that the definition of regions follow a human representation. In human augmented mapping [Topp and Christensen, 2006] a person guides a mobile robot in a domestic environment, and gives the robot information about the different locations. During this guided home tour the robot learns about the environment from the user or from several users. A hierarchical representation of the environment is created and segmented using the following concepts:

- Objects – things that can be manipulated.
- Locations – areas from where objects can be observed or manipulated, often smaller than a room.
- Regions – contain one or several locations.
- Floor – connects a number of regions with the same height in order to be able to distinguish between similar room configurations at different levels.

The guidance procedure includes dialog between the robot and the user where the robot can ask questions in order to remove ambiguous information.

Another robot system that learns places in a home environment is BIRON [Spexard et al., 2006]. BIRON uses an integration of spoken dialog and visual localization to learn different rooms in an apartment.

Mozos *et al.* semantically label indoor environments as corridors, rooms, doorways, etc. Features are extracted from range data collected with 180-degree laser range scanners [Mozos et al., 2005, Mozos, 2004]. These features are the input to a classifier learned using the AdaBoost algorithm. The features are based on 360-degrees scans and to obtain them two configurations

have been used. The first configuration uses two 180-degree laser range scanners and the second configuration uses one 180-degree laser range scanner and the remaining 180-degrees are simulated from a map of the environment. In [Rottmann et al., 2005] additional features extracted from a vision sensor are used. The use of visual features is limited to recognition of a few objects (e.g. monitor, coffee machine, and faces), due to its complexity. Nevertheless, in the environments where the system was evaluated, it was demonstrated that using these visual features made it easier to classify a room as either a seminar room, an office or a kitchen. The method has been tested in different office environments.

Friedman *et al.* extract the topological structure of indoor environments via place labels ("room", "doorway", "hallway", and "junction") [Friedman et al., 2007]. A map is built using measurements from a laser range scanner and SLAM. Similar features to those defined by Mozos are extracted at the nodes of a Voronoi graph defined in the map. Voronoi random fields, a technique to extract the topological structure of indoor environments, are introduced. Points on the Voronoi graph are labelled by converting the graph into a conditional random field [Lafferty et al., 2001] and the feature set is extended with connectivity features extracted from the Voronoi graph. With these new features it is possible to differentiate between actual doorways and narrow passages caused by furniture, since it is more likely that short loops are found around furniture than through doorways. Adding these features to a classifier learned with AdaBoost improved the resulting topological map. Further improvement was reported when the Voronoi random fields were used together with the best weak classifiers found by AdaBoost.

A purely visual approach to the classification of indoor environments is presented by Pirri [Pirri, 2004]. The method makes use of a texture database obtained from a large number of images of indoor environments. The textures are processed with a wavelet transform to describe their characteristics. Textures of furniture and wall materials are stored in the database and combined with a statistical memory that includes probability distributions of the likelihood of rooms with respect to the furniture.

3.2.3 Hierarchies for Semantic Mapping

Approaches that use semantic mapping and are intended to handle navigation at different scales and complexity often present maps in the form of a hierarchy. Different levels of refinement are used with at least one layer of semantic information.

The concept of the Spatial Semantic Hierarchy (SSH) evolved during the 1990's [Kuipers, 2000]. SSH is inspired by properties of cognitive mapping, the principles that humans use to store spatial knowledge of large-scale areas. Spatial knowledge describes environments and is essential for getting from one

place to another. SSH consists of several interacting representations of knowledge about large-scale spaces that are divided into five levels:

- The *sensory level* is the interface to the sensor systems such as vision and laser with the focus to handle motion and exploration.
- The *control level* uses continuous control laws as world descriptors. The level can create and make use of local geometric maps.
- The *causal level* contains information similar to what can be obtained from route directions and is essential in SSH.
- The *topological level* includes an ontology of places, paths, connectivity etc. intended for planning.
- The *metrical level* contains a global metric map. This map is not an essential part of SSH.

The metrical level can be used in path planning or to distinguish between places that appear to be identical in the other levels, but navigation and exploration can still be possible without this information.

Galindo *et al.* present a multi-hierarchical semantic map for mobile robots [Galindo et al., 2005]. The map consists of two hierarchies; the *spatial* hierarchy and the *conceptual* hierarchy. The *spatial* hierarchy contains information gathered by the robot sensors. The information is stored in three levels; local grid maps, a topological map and an abstract node that represents the whole spatial environment of the robot. The *conceptual* hierarchy models the relationship between concepts, where concepts are categories (objects and rooms) and instances (e.g. "*room-C*" and "*sofa-1*"). The two hierarchies are integrated to allow the robot to perform tasks like "*go to the living room*". This multi-hierarchical semantic map is further developed in [Galindo et al., 2007]. The two hierarchies resemble the previous ones, but are here named *spatio-symbolic* hierarchy and the *semantic* hierarchy. The semantic map is used to discard elements of the domain that should not be considered in the planning phase in order to speed up the planning process. For instance, if the robot should go from the living room to the kitchen and get a fork in a drawer it should not consider objects in the bath room.

Mozos *et al.* present a complete system for a service robot using representations of spatial and functional properties in a single hierarchical semantic map [Mozos et al., 2007]. The hierarchy is composed of four layers; the metric map, the navigation map, the topological map and the conceptual map. The navigation map is a graph with nodes that are placed within a maximum distance of each other and the map is used for planning and autonomous navigation. It is similar to the topological map but represents the environment in more detail. The conceptual map contains descriptions of concepts and their relations in the form of "*is-a*" and "*has-a*", e.g., "*LivingRoom is-a Room*" or "*LivingRoom*

hasObject LivingRoomObj" and "*TVSet is-a LivingRoomObj*". The system includes speech synthesis and speech recognition for operation, which is used in turn by human augmented mapping to learn the places in the topological map. Using semantic information is motivated by the intended use of the robot to interact with people that are not trained robot operators.

NavSpace [Ross et al., 2006] is a stratified spatial representation that includes lower tiers for navigation and localization, and upper tiers for human-robot interaction. It was developed to enable navigation of a wheelchair using dialogs with the human. These dialogs can handle concepts such as "*left of*" and "*beside*", place labels (e.g., "*kitchen*"), action descriptions (e.g., "*turn*") and quantitative terms (e.g., "*10 metres*").

It can be noted that in the works described above, the conceptual relationships are hand-coded and not learned by the robot itself.

3.3 Outdoor Semantic Mapping

The number of publications related to outdoor semantic mapping is lower than the works on indoor semantic mapping that were reported above.

Wolf and Sukhatme [Wolf and Sukhatme, 2007] describe two mapping approaches that create outdoor semantic maps from laser range scanner readings. Two techniques for supervised learning were used: Hidden Markov Models (HMM) and Support Vector Machines (SVM). The first semantic map is based on the activity caused by passing objects of different sizes. Using this information, the area is classified as either road or sidewalk. The resulting map is stored in a two-dimensional grid of symmetric cells. Two robots are placed on each side of the road with overlapping fields of view in order to decrease the influence of occlusion. During data collection the positions of the robots were fixed and known. Four properties were extracted from the laser data and stored in the map: activity, occupancy, average object size and maximum object size. The authors also included one more class, stationary objects, in addition to road and sidewalk and then used a multi-class SVM for the classification. The second type of semantic map classifies ground into two classes; navigable and non-navigable. The classification is based on the roughness of the terrain measured by the laser and is intended to be used for path planning.

Triebel *et al.* have developed a mapping technique for outdoor environments, called multi-level surface maps, that can handle structures like bridges [Triebel et al., 2006]. Multiple surfaces can be stored in each grid cell and by analysing the neighbouring cells, classification of the terrain into traversable, non-traversable and vertical surfaces is performed.

Closely related work to these terrain mapping approaches concerns detection of drivable areas for mobile robots using vision [Dahlkamp et al., 2006, Guo et al., 2006, Song et al., 2006]. These works do not primarily build semantic maps but they use semantic information for road localization in navigation.

The work performed by Torralba *et al.* delivers the most extensive semantic mapping system found in the literature. Place recognition is performed in both indoor and outdoor environments with the same system [Torralba et al., 2003]. The system identifies locations (e.g. *office 610*), categorises new environments ("office", "corridor", "street", etc) and performs object recognition using the knowledge of the location as extra information. Global image features based on wavelet image decomposition of monochrome images are used and Principal Components Analysis (PCA) reduces the dimensionality to 80 principal components. The presented system recognises over 60 locations and 20 different objects, which is a high number compared with many other reported systems. For training and evaluation, a mobile system consisting of a helmet mounted web camera with resolution 120×160 pixels is used. The system is claimed to be robust to a number of difficulties, such as motion blur, saturation and low contrast.

3.3.1 3D Modelling of Urban Environments

Several research projects directed toward automatic modelling of outdoor environments, especially urban environments, have been presented in the last decade. Even though these projects do not explicitly use semantic information, they need to be able to classify data in order to remove data belonging to classes that should not be included in the final model. From our perspective, these types of projects are also interesting as references for building semantic grid maps as described in Chapter 5.

A project for rapid urban modelling, with the aim to automatically construct 3D walk-through models of city centres without objects such as pedestrians, cars and vegetation, is presented in [Früh and Zakhor, 2003]. The system uses laser range scanners and cameras both on ground and in air. A Digital Elevation Model (DEM) is constructed from an airborne laser range scanner mounted on an airplane and overview images are captured. The experimental set-up used for the data acquisition on the ground consists of a digital colour camera and two 2D-laser range scanners mounted so that one measures a horizontal plane and the other measures a vertical plane [Früh and Zakhor, 2001]. The equipment is mounted at 3.6 meters height on a truck that drives along the roads and the collection of data is performed for one side of the road at a time. The horizontal scanner is used for position estimation and the vertical scanner captures the shape of the buildings. Images from the digital camera are used as texture on the 3D-models built from the scanned data. Turns of the truck cause problems and data affected by this are ignored [Früh and Zakhor, 2002]. Vegetation and pedestrians occlude the facades and are therefore removed from the model using semantic segmentation of the data. The scans are divided into a background part including buildings and ground, and a foreground part that should be removed. Removing the foreground will in turn leave holes in the measurements that need to be filled. The missing spatial information is recon-

structed and the images from the most direct views for the background are used to fill in the holes. The application has problems with, e.g., vegetation and is therefore not suitable for residential areas.

Another system for urban modelling is AVENUE [Allen et al., 2001]. The main goal with AVENUE is to automate site modelling of urban environments. The system consists of three components: a 3D modelling system, a planning system for deciding where to take the next view and a mobile robot for acquiring data. Range sensing (CYRAX 2400 3D laser range scanner) is used to provide dense geometric information, which are then registered and fused with images to provide photometric information. The planning phase, *Next-Best-View*, makes sure that the new scanning includes object surfaces not yet modelled. The navigation system of the mobile robot uses odometry and DGPS [Georgiev and Allen, 2004]. Visual localization is performed when the GPS is shadowed. Using coarse knowledge of the position based on previous sensor readings, the robot knows in what direction to search for building models and matches the corresponding model with the current view in order to accurately determine its pose.

An additional project working on 3D-mapping is presented in [Howard et al., 2004]. A large area (one square kilometre) is mapped with a two-wheeled mobile robot (Segway RMP, Robotic Mobility Platform) equipped with both a vertical and a horizontal laser range scanner. The vertical scanner is directed upwards giving readings both to the right and to the left of the robot. Assumptions made are that the altitude is constant and that the environment is partially structured. Two levels of localization are used. The first is the fine-scale localization that uses the horizontal laser range scanner, roll and pitch data and the odometry. This gives a detailed localization with a drift. The second navigation system is the coarse-scale localization. This uses either GPS, good in open areas, or Monte Carlo Localization (MCL) that is good close to buildings. The MCL requires a prior map that can be extracted from an aerial or satellite image. To combine the coarse and fine localization, feature-based fitting of sub-maps is used.

A fourth project with its main focus on the environment close to roads is presented in [Abuhadrous et al., 2004]. A 2D laser range scanner with 270° scanning angle is mounted on the backside of a car. Histograms are used for identification of objects along a road. The system separates three object types: roads, building facades and trees, and illustrates these using simple 3D models.

These four projects (summarized in Table 3.1) all represent semantic information even though it is not explicitly mentioned. Früh removes vegetation and pedestrians from the model of ground and buildings. In AVENUE buildings are used for localization. Other examples are the use of building outlines extracted from aerial images [Howard et al., 2004] and classification of trees and buildings in laser range point clouds [Abuhadrous et al., 2004].

References	Description	Sensors & Navigation
[Früh and Zakhor, 2001, 2002, 2003]	3D-modelling of buildings in urban areas. Occluding objects are removed.	2 cross-mounted 2D laser range scanners, vision, and DEM. MCL in aerial photo.
[Allen et al., 2001]	3D-modelling of urban environments.	Vision and laser range scanner (3D). Odometry and DGPS.
[Howard et al., 2004]	Outdoor 3D-modelling, assumes constant altitude.	2 cross-mounted 2D laser range scanners. IMU, odometry, laser range scanner for fine localization, GPS or MCL for coarse localization.
[Abuhadrous et al., 2004]	3D-modelling of roads, building facades and trees.	Vertically mounted 2D laser range scanner with 270° field-of-view. GPS and odometry.

Table 3.1: Summary of 3D-modelling projects where MCL is the abbreviation of Monte Carlo localization and DEM means Digital Elevation Model. Cross-mounted laser range scanners means a configuration where the planes spanned by the laser beams are perpendicular to each other.

3.4 Applications Using Semantic Information

In this section several examples are given that demonstrate the broad use of semantic information for different applications. The section finishes with an example where the use of semantic information is suggested but not yet well explored.

The importance of semantic information in the form of human spatial concepts is evident in the communication between robots and humans. A project that aims for dialogs with robots is "Talking to Godot" [Theobalt et al., 2002]. The representation of the environment uses three layers; grid map, topological map, and a semantic map that connects regions in the topological map with semantic symbols [Bos et al., 2003]. Other robot systems that use dialog functions are the already mentioned [Topp and Christensen, 2006] and [Spexard et al., 2006]. Skubic *et al.* discussed the benefits of linguistic spatial descriptions for different types of robot control, and pointed out that this is especially important when there are novice robot users [Skubic et al., 2003]. In these situations it is necessary for the robot to be able to relate its sensor readings to human spatial concepts.

Semantic information extracted from an aerial image has been used for localization [Oh et al., 2004]. Oh *et al.* used map data to bias a robot motion model in a Bayesian filter to areas with higher probability of robot presence.

They assumed that mobile robot trajectories are more likely to follow paths in the map and that semantic information in the form of probable paths was known. Using these assumptions, GPS position errors due to reflections from buildings were compensated.

Stachniss *et al.* make use of the space labelling technique introduced by Mozos [Mozos, 2004] for multi-robot exploration [Stachniss, 2006, Stachniss et al., 2006]. The idea is to use the semantic information in the form of space labels to direct the robots in an efficient way. A corridor is a natural place to start mapping an area since it often has connections to different rooms. Stachniss shows by simulation of up to 50 mobile robots that the exploration time is reduced when this type of strategy is applied.

To be able to benchmark SLAM-algorithms in urban environments, Wulf *et al.* use semantic information as a link between a 3D city model and drawings of the buildings [Wulf et al., 2007]. By extracting walls from the 3D data they could obtain a verification of the model accuracy by comparing the 3D data belonging to walls with the walls in the drawings.

Another area where semantic information has been used is in execution monitoring [Bouguerra et al., 2007]. In execution monitoring the goal is to find problems in the execution of a plan. Semantic knowledge was organised in a knowledge base with definitions of concepts and relations between them. Such a relation can, for example, be that a "*bedroom has-at-least one bed*". The semantic information was encoded using description logics and a system for knowledge representation and reasoning, LOOM[1], was used for managing the semantic information.

In [Nielsen et al., 2004] semantics is defined to give meaning to something. To give meaning to an environment the authors let a mobile robot take images (snapshots) of the environment and tag these with the present location. A test is performed where novice robot operators should fulfil a certain task using either a standard 2D map or a 3D-map augmented with snapshots. The practical experiments showed that navigation was facilitated with the latter configuration and the operators experienced that they had better control of the robot.

Semantic information can be used in mobile robotics in, e.g., search and exploration, SLAM, and perception [Calisi et al., 2007]. SLAM, as an example, is a well-studied problem that most often takes into account geometrical formations. The combination of SLAM and semantic information has been proposed by Dellaert and Bruemmer [Dellaert and Bruemmer, 2004]. But, while the use of semantic and contextual information has been shown to be promising in many situations, it has rarely been incorporated in the mapping process [Calisi et al., 2007].

[1] http://www.isi.edu/isd/LOOM/

3.5 Summary and Conclusions

This chapter has presented an overview of literature on semantic mapping. The major part of the work that deals with semantic mapping considers an indoor environment and only a few works deal with outdoor environments. The scope of the literature survey was therefore extended to include also projects dealing with 3D-modelling of urban environments where semantic information implicitly has an important role in order to reduce effects from false readings and occlusions.

Semantic information has been shown to be useful in a number of situations. The most obvious is in human robot interaction where the semantic information facilitates communication in both directions. Other examples where semantic knowledge can be used include localization, exploration, evaluation, execution monitoring, and remote control.

Several promising concepts where topological and metric maps are labelled with semantic information have been presented in recent years and interest in semantic mapping seems to be increasing. Hybrid maps in the form of hierarchies where lower layers represent metric and topological information, and upper layers contain the semantic information are used by a number of researchers. Examples of methods that can automatically classify objects and spaces, mainly indoor, into semantic classes have been given.

Based on the content and references presented in this chapter it can be noted that there is an increasing interest in semantic mapping. This interest has so far been concentrated to indoor environments. There is therefore a gap in current research regarding outdoor semantic mapping, a gap that this thesis intends to fill in.

Chapter 4
Virtual Sensor for Semantic Labelling

4.1 Introduction

To enable human operators to interact with mobile robots, semantic information is of high value, as pointed out in the previous chapter. Several other applications for the semantic information were also mentioned. To extract the semantic information the robot system needs a mechanism that can transform the sensor readings into human spatial concepts.

In this chapter it is shown that virtual sensors can constitute this mechanism, being the component that processes and interprets information about the surroundings from data collected by mobile robots. A virtual sensor is understood as one or several physical sensors with a dedicated signal processing unit for recognition of real world concepts. An important aspect of the virtual sensor proposed in this work is a general method for learning a particular instance of the virtual sensor from a set of generic features.

As an example, this chapter describes virtual sensors for detection of four human spatial concepts: buildings, nature, windows and trucks. Three of these concepts are man-made structures. The main example is a virtual sensor for building detection using methods for classification of views as buildings or non-buildings based on vision. The purpose of this is twofold. First of all, buildings are one type of very distinct objects that often is used in, e.g., textual description of route directions. Second, the virtual sensor is needed as a building detector in Chapter 7, where building outlines identified by a ground-level mobile robot are used for segmentation of aerial images.

4.1.1 Outline

The method to obtain a virtual sensor for building detection is based on learning a mapping from a set of image features to a particular concept. It combines different types of features such as edge orientation, gray level clustering, and corners. Section 4.2 describes the feature extraction.

The AdaBoost algorithm [Freund and Schapire, 1997] is used for learning a classifier for monocular gray scale images based on the features. AdaBoost has an ability to select the best so-called weak classifiers out of many others. The selected weak classifiers are linearly combined by AdaBoost to produce one strong classifier. Due to this selection process it is possible to start with a large feature set, calculate classifiers based on each feature and let AdaBoost select the best ones. This allows the same feature set to be utilized also for identification of other concepts. AdaBoost is presented in Section 4.3.

Many different learning approaches could possibly be used to solve the classification part of the virtual sensor. To compare the performance of AdaBoost with an alternative classifier, the Bayes Optimal Classifier (BOC) was selected. BOC uses the variance and covariance of the features in the training data to weight the importance of each feature. Section 4.4 introduces BOC.

Section 4.5 presents an evaluation of a virtual sensor for building detection based on building and nonbuilding images. It is shown that the virtual sensor robustly establishes the link between sensor data and the particular human spatial concept. Experiments with an outdoor mobile robot show that the method is able to separate the two classes with a high classification rate, and that the method extrapolates well to images collected under different conditions.

In Section 4.6, the virtual sensor for buildings is applied on a mobile robot, combining classifications of sub-images from a panoramic view with spatial information (location and orientation of the robot) in order to estimate the directions of buildings in an outdoor environment. This information could then be communicated, for example, to a remote human operator.

The method can be extended to virtual sensors for other human spatial concepts. This is demonstrated by learning and evaluating two further virtual sensors. Section 4.7 shows that the feature set can be used to distinguish between three human spatial concepts; buildings, nature, and windows, and in Section 4.8 a virtual sensor for trucks complements the first three virtual sensors, giving a total of four different human spatial concepts within the same experiment.

A summary and conclusions of this chapter are given in Section 4.9.

4.2 The Feature Set

A vision sensor onboard the mobile robot acquires the environmental data used by the virtual sensor to determine how the surroundings should be classified. These data are processed and stored as gray scale images and hence the feature set defined in this section contains features that can be extracted from this type of images. The objective of the feature set is to capture significant properties in the images allowing learned classifiers to distinguish between different concepts. To make such a method work for a wide variety of classes requires a

generic set of features. To define such a set is hard and therefore a heuristic method has been used in the design of the feature set.

The philosophy regarding the selection of features that should be included in the feature set is to include features that are believed to contribute with a high probability to an increase of the classification rates of the virtual sensors. Since AdaBoost is used as the learning mechanism only features that in fact contribute will be used in the final classifier. It is therefore not a problem if the feature set contains features that are of modest use for a particular classification task. A benefit of this approach is that whenever a classification problem needs some more features to perform well, these features can be added to the feature set. Adding features can in turn also result in better performance of previously trained virtual sensors if they are retrained utilizing the extended feature set.

The first virtual sensor presented here is the virtual sensor for the concept of buildings. The discussion therefore continues with an analysis of suitable features for building detection. Many systems for building detection, both for aerial and ground-level images, use line and edge related features. Building detection from ground-level images often uses the fact that, in many cases, buildings show mainly horizontal and vertical edges. In nature, on the other hand, edges tend to have more randomly distributed orientations. Inspection of histograms based on edge orientation confirms this observation. Histograms of edge direction can be classified by, e.g., Support Vector Machines (SVM). This has been used for detection of buildings in natural images [Malobabic et al., 2005]. Another method, developed for building detection in content-based image retrieval, uses consistent line clusters with different properties [Li and Shapiro, 2002]. These properties are based on edge orientation, edge colours, and edge positions.

An example of other features that have been used in outdoor environments is the output from Gabor filters. Gabor filters are considered as a robust method for texture estimation and have been used for recognising roads in aerial images and natural objects such as trees and coral [Ramos et al., 2006]. The images were divided into 11×11 pixel patches and filtered using four Gabor filters defined for two scales and two orientations.

Another example of features extracted from images taken by a mobile robot was presented in [Morita et al., 2005, 2006]. Features were calculated from image patches (16×16 pixels) taken from the upper part of the images and consisted of the average of R, G, and B normalized colours, the edge density, the distribution of edge orientation, and the number of line segments. The patches were classified into three classes; trees, buildings and sky (uniform) with a SVM and the classification result was used for self-localization of the mobile robot along a previously visited path.

For the virtual sensor presented in this section, a large number of image features that are extracted from the whole image have been selected. The features can be divided into three groups. The obvious indication of man-made structures is that they have a high content of vertical and horizontal edges, which is

also confirmed in the related work presented above. The first type of features represents this property. The second type of features combines the edges into more complex structures such as corners and rectangles. The third type of features is based on the assumption that certain types of concepts contain surfaces in homogeneous colours, e.g., facades.

The features that are calculated for each image are numbered 1 to 24. All features except 9 and 13 are normalized in order to avoid scaling problems. This feature set was selected with regard to a virtual sensor for building detection. An extended form of the feature set is used in Section 4.8 for a virtual sensor for truck detection.

4.2.1 Edge Orientation

The first group of features is based on edge orientation. The calculation of the orientations includes a few steps. First an edge image is calculated resulting in a binary image. From this binary image straight line segments representing straight edges are calculated and finally the orientations of these lines are used to compute the features.

For edge detection Canny's edge detector [Canny, 1986] is used. It includes a Gaussian filter and is less likely than others to be affected by noise. A drawback is that the Gaussian filter can distort straight lines. For line extraction in the edge image an implementation by Peter Kovesi [Kovesi, 2000] was used, see Appendix B.1. The absolute values of the line orientations are calculated and sorted into different histograms. The features based on edge orientation are:

1. 3-bin histogram of absolute edge orientation values.
2. 8-bin histogram of absolute edge orientation values.
3. Fraction of vertical lines out of the total number.
4. Fraction of horizontal lines.
5. Fraction of non-horizontal and non-vertical lines.
6. As 1) but based only on edges longer than 20% of the longest edge.
7. As 1) but based only on edges longer than 10% of the shortest image side.
8. As 1) but weighted with the lengths of the edges.

To be able to compare features 1-3 and 6-8 between images with a different number of lines, each bin value is divided by the total number of lines in the histogram, giving relative values of the orientation distribution.

Feature 1

Feature 1, denoted as $\vec{f_1}$, is a 3-bin histogram of the absolute edge orientation values, where the edge orientation is defined in the interval $[-\pi/2, \pi/2]$. The histogram has class limits $[0, 0.2, \pi/2-0.2, \pi/2]$ radians giving that $\vec{f_1}$ is a vector with 3 elements.

$$\vec{H_3} = Hist(|\vec{\theta}|, c_3),\ c_3 = [0, 0.2, \pi/2 - 0.2, \pi/2] \tag{4.1}$$

where $\vec{\theta}$ is the orientations of the edges. The normalized $\vec{f_1}$ is then calculated as

$$\vec{f_1} = \vec{H_3} / \sum \vec{H_3}. \tag{4.2}$$

Feature 2

Feature 2, $\vec{f_2}$, is defined in the same way as $\vec{f_1}$ with the only difference that an 8-bin histogram is used with limits $[0, \pi/16, \ldots, 7\pi/16, \pi/2]$:

$$\vec{H_8} = Hist(|\vec{\theta}|, c_8),\ c_8 = [0, \pi/16, \ldots, 7\pi/16, \pi/2]. \tag{4.3}$$

$$\vec{f_2} = \vec{H_8} / \sum \vec{H_8}. \tag{4.4}$$

Feature 3 to 5

Feature 3 is the fraction of vertical lines, taken from $\vec{f_1}$, out of the total number of lines:

$$f_3 = \vec{f_1}(3) \tag{4.5}$$

where $\vec{f_1}(3)$ denotes the member value of bin 3. Feature 4, the fraction of horizontal lines and Feature 5, the fraction of non-horizontal and non-vertical lines, are defined in an analogous way:

$$f_4 = \vec{f_1}(1) \tag{4.6}$$

$$f_5 = \vec{f_1}(2) = 1 - \vec{f_1}(1) - \vec{f_1}(3). \tag{4.7}$$

Features 6 to 8

In Feature 1 to 5, all line segments found in the image were used. To capture only the major lines in the image, features 6, 7, and 8 decrease the influence of short lines. Feature 6 is calculated like $\vec{f_1}$, but only with the edges longer than 20% of the longest edge. In Feature 7, the calculations are based on the edges that are longer than 10% of the shortest image side and in Feature 8 edge

orientation members are weighted with the edge length in the calculation of the histogram.

4.2.2 Edge Combinations

The lines found to compute features 1-8 can be combined to form corners and rectangles. The features based on these combinations are:

9. Number of right-angled corners.
10. 9) divided by the number of edges.
11. Fraction of right-angled corners with direction angles close to $45° + n \cdot 90°, n \in 0, \ldots, 3$.
12. 11) divided by the number of edges.
13. The number of rectangles.
14. 13) divided by the number of corners.

Features 9 and 10

Feature 9 is the number of right-angled corners in the image. Here, a right-angled corner is defined as two lines with close end points (maximum 4 pixels separation) and $90° \pm \beta_{dev}$ angle in between. During the experiments $\beta_{dev} = 20°$ was used. Feature 10 is the number of right-angled corners relative to the number of edges:

$$f_{10} = f_9 / \sum \vec{H}_3. \tag{4.8}$$

Features 11 and 12

For buildings with vertical and horizontal lines from doors and windows, the corners most often have a direction of $45°$, $135°$, $225°$ and $315°$, assuming that the camera is oriented correctly, see Section 4.2.5. The direction is defined as the 'mean' value of the orientation angle for the two lines defining the corner. This property is captured in feature 11.

$$f_{11} = C_{45} / f_9 \tag{4.9}$$

where C_{45} is the number of right-angled corners with the above defined directions $\pm 15°$. Feature 12 is Feature 11 relative to the number of edges:

$$f_{12} = f_{11} / \sum \vec{H}_3. \tag{4.10}$$

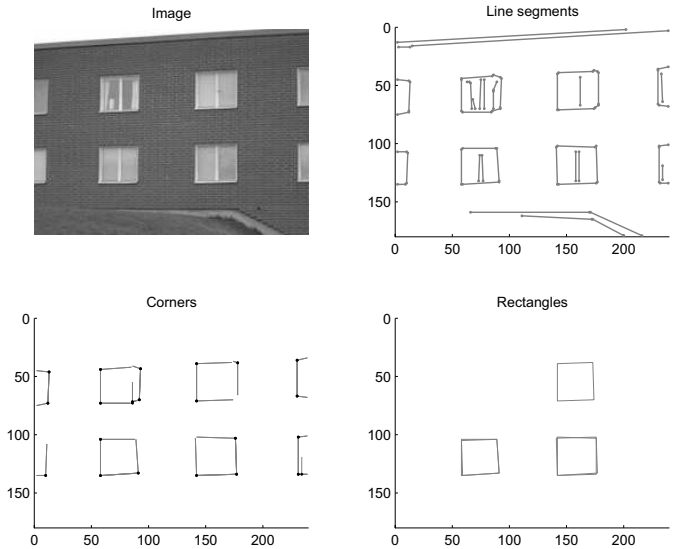


Figure 4.1: Example of extraction of edge combination features. The upper right part shows line segments extracted from the image to the left. The lower left image shows the lines that are connected as corners and the lower right part shows rectangles formed from the corners.

Features 13 and 14

From the lines and corners defined above rectangles are detected. A rectangle is formed from two corners and the pair of lines constructing these corners. The conditions on the final rectangle are that all four corners shall have an angle of $90° \pm \beta_{dev}$ and that the distance between the line endpoints forming the rectangle corners are maximum 4 pixels. The number of rectangles is stored in Feature 13 and Feature 14 is the number of rectangles relative to the number of right-angled corners:

$$f_{14} = f_{13}/f_9 \tag{4.11}$$

An example on the extraction of edges, corners and rectangles is given in Figure 4.1.

4.2.3 Gray Levels

Unlike to the above defined features using edges, features 15 to 24 are based on gray levels. Features 15 to 19 use gray level histograms to estimate if dominating gray levels exist. Features 20 to 24 are computed from connected areas to

quantify the existence of homogeneous areas. The features are scaled with the image size.

Features 15 to 19

Equally binned gray level histogram with 25 bins are used and the result is normalized with the image size. In order to capture several dominating gray levels the largest bins are accumulated in features 16–19. These features capture the distribution of gray levels in the image.

15. Largest value in gray level histogram.
16. Sum of the 2 largest values in gray level histogram.
17. Sum of the 3 largest values in gray level histogram.
18. Sum of the 4 largest values in gray level histogram.
19. Sum of the 5 largest values in gray level histogram.

Features 20 to 24

With features 20–24, images showing large variations in intensity can be separated from those that have large homogeneous areas. Typical areas that can be homogeneous are, e.g., building facades, roads, lawns, water and the sky. To find local areas with homogeneous gray levels a search for the largest connected areas with similar gray level is performed. Gray levels are considered to be similar if they fall into the same bin of the 25-bin histogram. The largest regions of interest that are 4-connected are calculated and up to the 5 largest are accumulated in features 21–24:

20. Largest 4-connected area.
21. Sum of the 2 largest 4-connected areas.
22. Sum of the 3 largest 4-connected areas.
23. Sum of the 4 largest 4-connected areas.
24. Sum of the 5 largest 4-connected areas.

4.2.4 Camera Invariance

In the work presented in this chapter, virtual sensors will use input data that were obtained from a number of different cameras. To illustrate that the defined feature set can be utilized with different cameras, experiments using different image sets are performed in Section 4.5. In this paragraph, a direct comparison of images taken by two of the different cameras is performed to illustrate that

Figure 4.2: The seven image pairs used for the camera invariance test. The left image in each pair was obtained from the digital camera and the right image in each pair was obtained from the mobile phone camera.

the defined features are not really affected by the choice of camera. The cameras are a 5 megapixel Sony DSC-P92 digital camera and a Sony Ericsson K750i mobile phone 2 megapixel camera, see Section 2.4.

The evaluation of the feature set is performed on seven image pairs. Each pair was taken at the same place and with the handheld cameras. The images were manually cropped to cover essentially the same area. The cropped images are shown in Figure 4.2.

Features were calculated for each of the 14 images and each image was compared with all images, giving a total of 196 comparisons. The comparison was performed using a similarity measure M calculated from r_i, a measure for each individual feature that can vary between 0 (identical features) and 1. The similarity measure M was calculated as the mean value of r_i

$$M = \frac{1}{24} \sum_{i=1}^{24} r_i \tag{4.12}$$

where i denotes the feature number. The measure r_i is calculated as a weighted mean value

$$r_i = \sum_{j=1}^{N} r_{i_j} w_j \tag{4.13}$$

for the features that are histograms. For each bin j, r_{i_j} is calculated as

$$r_{i_j} = \frac{\left| f_{i_j}^c - f_{i_j}^m \right|}{\max(f_{i_j}^c, f_{i_j}^m)} \tag{4.14}$$

with c and m denoting the digital camera and the mobile phone respectively. The above calculation guarantees that $r_{i_j} \in [0, 1]$. The weights w_j are introduced to put higher weights on the relative differences originating from bins with a large number of hits. They are calculated as

$$w_j = \frac{f_{i_j}^c + f_{i_j}^m}{\sum_{j=1}^{N}(f_{i_j}^c + f_{i_j}^m)} \tag{4.15}$$

with N being the number of bins. For the single value features, when $N = 1$, Equations 4.13 - 4.15 can be simplified to

$$r_i = \frac{|f_i^c - f_i^m|}{\max(f_i^c, f_i^m)} \tag{4.16}$$

and no weights need to be used.

The results from the comparisons are stored in a symmetric matrix that is illustrated in Figure 4.3 by squares representing the measure M for each image combination. The first seven rows and columns correspond to the digital camera and the following seven rows and columns to the mobile phone. Low M-values from the comparison are represented by dark squares and high M-values are represented by lighter gray shades. The figure shows two distinct diagonals. The diagonal starting in the upper left corner is the self-comparison (features from image 1 are compared with features from image 1 etc.) and all squares therefore got the M-value 0 (black). The other diagonal represents the image pairs taken at the same location (image 1 is in pair with image 14, 2 with 13 etc.) and got M-values between 0.092 and 0.205. All other squares represent pairing of images taken at different locations. The M-values for these squares were in the interval of 0.28 to 0.54. This result implies that the features from the image pairs are more similar than the features from any other combination of images taken with the same camera.

By this test it has been shown that for the selected seven image pairs, the feature set clearly distinguishes between different scenes instead of different cameras. The conclusion is therefore that the extracted features are sufficiently invariant to these cameras meaning that alternating between these cameras should not affect adversely the performance of the system.

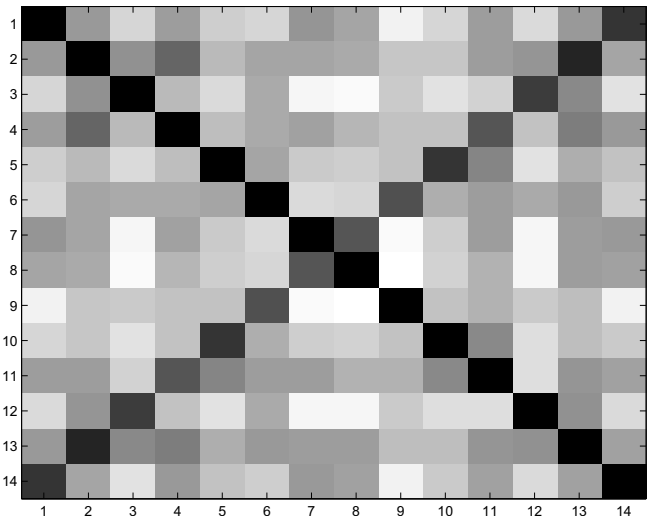


Figure 4.3: Result of the camera invariance test. Rows/columns 1-7 correspond to the digital camera and rows/columns 8-14 to the mobile phone. Dark squares show high similarity.

4.2.5 Assumptions

The calculation of the features is based on the assumptions that the camera orientation is close to horizontal and that the perspective distortion is small. Concerning the first assumption, the presented system is not rotational invariant, i.e., it cannot handle images with arbitrary rotational angles. A function that can estimate the rotational angle can be implemented by identification of peaks in the edge histograms assuming that the approximate 90 degree difference between horizontal and vertical edges exists. On the other hand, it is probably not needed to perform that operation, since an outdoor robot that is supposed to navigate in non-flat terrain needs to estimate not only position and orientation, but also pitch and roll angles. It is therefore assumed that the images can be re-rotated with an accuracy that is sufficient for the system (e.g. the smallest bin used for edge orientation classifies an edge as horizontal if it is within $\pm 11°$).

The second assumption concerns the perspective of the images. It is assumed that the optical axis of the camera has a limited deviation from the normal of the object surfaces in the images. If this is not the case the use of histograms for edge directions is not optimal for the performed task. In the system proposed in this thesis, small deviations are compensated for by the quantization into only a few histogram bins and the tolerances in the definition of qualitative features such as corners. It is possible to transform an image in order to restore

the perspective of the main surface. One way is to find the vanishing point, but this is often a costly operation. Another possibility, if 3D-models of the scene exist, is to use image depth information to obtain the parameters needed for a perspective correction.

4.3 AdaBoost

AdaBoost is the abbreviation for adaptive boosting. It was developed by Freund and Schapire [Freund and Schapire, 1997] and has been used in diverse applications, e.g., as classifiers for image retrieval [Tieu and Viola, 2000] and real-time face detection [Viola and Jones, 2004]. In mobile robotics, AdaBoost has, e.g., been used in ball tracking for soccer-robots [Treptow et al., 2003] and to classify laser scans for learning of places in indoor environments [Mozos et al., 2005]. Mozos' work is a nice demonstration of the use of machine learning and a set of generic features to transform sensor readings to human spatial concepts. A cascaded classifier was developed where the first steps should be fast and have a very low false negative rate. The following steps include classifiers with increasing complexity that can resolve the 'hard cases' from the set of positive classifications found in the first steps.

The main purpose of AdaBoost is to produce a strong classifier by a linear combination of weak classifiers, where *weak* means that the classification rate has to be only slightly better than 0.5 (better than guessing). Figure 4.4 shows pseudo code for the implemented AdaBoost algorithm (see [Schapire, 1999] for a formal algorithm). The principle of AdaBoost is as follows.

The input to the algorithm is a number, N, of positive and negative examples. The training phase is a loop. For each iteration t, the best weak classifier h_t is calculated and a distribution $\boldsymbol{D}_t$ is recalculated. The boosting process uses $\boldsymbol{D}_t$ to increase the weights of the hard training examples in order to focus the weak learners on the hard examples. In our case $\boldsymbol{D}_t$ has been used to bias the hard examples by including it in the calculation of a weighted mean value for the MDC, see Section 4.3.1.

Implementing and using AdaBoost, one should note a few things. First, if the evaluation of training examples results in 100% correct classification by a weak classifier, there will be no more change in D_t ($D_{t+1} = D_t$). If this happens in the first iteration only one feature will be used in the *strong classifier*. Second, the general AdaBoost algorithm does not include rules on how to choose the number of iterations T of the training loop. The training process can be aborted if the distribution $\boldsymbol{D}_t$ does not change, or alternatively a fixed maximum number of iterations T is used. Boosting is known to be not particularly prone to the problem of overfitting [Schapire, 1999]. $T = 30$ was used for training in all the experiments presented and no indications of overfitting were noted when evaluating the performance of the classifier on an independent test set.

- Use N training examples $(x_1, y_1) \ldots (x_N, y_N)$, where $y_n = 1$ for N_p positive examples and $y_n = -1$ for N_n negative examples ($N = N_p + N_n$)
- Initialize the distribution: $\boldsymbol{D}_1(n_p) = 1/N_p$ for the positive examples (with indices n_p) and $\boldsymbol{D}_1(n_n) = 1/N_n$ for the negative examples (with indices n_n)
- For $t = 1, \ldots, T$:
  1. Normalize distribution $\boldsymbol{D}_t$
  2. Train weak learners using distribution $\boldsymbol{D}_t$
  3. Find weak hypothesis h_j that minimizes ϵ_j and set $h_t = h_j$ and $\epsilon_t = \epsilon_j$. The goodness of a weak hypothesis h_j is measured by its error $\epsilon_j = \sum_{n:h_j(x_n) \neq y_n} \boldsymbol{D}_t(n)$.
  4. Set $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$
  5. Update $\boldsymbol{D}_t$:
     $\boldsymbol{D}_{t+1}(n) = \boldsymbol{D}_t(n) \exp(-\alpha_t y_n h_t(x_n))$
- The final strong classifier is given by:
  $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Figure 4.4: General AdaBoost algorithm, where ϵ_t is a measure of the performance of the best weak classifier h_t at t. Index j points out the used feature, and α_t is the weight of h_t used both in the update of $\boldsymbol{D}_t$ and in the final strong classifier $H(x)$.

4.3.1 Weak Classifiers

In its standard form, the weak classifiers in AdaBoost use single value features. To be also able to handle feature arrays from the histogram data, a minimum distance classifier (MDC) was used to calculate a scalar weak classifier. $\boldsymbol{D}_t$ was used to bias the hard training examples by including it in the calculation of a weighted mean value for the MDC prototype vector:

$$\boldsymbol{m}_{l,k,t} = \frac{\sum_{\{n=1\ldots N | y_n = k\}} \boldsymbol{f}(n,l) \boldsymbol{D}_t(n)}{\sum_{\{n=1\ldots N | y_n = k\}} \boldsymbol{D}_t(n)} \tag{4.17}$$

where $\boldsymbol{m}_{l,k,t}$ is the mean value array for iteration t, class k, and feature l and y_n is the class of the n:th image. The features for each image are stored in $\boldsymbol{f}(n,l)$ where n is the image number. For evaluation of the MDC at iteration t, a distance value $d_{k,l}(n)$ for each of the two classes k is calculated as

$$d_{k,l}(n) = \|\boldsymbol{f}(n,l) - \boldsymbol{m}_{l,k,t}\| \tag{4.18}$$

and the shortest distance for each feature l indicates the winning class for that feature.

4.4 Bayes Classifier

It is instructive to compare the result from AdaBoost with another classifier and for that Bayes Classifier was selected. Bayes Classifier, or Bayes Optimal Classifier (BOC) as it is sometimes called, classifies normally distributed data with a minimum misclassification rate. The decision function is [Duda et al., 2001]

$$d_k(\boldsymbol{x}) = \ln P(w_k) - \frac{1}{2}\ln|\boldsymbol{C}_k| - \frac{1}{2}[(\boldsymbol{x}-\boldsymbol{m}_k)^T\boldsymbol{C}_k^{-1}(\boldsymbol{x}-\boldsymbol{m}_k)] \tag{4.19}$$

where $P(w_k)$ is the prior probability (in the experiments set to 0.5), $\boldsymbol{m}_k$ is the mean vector of class k, and $\boldsymbol{C}_k$ is the covariance matrix of class k calculated on the training set, and $\boldsymbol{x}$ is the feature value to be classified.

Not all of the defined features can be used in BOC. Linear dependencies between features give numerical problems in the calculation of the decision function. Therefore normalized histograms cannot be used, hence features $\vec{f}_1$, $\vec{f}_2$, $\vec{f}_6$, $\vec{f}_7$, and $\vec{f}_8$ were not considered. The set of features used in BOC was represented by the following vector:

$$\vec{f} = [f_3, f_4, f_9 \ldots f_{15}, f_{17}, f_{20}, f_{23}]^T. \tag{4.20}$$

This set was constructed by starting with the best individual feature as found by running AdaBoost (see Figure 4.7, Section 4.5.3) and adding the second best feature etc., while observing the condition value of the covariance matrices.

4.5 Evaluation of a Virtual Sensor for Buildings

In this section a virtual sensor for building detection is evaluated using images from two classes; buildings and nonbuildings. For this type of classification it is generally difficult to define representative negative samples for the training set. In order to cope with this, nature images were used as the representation of nonbuildings in the evaluation presented here and, hence, *nature* is used as the name of the class nonbuildings. This shall not be seen as a limitation since the virtual sensor is learned and the training set can be adapted to an intended operational environment.

Results for AdaBoost are compared with results using BOC and two different image resolutions are tested and compared.

4.5.1 Image Sets

Three different sources were used for the collection of nature and building images utilized in the experiments. For Set 1, images were taken by an ordinary consumer digital camera. These images were taken over a period of several months in outdoor environments that are potential operational areas for a mobile robot. The virtual sensor was designed to be used to classify images taken by a mobile robot. Therefore, Set 2 consists of images taken on manually controlled runs with a mobile robot, performed on two different occasions. Set 1 and 2 are disjunctive in the sense that the images do not depict the same buildings or the same nature views.

In order to verify the system performance with an independent set of images, Set 3 contains images downloaded[1] from the Internet using Google's Image Search. For buildings the search term *building* was used. The first 50 images with a minimum resolution of 240×180 pixels containing a dominant building were downloaded. For nature images, the search terms *nature* (15 images), *vegetation* (20 images), and *tree* (15 images), were used. Only images that directly applied to the search term and were photos of real environments (no arts or computer graphics) were used. Borders and text around some images were removed manually. Table 4.1 summarizes the different sets of images and the number of images in each set.

All images were converted to gray scale and stored in two different resolutions (maximum side length 120 pixels and 240 pixels, referred to as size 120 and 240 respectively). In this way the performance for different resolutions can be compared. In the case of comparable performance, using low resolution images has the advantage of faster computations and decreased demands on the used equipment or alternatively, sub-images depicting objects at longer distances will be possible to use.

Examples of images from Set 1 and 2 are shown in Figure 4.5. Note the difference between the building images in Set 1 and 2. The images taken with the handheld cameras are most often centred on the view while the optical axis of the camera on the robot was mostly horizontal. This resulted in images in Set 2 that only contain building objects in the upper half and where the lower part shows the ground surface.

4.5.2 Test Description

Five tests have been defined for evaluation of the virtual sensor. *Test 1* shows whether it is possible to collect training data with a consumer camera and use this for training of a classifier that is evaluated with a different camera on the intended platform, the mobile robot. *Test 2* trains and evaluates on a mixture of Set 1 and Set 2. *Test 3* shows how well the learned model, trained with images taken in our neighbourhood with known equipment, extrapolates to

[1]The images were downloaded 7^{th} Sept 2005.

Set	Origin	Area	Buildings	Nature
1	Handheld digital camera	Urban and nature	40	40
2	Camera on mobile robot	Örebro Campus	66	24
3	Internet search	Worldwide	50	50
	Total number		156	114

Table 4.1: Summary of the image sets used for the evaluation of the virtual sensor for buildings. The digital camera is a 5 megapixel Sony (DSC-P92) and the mobile camera is an analogue camera mounted on the mobile robot Rasmus, see Section 2.3.

images taken around the world by different photographers. *Test 4* evaluates the performance on the complete collection of images. Table 4.2 summarizes the test cases. These tests have been performed with AdaBoost and BOC separately for each of the two image sizes. For Test 2 and 4, holdout validation [Kohavi, 1995] is performed. A random function is used to select the training partition (90% of the images) and the images not selected are used for the evaluation of the classifiers. This was repeated N_{run} times.

No.	N_{run}	Train Set	Test Set
1	1	1	2
2	100	90% of {1,2}	10% of {1,2}
3	1	{1,2}	3
4	100	90% of {1,2,3}	10% of {1,2,3}
5	100	90% of {1,2}	10% of {1,2}

Table 4.2: Description of defined tests (N_{run} is the number of runs).

A fifth test, *Test 5*, is designed to test scale invariant properties of the system. Test 2 is repeated, but now the training is performed with one image size and the evaluation is performed with the other image size.

4.5.3 Analysis of the Training Results

AdaBoost can compute multiple weak classifiers from the same features by means of a different threshold, for example. Figure 4.6 presents statistics on the usage of different features in Test 2. The feature most often used for image size 240 is the orientation histogram ($\vec{f_2}$). For image size 120, features $\vec{f_2}$, f_8 (corners), f_{13} (rectangles) and f_{14} (rectangles related to corners) dominate. Figure 4.7 shows how well each individual feature manages to classify images in Test 2. Several of the histograms based on edge orientation are in themselves close to

Figure 4.5: Examples of images used for training. The uppermost row shows buildings in Set 1. The following rows show buildings in Set 2, nature in Set 1, and nature in Set 2.

the result achieved for the classifiers presented in the next section. Comparing Figure 4.6 and Figure 4.7 one can note that several features with high classification rates are not used by AdaBoost to the expected extent, e.g., features $\vec{f}_1$, f_3, f_4, and f_5. This is most likely caused by the way in which the distribution $\boldsymbol{D}_t$ is updated. Because the importance of correctly classified examples is decreased after a particular weak classifier is added to the strong classifier, similar weak classifiers might not be selected in subsequent iterations.

As a comparison to the results of *Test 1* to *Test 4* presented in Section 4.5.4, the result obtained on the training data using combinations of image sets is also presented in Table 4.3. Φ_{TP} is the true positive rate (in this case related to buildings), Φ_{TN} is the true negative rate (related to nature), and Φ_{Acc} is the accuracy[2].

4.5.4 Results

Training and evaluation were performed for the tests specified in Table 4.2 with features extracted from images of size 120 and 240 separately. The result is pre-

[2]The accuracy Φ_{Acc} is based on the sum of the true positives and the true negatives.

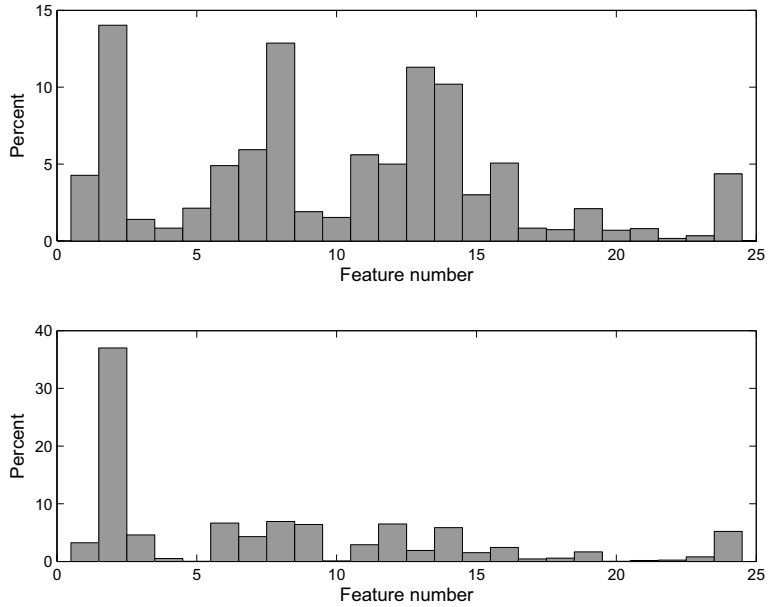


Figure 4.6: Histogram describing the frequency with which features are selected for the strong classifier by AdaBoost in Test 2 as an average of 100 runs, using image size 120 (upper) and 240 (lower).

sented in Table 4.4 and 4.5 respectively. The tables show Φ_{TP} (buildings), Φ_{TN} (nature), and Φ_{Acc} (the total classification rate with the corresponding standard deviation). Results from both AdaBoost and BOC using the same training and testing data are given.

In Test 1 a classification rate of over 92% is obtained for image size 240. This shows that it is possible to build a classifier based on digital camera images and achieve very good image classification results with another camera onboard the mobile robot, even though Set 1 and 2 have structural differences, see Section 4.5.1.

Test 2 is the most interesting test for us. Here, images that have been collected in the target environment of the mobile robot are used for both training and evaluation. This test shows high (and highest) classification rates. For both AdaBoost and BOC they are around 97% using the image size 240. Figure 4.8 shows the distribution of wrongly classified images for AdaBoost compared to BOC. It can be noted that for image size 120 several images give both classifiers problems, while for image size 240 different images cause problems. Figures 4.9 and 4.10 show the images that were wrongly classified. The numbers of the images relate to the image numbers in Figure 4.8.

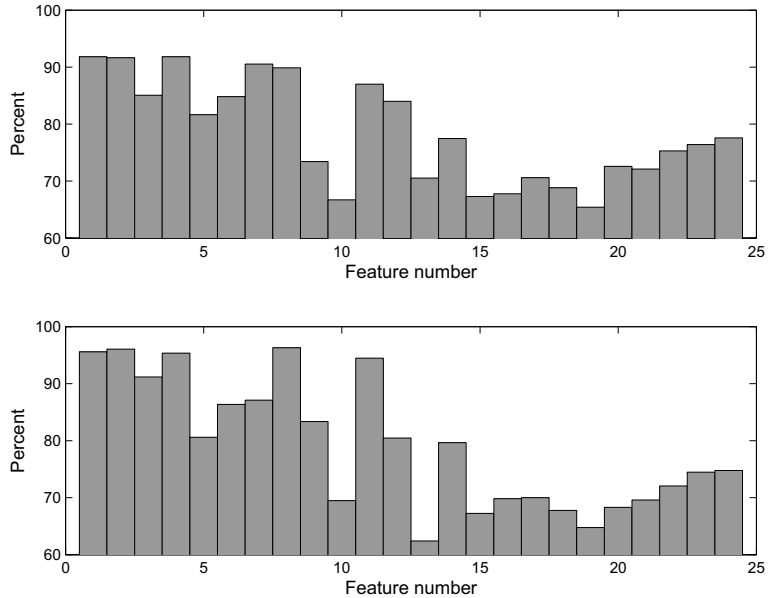


Figure 4.7: Histogram of classification rate of individual features in Test 2 as an average of 100 runs with $T = 1$, image size 120 (upper) and 240 (lower).

Test 3 is of the same type as Test 1. They both train on one set of images and then validate on a different set. Test 3 shows lower classification rates than Test 1 with the best result for AdaBoost using image size 240. It is not surprising that lower classification rates are obtained since the properties of the downloaded images differ from the other image sets. The main difference between the image sets is that the buildings in Set 3 often are larger and located at a greater distance from the camera. The same can be noted in the nature images, where Set 3 contains a number of landscape images that do not show close range objects. The conclusions from this test are that AdaBoost generalizes better than BOC and that the classification works very well even though the training images were taken in our neighbourhood and the images used for evaluation were downloaded from the Internet.

Comparing the results of Test 2 and Test 4, it can be noted that the classification rate is lower for Test 4, especially for image size 120. Investigation of the misclassified images in Test 4 shows that the share belonging to image Set 3 (Internet) is large. For both image sizes 60% of the misclassified images came from Set 3 although Set 3 only represent 37% of the total number of images. This shows, once more, that the Internet images are harder to classify due to their different properties.

Sets	Size	Classifier	Φ_{TP} [%]	Φ_{TN} [%]	Φ_{Acc} [%]
1	120	AdaBoost	100.0	100.0	100.0
		BOC	100.0	100.0	100.0
1,2	120	AdaBoost	97.2	100.0	98.2
		BOC	95.3	93.8	94.7
1,2,3	120	AdaBoost	89.7	94.7	91.9
		BOC	86.5	94.7	90.0
1	240	AdaBoost	100.0	100.0	100.0
		BOC	100.0	100.0	100.0
1,2	240	AdaBoost	100.0	100.0	100.0
		BOC	98.1	100.0	98.8
1,2,3	240	AdaBoost	98.7	99.1	98.9
		BOC	95.5	98.2	96.7

Table 4.3: Results from evaluation of the virtual sensor on the same image sets as used for the training. The image sets are defined in Table 4.1.

Test 5 demonstrates the scale invariance of the system by performing a cross resolution test. Classifiers were trained with images of size 120 and evaluated with images of size 240 and vice versa. The result is presented in Table 4.6 and should be compared to Test 2 in Tables 4.4 and 4.5. The conclusion from this test is that the features used have scale invariant properties over a certain range and that AdaBoost shows notably better scale invariance than BOC, which again demonstrates AdaBoost's better extrapolation capability.

Test no.	Classifier	Φ_{TP} [%]	Φ_{TN} [%]	Φ_{Acc} [%]
1	AdaBoost	81.8	91.7	84.4
	BOC	93.9	58.3	84.4
2	AdaBoost	93.0	91.8	92.6 ± 5.8
	BOC	95.7	89.0	93.4 ± 5.5
3	AdaBoost	68.0	90.0	79.0
	BOC	72.0	74.0	73.0
4	AdaBoost	86.6	89.8	87.9 ± 6.2
	BOC	86.4	88.5	87.3 ± 6.0

Table 4.4: Results for Test 1-4 using images with size 120.

Test no.	Classifier	Φ_{TP} [%]	Φ_{TN} [%]	Φ_{Acc} [%]
1	AdaBoost	89.4	100.0	92.2
	BOC	95.5	87.5	93.3
2	AdaBoost	96.1	98.3	96.9 ± 4.3
	BOC	98.1	95.7	97.2 ± 4.0
3	AdaBoost	88.0	94.0	91.0
	BOC	90.0	82.0	86.0
4	AdaBoost	94.1	95.5	94.6 ± 3.8
	BOC	94.8	93.4	94.2 ± 4.7

Table 4.5: Results for Test 1-4 using images with size 240.

Train	Test	Classifier	Φ_{TP} [%]	Φ_{TN} [%]	Φ_{Acc} [%]
120	240	AdaBoost	94.2	96.7	95.1 ± 4.2
		BOC	93.0	94.3	93.5 ± 5.3
240	120	AdaBoost	95.1	90.8	93.6 ± 6.0
		BOC	100.0	44.8	80.5 ± 6.7

Table 4.6: Results for Test 5, the cross resolution test where data from Test 2 were used. Training was performed with images of size 120 and testing with images of size 240 and vice versa.

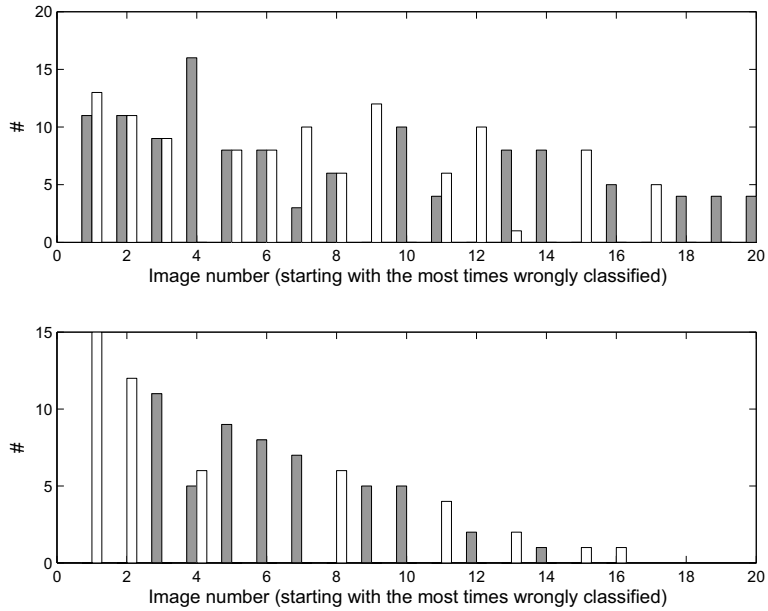


Figure 4.8: Distribution of the 20 most frequently wrongly classified images from AdaBoost (gray) and BOC (white) in Test 2, using image size 120 (upper) and 240 (lower).

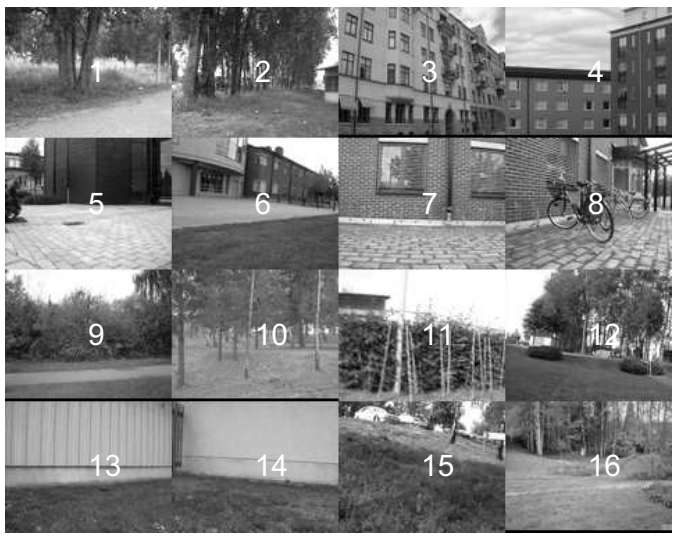


Figure 4.9: The 16 most frequently misclassified images in Test 2 with image size 120.

Figure 4.10: The 16 misclassified images in Test 2 with image size 240.

4.6 A Building Pointer

The learned building detection algorithm has been used to construct a virtual sensor. This sensor indicates the presence of buildings in different directions related to a mobile robot. In this case the robot performed a sweep with its camera (from $-120°$ to $+120°$ in relation to its heading) at a number of points along its track. The images were then classified into buildings and nonbuildings. The virtual sensor was trained using AdaBoost and the images in Set 1, see Section 4.5. The experiments were performed using Tjorven, a Pioneer robot equipped with GPS and a camera on a PT-head, see Section 2.2 for more details. Figure 4.11 shows the result of a tour in the Campus area. The arrows show the direction towards buildings and the lines point toward nonbuildings. Figure 4.12 shows an example of the captured images and their classes from a sweep with the camera at the first sweep point (the lowest rightmost sweep point in Figure 4.11).

This experiment was conducted with yet another camera than those used in Section 4.5 and during winter with no leaves on the trees and snow on the ground. An estimation of the performance based on the evaluation by a human expert is shown in Table 4.7. Some images are hard to classify into the two classes since they include large portions belonging to both of the classes. These images are denoted ambiguous and are shown in a separate column of the table. Problems were noted, for example, when a hedge is in front of a building or when the robot drives close to a bicycle stand and sees buildings and vegetation through the bikes. A specific error occurred at the second stop (lower middle stop) when a car stopped in front of the robot. Among the unambiguous images 89% where correctly classified. Note that the good generalization of AdaBoost together with a suitable feature set is expressed by the fact that the classifier was trained on images taken in a different environment and during a different season.

True	False	Ambiguous
76.4%	9.4%	14.2%

Table 4.7: Results for the building pointer. The column of ambiguous data represents those images that have a mixture of building and nature where neither dominates.

Figure 4.11: Using a virtual sensor to point out the class *buildings* along the mobile robot path. Arrows indicate buildings and lines nonbuildings. (Aerial image ©Örebro Community Planning Office.)

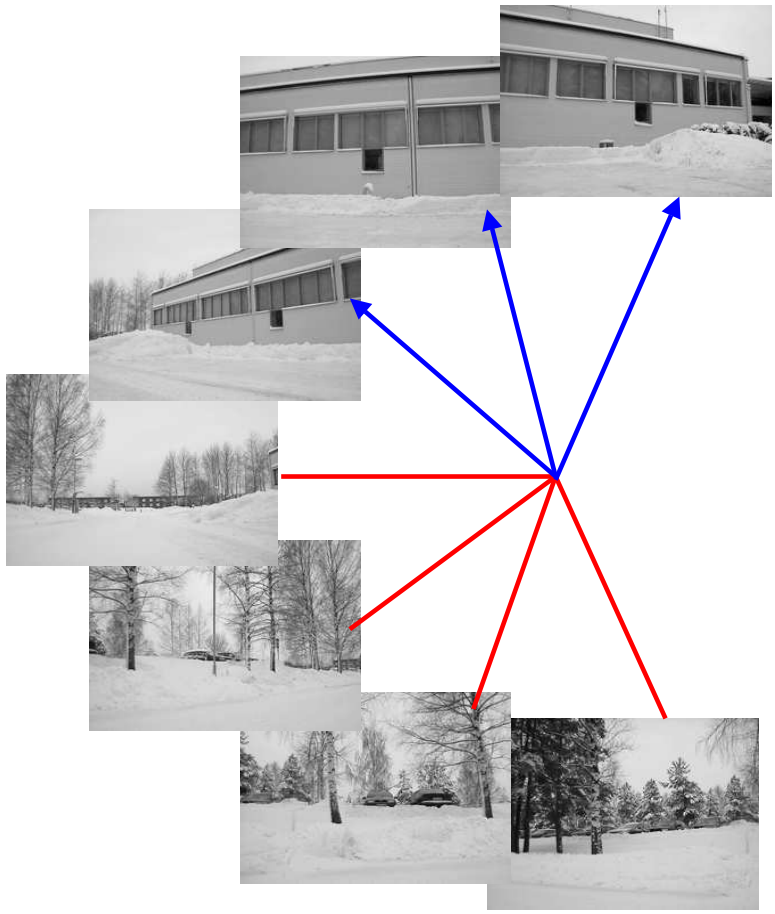


Figure 4.12: Example of a single sweep with the camera. The arrows point at images classified as buildings and the lines point at nonbuildings.

4.7 Evaluation of a Virtual Sensor for Windows

In the previous sections of this chapter, a virtual sensor has been presented. To illustrate the concept experiments with the aim of detecting buildings were presented. This section shows that the general approach defined for the virtual sensor is not limited to building detection. A third class, *windows*, is introduced in order to show that the set of features can be used to distinguish between classes that have similar structure. Both buildings and windows have a large content of vertical and horizontal lines, corners and rectangles.

4.7.1 Image Sets and Training

In the evaluation two sets of images, denoted A and B, are used. Set A consists of images taken by handheld digital cameras and include images of buildings, windows and nature. All nature images from Set 1 (Section 4.5.1) are included in Set A. The building images consist of the building images from Set 1 with the exception that a few images have been added to extend the image set and images that only contained single windows, and therefore belong to the *windows* class, were removed. Half of the window images were collected using a digital camera and the other half using a mobile phone camera, see Section 2.4. Set B consists of Set A extended with Set 3 consisting of Internet images of buildings and nature[3]. The two image sets and the number of images in each set are listed in Table 4.8. Examples of images in the window dataset are given in Figure 4.13.

Set	Windows	Buildings	Nature
A	100	50	50
B	100	100*	100*

Table 4.8: The number of images used for the tests. The images were collected with a digital camera (5 megapixel Sony DSC-P92), with a mobile phone camera (2 megapixel SonyEricsson K750i) and downloaded from the Internet. (* includes 50 from the Internet.)

One virtual sensor is trained for each image class, i.e., one virtual sensor for buildings, one for windows and one for nature. In the training process the images from the other two classes constituted the negative examples. The training was based on 90% of the positive and negative examples and the evaluation was performed on the remaining 10%. The selection of the training and test sets was random and was repeated 100 times in the same manner as described in Section 4.5.

[3]It was not as straightforward to search for windows as for buildings and nature images on the Internet. Hence no images have been collected from the Internet for the windows class.

Figure 4.13: Examples of images in the window dataset.

4.7.2 Result

The result from the evaluation of the different virtual sensors for image Sets A and B is presented in Tables 4.9 and 4.10. Table 4.9 shows the true positive rate, Φ_{TP}, the true negative rate, Φ_{TN}, the accuracy, Φ_{Acc}, and the normalized accuracy[4] Φ_{Nacc}. Table 4.10 divides the false positive images into their true classes. There are three columns with data extracted from the experiments. One cell in each row refers to the true positive rate, Φ_{TP} from Table 4.9, and the remaining two cells show the false positive rates for the respective classes in the negative dataset. The diagonals in the tables should contain high numbers and the other cells should contain low numbers.

Result for Set A The best results of the virtual sensors using the images taken with the digital camera are achieved for the nature virtual sensor with an accuracy of 98%. This shows that the feature set is able to separate man-made structures from natural objects with a high detection rate. The accuracy for the window virtual sensor and for the building virtual sensor is 87% and 89%

[4]The normalized accuracy Φ_{Nacc} is a value that takes the sizes of the positive and negative sets into account and represents a value for the case when the number of positives equals the number of negatives in the evaluated set. Φ_{Nacc} can be calculated as the mean value of Φ_{TP} and Φ_{TN}.

Test	VS	Set	Φ_{TP} [%]	Φ_{TN} [%]	Φ_{Acc} [%]	Φ_{Nacc} [%]
1	Windows	A	94.3	79.1	86.7 $\pm$ 7.2	86.7
2	Buildings	A	79.0	92.7	89.3 $\pm$ 7.0	85.8
3	Nature	A	98.6	97.6	97.8 $\pm$ 3.2	98.1
4	Windows	B	94.6	87.0	89.6 $\pm$ 4.7	90.8
5	Buildings	B	77.9	80.3	79.5 $\pm$ 6.9	79.1
6	Nature	B	97.5	95.3	96.1 $\pm$ 3.3	96.4

Table 4.9: Results from the evaluation of the windows virtual sensor based on 100 runs. Φ_{TP} is the true positive rate, Φ_{TN} is the true negative rate, Φ_{Acc} is the accuracy, and Φ_{Nacc} is the normalized accuracy.

Test	VS	Set	Wind. [%]	Build. [%]	Nat. [%]
1	Windows	A	94.3	35.8	6.0
2	Buildings	A	11.0	79.0	0.0
3	Nature	A	1.4	4.4	98.6
4	Windows	B	94.6	23.6	2.3
5	Buildings	B	24.1	77.9	15.2
6	Nature	B	2.9	6.4	97.5

Table 4.10: Results from the evaluation of the three virtual sensors (windows, buildings, and nature), using 100 runs, and with the result separated into the three classes.

respectively. It can be noted that the window virtual sensor has a larger true positive rate than the building virtual sensor. The reason for this is probably that the building images often include parts of the surroundings like small parts of vegetation and ground, while the surroundings of the windows are homogeneous walls. From Table 4.10 it can be noted that the nature virtual sensor has a low false positive rate and that the other two virtual sensors show low false positive rates for nature images (6% and 0%). The window virtual sensor has substantially more false positives from the building images than from the nature images (36% versus 6%).

Result for Set B Test 4-6, which includes additional images from the Internet, show the same tendency as Tests 1-3. The figures for Test 4 in Table 4.9 are higher than for Test 1, while Test 5 and 6 show slightly lower results. The largest change in Table 4.10 between Set A and B occurs for the building virtual sensor (Test 2 and 5). The true positive rate is approximately the same as for

Set A, but the false positive rate for *windows* has increased from 11% to 24% and for *nature* from 0% to 15%.

The sets of building images are more closely related to the images of windows and nature than window images are related to nature images. The separation between the different classes is therefore harder for the building virtual sensor, which is more evident when the Internet images with their larger variety are introduced. For instance, the building images show buildings from larger distances and some images also include vegetation, compare with Test 3 and 4 in Section 4.5.4.

4.8 Evaluation of a Virtual Sensor for Trucks

In this section, a virtual sensor for trucks is learned and evaluated. The image sets used in the previous section are extended with images of trucks and trailers. For the evaluation, virtual sensors for the four classes *windows*, *buildings*, *nature*, and *trucks* are learned.

A virtual sensor for trucks is useful in outdoor environments since buildings can be confused with other large objects and it can be beneficial to distinguish between stationary and movable objects in any mapping process.

4.8.1 Image Sets and Training

Image sets A and B used in Section 4.7 are extended with images of trucks that were collected manually using handheld digital cameras (Sony DSC-P92 and SonyEricsson K750i) and downloaded from the Internet[5]. The new sets were denoted as C and D, see Table 4.11. Examples of images from the truck dataset are given in Figure 4.14.

Set	Windows	Buildings	Nature	Trucks
C	100	50	50	50
D	100	100*	100*	100*

Table 4.11: The number of different images used for the tests presented in this section. The images were collected with a digital camera (5 megapixel Sony DSC-P92), with a mobile phone camera (2 megapixel SonyEricsson K750i) and from the Internet (* includes 50 from the Internet.).

As in the previous section, one virtual sensor was trained for each image class, i.e., one virtual sensor for buildings, one for windows, one for nature and one for trucks. In the training process the images from the other three classes constituted the negative examples. The training was based on 90% of

[5]Search terms “truck”, “lorry” and “lastbil” (lorry in Swedish) were used.

Figure 4.14: Example of images in the truck dataset. The upper two rows show trucks from set C and the lower two rows show truck images found on the Internet.

the positive and negative examples and the evaluation was performed on the remaining 10%. The selection of the training and test sets was random and was repeated 100 times in the same manner as described in Section 4.5.

The feature set that has been defined so far (features $\vec{f_1}$ to f_{24}) is based on straight edges and gray levels. One typical qualitative feature that could be used to detect vehicles is circular shapes from, e.g., wheels. In order to better capture properties that can be related to vehicles, new features were introduced for an additional test. These features aim to detect the vehicle's tyres and circular shaped fenders. The features added to the previously defined feature set are defined as:

25. Number of circle segments in the image that covers at least 75% of a circle, measured by dividing the length of the edge segment by the estimated perimeter of a found circle.

26. Number of circle segments in the image that cover between 25% and 50% of a circle.

27. Number of circle segments in the image that cover between 50% and 75% of a circle.

28. Radius to distance ratio. The mean value of the two most similar radii is divided by the distance between the circles. This feature is set to zero if less than two circles have been found.

The search for circles makes use of the Random Sample Consensus (RANSAC) algorithm [Fischler and Bolles, 1981] to map line segment points to circles. RANSAC is used to fit a model to data and is known to accept a large portion of outliers in the data. The used model for a circle is defined as a centre coordinate and the radius of the circle. The algorithm randomly selects three points from the dataset (three points are needed to define a circle) and calculates the circle model that fits the selected points. The remaining points in the dataset are evaluated against this model and those points that are within a tolerance t_1 form a subset S_i. The process repeatedly performs the random initialization, compares the obtained subsets and stores the largest subset S_i. The final model of the circle is calculated using the least squares method based on the points in the largest subset. During the experiments $t_1 = 0.4$ was used and the dataset consisted of individual edge segments. Since straight lines can also be identified as circles if the radius is infinite, the radii of the circles and circle segments were limited to half of the image height. A general RANSAC implementation [Kovesi, 2000] was adapted to find circular shapes.

4.8.2 Result

The result from the evaluation of the different virtual sensors and image sets is presented in Table 4.12. Test 1-8 refers to tests with the original 24 features and Test 9-16 to the extended feature set. The rightmost column shows the normalized accuracy (Φ_{Nacc}), which is the expected accuracy when using equally large positive and negative evaluation sets.

Result for Set C From Table 4.12 it can be noted (comparing test 1-4 with 9-12) that the results for both Φ_{Acc} and Φ_{Nacc} are better for three out of four virtual sensors when using the feature set with 28 features. A small degradation is seen for buildings, but more importantly, a notable improvement is seen for the truck virtual sensor. The true positive rate has in fact decreased, but the true negative rate has improved more, resulting in an increase of Φ_{Acc} from 81.5% to 87.4%. For test 9-12 Φ_{Nacc} varies from 86% for buildings up to almost 99% for nature.

Result for Set D Set D includes images collected from the Internet. It was noted in previous sections that the sets of Internet images are not as consistent as the images taken manually with digital cameras. This is clearly reflected in the decreased accuracy for buildings, nature and trucks. The set of window images does not include images from the Internet and for Test 5 and 13 the

Test	VS	Set	Φ_{TP} [%]	Φ_{TN} [%]	Φ_{Acc} [%]	Φ_{Nacc} [%]
1	Windows	C	92.9	85.3	88.4 ± 5.8	89.1
2	Buildings	C	81.6	91.6	89.6 ± 5.8	86.6
3	Nature	C	96.6	98.3	98.0 ± 2.8	97.5
4	Trucks	C	91.6	79.0	81.5 ± 7.8	85.3
5	Windows	D	94.1	87.8	89.4 ± 4.8	91.0
6	Buildings	D	73.3	82.8	80.4 ± 5.7	78.0
7	Nature	D	96.1	95.9	96.0 ± 2.8	96.0
8	Trucks	D	83.9	75.3	77.4 ± 5.6	79.6
9	Windows	C	93.2	85.7	88.7 ± 5.5	89.4
10	Buildings	C	81.4	91.0	89.1 ± 5.4	86.2
11	Nature	C	98.4	99.0	98.9 ± 2.0	98.7
12	Trucks	C	89.6	86.8	87.4 ± 7.1	88.2
13	Windows	D	94.6	87.9	89.6 ± 4.5	91.3
14	Buildings	D	78.9	81.9	81.2 ± 6.5	80.4
15	Nature	D	96.0	95.6	95.7 ± 3.2	95.8
16	Trucks	D	79.8	81.4	81.0 ± 5.5	80.6

Table 4.12: Results from the evaluation of the four virtual sensors for 100 runs. *Test* 1-8 use the original 24 features and *Test* 9-16 use the extended feature set with 28 features. Φ_{TP} is the true positive rate, Φ_{TN} is the true negative rate, Φ_{Acc} is the accuracy, and Φ_{Nacc} is the normalized accuracy.

accuracy has in fact increased slightly because the window images in Set D constitute a relatively more homogeneous group of images compared to Set C.

Additional Features A few features intended to capture properties of trucks were added to the feature set and the results are shown in Table 4.12, Test 9 to 16. For the truck VS the positive detection rate actually decreased, but the accuracy was increased due to that the true negative rate increased more than the true positive rate decreased. In Set C, the values of Φ_{Acc} are similar for all virtual sensors except for the truck virtual sensor. Here the value has increased from 81.5% to 87.4% and it can therefore be concluded that the use of the additional features actually improved the overall result.

Comparing the results in the previous section, Table 4.9 with the result in Table 4.12 it can be seen that the introduction of the set of truck images did

not introduce any major differences to the overall result for the three other virtual sensors.

To better understand which classes cause problems in the different tests, the classification results have been separated. Table 4.13 shows the corresponding figures. The first three columns are the same as in Table 4.12. Next, there are four columns with data extracted from the experiments. One cell on each row refers to the true positive detection rate (the diagonals), and the remaining three cells show the false positive rates for the respective classes in the negative dataset. The diagonals in the tables should contain high numbers and the other cells should contain low numbers.

From the table it can be noted that the lowest numbers (low false positive rate) occur for nature images classified by the window virtual sensor and for window images classified by the nature virtual sensor. This indicates that windows and nature are well separated by the selected features. One can also note that the introduction of additional features decreased all false positive rates for the truck virtual sensor (cf. Test 4 with 12 and Test 8 with 16).

Test	VS	Set	Wind. [%]	Build. [%]	Nat. [%]	Truck [%]
1	Windows	C	92.9	29.4	0.0	14.6
2	Buildings	C	13.9	81.6	1.8	4.0
3	Nature	C	0.4	2.4	96.6	3.4
4	Trucks	C	16.8	18.2	32.2	91.6
5	Windows	D	94.1	21.1	0.8	14.7
6	Buildings	D	24.5	73.3	9.1	18.1
7	Nature	D	1.9	5.2	96.1	5.1
8	Trucks	D	26.1	32.1	16.0	83.9
9	Windows	C	93.2	28.6	0.0	14.4
10	Buildings	C	13.7	81.4	1.6	6.8
11	Nature	C	0.0	1.8	98.4	2.0
12	Trucks	C	9.9	13.6	19.2	89.6
13	Windows	D	94.6	21.2	0.8	14.2
14	Buildings	D	25.5	78.9	9.1	19.6
15	Nature	D	1.7	5.2	96.0	6.4
16	Trucks	D	16.6	24.0	15.1	79.8

Table 4.13: Results from the evaluation of the four virtual sensors (windows, building, nature, and trucks), using 100 runs with the original 24 features (Test 1-8) and the extended feature set (Test 9-16), separated into the four classes.

In this section a virtual sensor for trucks was introduced and an evaluation of the virtual sensors for windows, building, nature, and trucks was performed. Additional features that notably improved the performance of the truck class (which was the intention) were introduced, but also the overall classification result was improved. For Set C, images taken by the handheld cameras, high accuracies from 87.4% up to 98.9% were achieved.

4.9 Summary and Conclusions

This chapter introduced the concept of a virtual sensor and presented instances of virtual sensors for four classes: *windows*, *buildings*, *nature*, and *trucks*. These virtual sensors use vision to classify the view and were trained using categorized images (supervised learning). Virtual sensors relate the robot sensor readings to human spatial concepts and are applicable, for example, when semantic information is necessary for communication between robots and humans.

First, two classifiers intended for use on a mobile robot to discriminate buildings from nature were evaluated. The results from the evaluation show that high classification rates can be achieved, and that Bayes classifier and AdaBoost give similar classification performance in the majority of the performed tests. The number of wrongly classified images is reduced by about 50% when the higher resolution images are used. The features that were used have scale invariant properties, demonstrated by the cross resolution test where the classifier was trained with one image size and tested on another size. The benefits gained from using AdaBoost include the highlighting of strong features and its improved generalization properties over the Bayes classifier. The tests also revealed that histogram of edge orientation is the best single feature in the feature set for finding building images.

To show the ability to learn other virtual sensors using the defined feature set, virtual sensors for windows and trucks were also learned and evaluated. Without extension of the feature set, they performed very well, although the performance of the virtual sensor for buildings degraded slightly. To even better distinguish between similar concepts, features that capture specific characteristic properties may be added to the feature set. This was shown for the virtual sensor dedicated to detection of trucks. With a few new features, the result for the accuracy of truck detection was notably improved.

It was also shown how a virtual sensor can be used for pointing out buildings along the trajectory of a mobile robot. In the performed experiment it turned out that the feature set could also handle seasonal changes.

The suggested method using machine learning and generic image features makes it possible to extend virtual sensors to a range of other important human spatial concepts.

Chapter 5
Probabilistic Semantic Mapping

5.1 Introduction

In Chapter 3 the importance of semantic information and the possibilities that it gives were discussed. It was noted that semantic information can, for instance, facilitate human-robot interaction (HRI) but several other application fields were also presented. A method to extract semantic information was presented in Chapter 4, where a virtual sensor that relates sensor readings to human spatial concepts was introduced. As one example, a virtual sensor for building detection using methods for classification of views as buildings or nature based on vision was described. The purpose was to detect one very distinctive type of object that is often used by humans, for example, in textual description of route directions.

A semantic map is an additional tool to represent semantic information and can be used, for instance, to improve HRI. This chapter introduces a method that fuses data from different sensor modalities, range sensors and vision sensors, to create a semantic map of the environment. The method was applied to an outdoor environment, but it is expected to also apply to indoor environments with appropriate classifiers for the desired classes of objects to be mapped. The introduced method for probabilistic semantic mapping combines information from a learned virtual sensor with a standard occupancy grid map. Here, this will be exemplified by using the learned virtual sensor for building detection presented in Chapter 4. Pose information (location and orientation) from the mobile robot together with the output from the virtual sensor is used to estimate the directions to detected buildings. These directions are used to update the occupancy grid map with semantic information. The result is a semantic map with two classes: 'buildings' and 'nonbuildings'. A probabilistic approach is applied in order to cope with uncertainty in the output from the virtual sensor, both uncertainty in the classification and uncertainty regarding which part of the image caused the classification.

5.1.1 Outline

Section 5.2 describes the process of building a probabilistic semantic map. Images taken by a mobile robot are fed into a virtual sensor learned for a specific class. Based on the result from the virtual sensor and the robot pose, local maps showing connected regions[1] representing the detected class are created for each robot position where images have been acquired. The local maps are then fused into a global probabilistic semantic map.

For the experiments an omni-directional camera mounted on the mobile robot giving a 360° view of the surroundings was used. From the omni-directional image N_{pv} planar views are created with a horizontal field-of-view of Δ degrees (the values of N_{pv} and Δ are provided in Table 5.2). These planar images are used as input to the virtual sensor. The process of calculating the planar views and descriptions of the experiments are given in Section 5.3. An evaluation of the results is given in Section 5.4, followed by a discussion in Section 5.5 that concludes this chapter.

5.2 Probabilistic Semantic Map

In this section a probabilistic approach to semantic mapping is presented. The objective is to take an occupancy grid map and introduce semantic labelling of the occupied cells. The semantic information comes from the output of a learned virtual sensor. It is assumed that an occupancy grid map including occupied cells that represent objects of the class of interest exists. An occupancy grid map can be built using a laser scanner or stereo vision for example.

The result from the virtual sensor applies for the whole input image. This means that all objects within the view are assumed to belong to the same class. To focus the attention on the main objects in the view, probabilities assigned for single objects are adjusted according to their proportions of the view, see Section 5.2.1.

The process of building the semantic map is divided into two steps. In the first step connected regions that are within view of the virtual sensor are searched in the grid map, and a local semantic map is created around the robot. In the second step the local maps are used to update a global map using a probabilistic method. The result in this case is a global semantic map where connected regions for buildings and nature can be distinguished.

Both the local maps and the global map are grid maps of the same size as the occupancy grid map and with the same cell size. Each cell can have a value P_{cell} in the interval $[0, 1]$. The maps are initialised with all cells set to *unknown* ($P_{cell} = 0.5$) and are then incrementally updated as the robot travels along the trajectory and evaluates the views with the virtual sensor. In this

[1] A *connected region* is understood as a connected component in a binarized occupancy grid representing the real object that caused the occupied area in the map.

chapter the virtual sensor for building detection is used and the resulting map will then contain the three following classes; *buildings* ($P_{cell} > 0.5$), *nature* ($P_{cell} < 0.5$) and *unknown* ($P_{cell} = 0.5$). Empty and unknown cells in the original occupancy grid map are not affected by the mapping, only the occupied cells are considered.

5.2.1 Local Semantic Map

Throughout this chapter it is assumed that an occupancy map has been built or is built during the mobile robot motion. The local semantic map is a probabilistic representation of a sector in the occupancy map as seen by the robot. The sector is defined by the robot pose, the direction of the virtual planar camera (Section 5.3.1), the opening angle θ of the sector (related to the classified camera image) and the expected maximum range L_{VS} of the virtual sensor, VS. The horizontal covering angles $\{\alpha_i\} = \alpha_1, \alpha_2, \ldots, \alpha_n$ of all connected regions within this sector are calculated. A sector is illustrated in Figure 5.1.

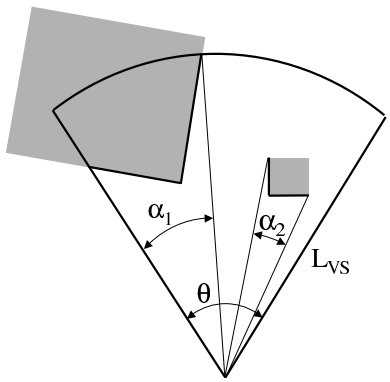


Figure 5.1: Illustration of a sector with an opening angle θ, representing the view of the virtual sensor. Two connected regions are found (the grey rectangles) within the sector and their respective sizes are represented by α_1 and α_2.

The sum of the horizontal covering angles cannot exceed the value of the sector opening angle θ:

$$\sum_{i=1}^{n} \alpha_i \leq \theta \tag{5.1}$$

The outlines of the connected regions found in the sector constitute the local map. These regions shall be assigned a probability based on the classification performed by the virtual sensor. As previously mentioned it is assumed that

larger objects are more likely to affect the virtual sensor. The larger regions in view are therefore given a higher probability than smaller regions.

Probabilities $P_i(\mathit{class}|\mathrm{VS}^{T_s}, \alpha_i)$ are assigned to the n regions in view (defined by the sector) in relation to their perceived size, measured by the horizontal covering angles α_i, using the following expression:

$$P_i(\mathit{class}|\mathrm{VS}^{T_s}, \alpha_i) = \frac{1}{2} + \frac{\alpha_i}{\theta}(P(\mathit{class}|\mathrm{VS}^{T_s}) - \frac{1}{2}) \tag{5.2}$$

where $P(\mathit{class}|\mathrm{VS}^{T_s})$ is the conditional probability that a view is *class* when the virtual sensor classification at sensor reading T_s is *class*. With Equation 5.2, the probabilities $P_i(\mathit{class}|\mathrm{VS}^{T_s}, \alpha_i)$ are assigned a value that is within the interval 0 to 1 and proportional to the perceived region sizes and $P(\mathit{class}|\mathrm{VS}^{T_s})$.

$P(\mathit{class}|\mathrm{VS}^{T_s})$ is assigned different values depending on the output of the VS. The two combinations that are interesting for calculating $P(\mathit{class}|\mathrm{VS}^{T_s})$, here examplified with *buildings* and *nonbuildings*, are:

$$P(\mathit{class}|\mathrm{VS}^{T_s}) = \begin{cases} P(\text{build}|\text{VS=build}) & > 0.5 \\ P(\text{build}|\text{VS=}\neg\text{build}) & < 0.5 \end{cases} \tag{5.3}$$

With this setting $P_i(\mathit{class}|\mathrm{VS}^{T_s}, \alpha_i)$ will always be assigned a value larger than 0.5 when the virtual sensor detects a building, and a value below 0.5 when a view is classified as a nonbuilding.

One problem in the practical use of the local semantic map is the selection of the parameters $P(\mathit{class}|\mathrm{VS}^{T_s})$. In situations where the performance of the virtual sensor is well known, the parameters can be determined based on this performance. The internal relation of the parameters should then reflect the probability of false readings in order to handle these in the best way. This approach has not been investigated due to reasons described in Section 5.3.4. Instead, in the experiments described further on in this chapter the parameters in Equation 5.3 were learned in a test run with the mobile robot.

5.2.2 Global Semantic Map

The second step deals with the global semantic map, which is updated whenever a new local semantic map is available. The standard Bayes update equation (as described in, e.g., Thrun et al. [1998]) is used to update the global semantic map with the local semantic map produced in the previous step. In the following it is shown how an individual grid cell at position (x, y) is updated based on a technique for updating occupancy grid maps.

The probability that grid cell (x, y) is occupied after T_s sensor readings is denoted by $P(\mathrm{occ}_{x,y}|s^1, s^2, \ldots, s^{T_s})$ where s^i denotes a sensor reading. Assuming that the conditional probability $P(s^{(t)}|\mathrm{occ}_{x,y})$ is independent of $P(s^{(\tau)}|\mathrm{occ}_{x,y})$ if $t \neq \tau$ (known as the Markov property), the probability at (x, y) can be computed as:

$$P(\text{occ}_{x,y}|s^{1:T_s}) = 1 - \left(1 + \frac{P_{prior}}{1-P_{prior}} \prod_{r=1}^{T_s} \frac{P(\text{occ}_{x,y}|s^{(r)})}{1-P(\text{occ}_{x,y}|s^{(r)})} \frac{1-P_{prior}}{P_{prior}}\right)^{-1}. \tag{5.4}$$

A second assumption that the prior probability for occupancy, P_{prior}, can be set to 0.5 simplifies Equation 5.4 to:

$$P(\text{occ}_{x,y}|s^{1:T_s}) = 1 - \left(1 + \prod_{r=1}^{T_s} \frac{P(\text{occ}_{x,y}|s^{(r)})}{1-P(\text{occ}_{x,y}|s^{(r)})}\right)^{-1}. \tag{5.5}$$

This equation can be rewritten to a recursive update formula. From Equation 5.5 we can write

$$\prod_{r=1}^{T_S-1} \frac{P(\text{occ}_{x,y}|s^{(r)})}{1-P(\text{occ}_{x,y}|s^{(r)})} = \left(1 - P(\text{occ}_{x,y}|s^{1:T_s-1})\right)^{-1} - 1 = \frac{P(\text{occ}_{x,y}|s^{1:T_s-1})}{1-P(\text{occ}_{x,y}|s^{1:T_s-1})} \tag{5.6}$$

Substituting Equation 5.6 in Equation 5.5 results in the recursive update formula

$$P(\text{occ}_{x,y}|s^{1:T_s}) = 1 - \left(1 + \frac{P(\text{occ}_{x,y}|s^{1:T_s-1})}{1-P(\text{occ}_{x,y}|s^{1:T_s-1})} \frac{P(\text{occ}_{x,y}|s^{T_s})}{1-P(\text{occ}_{x,y}|s^{T_s})}\right)^{-1}. \tag{5.7}$$

In our case the sensor reading s^{T_s} is the output VS^{T_s} from the virtual sensor at sensor reading T_s and the grid cells are assigned a probability denoting whether they belong to *class*. Using these notations Equation 5.7 is rewritten as:

$$P(class|\text{VS}^{1:T_s}) = 1 - \left(1 + \frac{P(class|\text{VS}^{1:T_s-1})}{1-P(class|\text{VS}^{1:T_s-1})} \frac{P(class|\text{VS}^{T_s})}{1-P(class|\text{VS}^{T_s})}\right)^{-1} \tag{5.8}$$

which is the update formula used for the grid cells of the global semantic map (the grid cell index (x, y) has been left out). The resulting global semantic map will contain three different classes:

$$\begin{array}{lll} \textit{Building} & \text{if} & P(class|\text{VS}^{1:T_s}) > 0.5 \\ \textit{Unknown} & \text{if} & P(class|\text{VS}^{1:T_s}) = 0.5 \\ \textit{Nonbuilding} & \text{if} & P(class|\text{VS}^{1:T_s}) < 0.5 \end{array} \tag{5.9}$$

where the degree of certainty for *Building* is higher close to 1 and the degree of certainty for *Nonbuilding* is higher close to 0.

5.3 Experiments

In this section the experiments that have been performed to validate the above presented method are described. The semantic information comes from the virtual sensor that was trained using AdaBoost. This training was performed using images with 240 pixels side length, images taken from image set 1 defined in Section 4.5.1. In the evaluation of the method, images from virtual planar cameras were used as input to the virtual sensor. The virtual planar cameras are described in Section 5.3.1.

The two datasets that were used in the experiment are described in Section 5.3.2. The calibration of the parameters was based on the first of these datasets. The validation was performed using three different occupancy grid maps, see Section 5.3.3. The calibration is described in Section 5.3.4.

5.3.1 Virtual Planar Cameras

In Chapter 4, a digital camera mounted on a pan-tilt head was used to create a panoramic view with approximately 280° horizontal field-of-view. A sweep of the pan-tilt head requires a few seconds, making the data acquisition time-consuming. In order to acquire a 360° field-of-view panoramic image in one single shot, an omni-directional camera is used here. As an additional benefit, the centre of the image plane is now the same throughout the whole panoramic view.

The robot used in the experiments, a Pioneer P3-AT from ActivMedia, is fitted with an omni-directional camera. The camera is a standard consumer-grade SLR digital camera (Canon EOS350D, 8 megapixels). On top of the lens, a curved mirror from 0-360.com is mounted. Further details on the equipment are given in Section 2.2.

The camera-mirror combination produces omni-directional images that can be unwrapped into high-resolution *spherical images* (also referred to as panoramic images) by a polar-to-Cartesian conversion. From a large spherical image, smaller planar images are extracted, using projections, as they would appear for a regular camera. Figure 5.2 shows an example of the omni-directional image, the unwrapped image, and some planar images. For details on camera projections, see for example Hartley and Zisserman [2004].

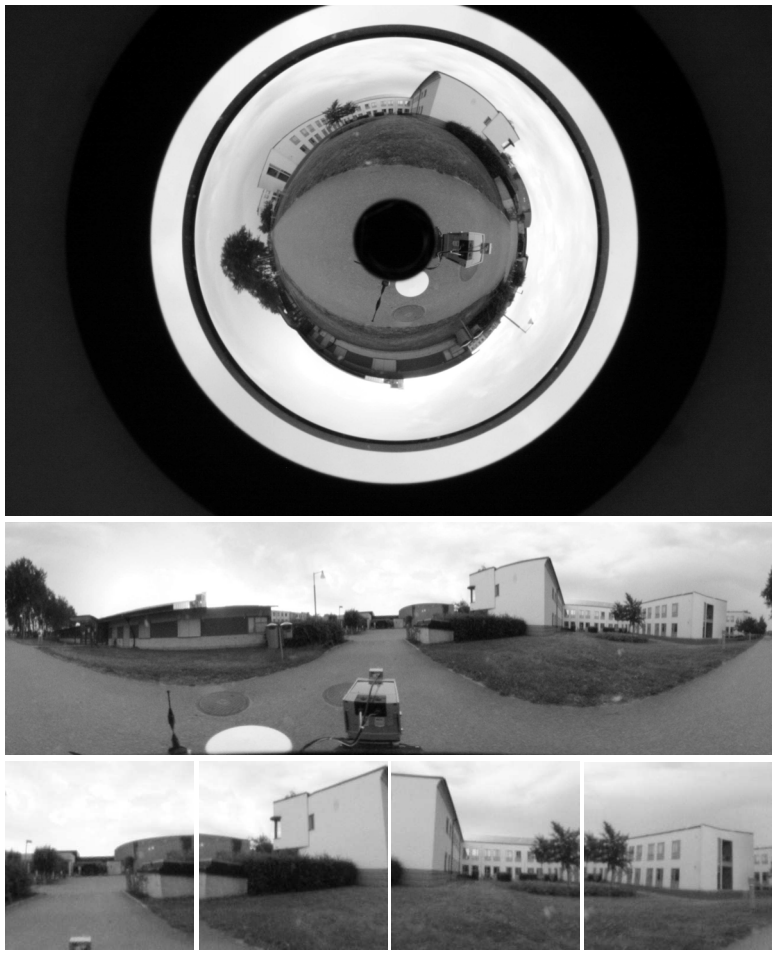


Figure 5.2: The upper part of the figure is one of the omni-directional images used in the experiments, the middle part is an unwrapped version of the same image, and the lower part shows some planar images extracted from the unwrapped image. One can note that the omni-directional image only uses about 30% of the available 8 megapixels, giving that the effective number of pixels in the omni-image is about 2.4 megapixels. The unwrapped image has 2.2 megapixels.

5.3.2 Image Datasets

Two datasets are used for the experiments described in this chapter, see Table 5.1. The datasets consist of omni-images and pose information acquired by the mobile robot. Each omni-image was converted into eight planar images, where each planar image has a resolution of 320×320 pixels and covers a horizontal and vertical field-of-view of $56°$. This means that there is a small overlap between planar images generated from the same omni-image. This overlap was introduced in order to reduce the probability that information on the image border would not be taken fully into account. Examples of the planar images are given in the lower part of Figure 5.2. In total 2384 planar images were used for the experiments. The images were collected at Örebro Campus and the mobile robot trajectories are shown in Figure 5.3. Differential GPS (DGPS) and odometry were used to compute the robot poses (position and orientation) along the trajectory, see Appendix B.3. Other alternatives for pose estimation, such as SLAM (Simultaneous Localization and Mapping), could also have been used.

Figure 5.3: The figure shows the trajectories for the two datasets used for training (Set 1) and evaluation (Set 2) respectively. Set 1 is the right trajectory (dashed) and Set 2 is the left trajectory (solid). The starting points are marked with circles. (Aerial image ©Örebro Community Planning Office.)

Set	Omni-images	Planar images	Length
1	88	704	146 m
2	210	1680	317 m

Table 5.1: Image datasets used for the semantic mapping experiments.

5.3.3 Occupancy Maps

The evaluation part of the semantic mapping experiment was repeated with three different occupancy maps. These maps cover parts of the area shown in Figure 5.3. The first map is a handmade occupancy map. Using the aerial image presented in Figure 5.3, the occupancy map shown in Figure 5.4 was constructed. The building outlines and groups of trees around the trajectory have been marked as filled polygons by hand and this occupancy map is binary, i.e., a grid cell is either empty or occupied.

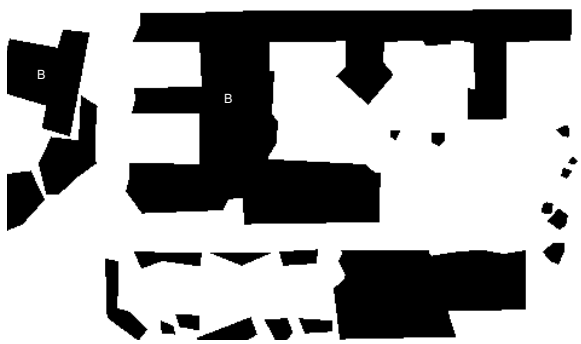
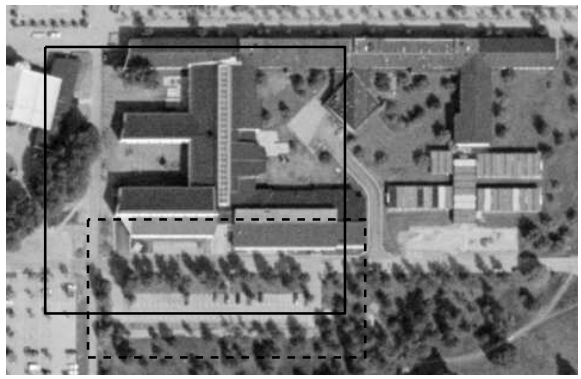


Figure 5.4: The handmade occupancy map. The two building regions are marked with a 'B'. All other regions are nonbuildings. This map with the labels also serves as the ground truth in the evaluation.

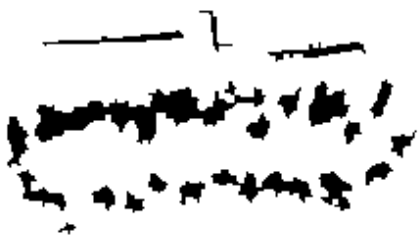
The second map covers the last two thirds of Set 2, see Figure 5.5(a). This map is created using 2D laser readings from the SICK laser range finder, horizontally mounted in front of the robot, see Section 2.2. The resulting map is shown in Figure 5.5(b). The reason why the first third of the dataset was not used is that it was not feasible to create a consistent map from the 2D laser range data since the robot trajectory included elevated terrain.

The third map is based on 3D-scans obtained by tilting the laser range scanner. In this case only readings with a minimum height of 2 meters were considered in the occupancy map. This eliminates some cars and bushes. The resulting



(a) The area of the 2D-map (solid) and the 3D-map (dashed).

(b) The occupancy map based on measurements using a 2D-laser range finder.



(c) The occupancy map based on measurements using a 3D-laser range finder.

Figure 5.5: The two occupancy maps based on 2D- and 3D-laser range finders.

occupancy map, shown in Figure 5.5(c), is based on data from the first half of Set 2 (the part where 3D-data were collected).

In the experiments, binarized occupancy grid maps were used, where either an empty grid cell (0) or an occupied grid cell (1) is represented. However, probabilistic occupancy maps could also be used directly with the suggested algorithm. Since it is assumed that the grid cells in the occupancy map are independent from the output of the virtual sensor, the values of the grid cells in the occupancy map can be seen as another sensor reading and the probabilistic semantic map is updated with an expression corresponding to Equation 5.8. Provided that nonbinarized occupancy grid maps are used, an alternative procedure for definition of objects in the sector has to be defined.

5.3.4 Used Parameters

Table 5.2 lists the important parameters for constructing the planar camera views and the probabilistic semantic maps. These parameters can be adjusted to change the desired properties of the system, for instance:

- P(build|VS=build) versus P(build|VS=¬build) is related to the performance of the virtual sensor.
- The sector opening angle θ should be related to the planar camera field of view. By decreasing θ the result from the virtual sensor focuses on the central parts of the planar image.

In our work the last four parameters (planar camera field-of-view - grid cell size) were set according to Table 5.2. The setting of the first three parameters (the sector opening angle θ and the probabilities P(build|VS=build) and P(build|VS=¬build)) have been varied in order to optimize the performance of the system. It would be preferable to be able to relate P(build|VS=build) and P(build|VS=¬build) directly to the classification rate of the virtual sensor. However, in reality there are a lot of views that contain a mix of buildings and nature that make a proper ground truth evaluation difficult. It was therefore decided to set the parameters based on the evaluation of the performance of the complete system including the virtual sensor and the map building algorithm. Set 1 was used for learning the first three parameters in Table 5.2 and these parameters were then evaluated using Set 2 in Section 5.4.

In total 21 combinations of different field-of-views θ and probability pairs were evaluated in order to find a good combination of parameters. This evaluation was performed using dataset 1. The following three θ were used: $56°, 45°$, and $30°$ and the following seven probability pairs (P(b|VS=b), P(b|VS=¬b)): (0.8, 0.3), (0.8, 0.4), (0.8, 0.45), (0.9, 0.3), (0.9, 0.4), (0.9, 0.45), and (0.95, 0.48). With θ=$56°$ the field-of-view is the same as for the virtual sensor. With

Parameter	Value	Description
P(build|VS=build)	learned (> 0.5)	Building probability
P(build|VS=¬build)	learned (< 0.5)	1 - Nature (nonbuilding) probability
θ	learned (30-56)	Sector opening angle [deg]
Δ	56	Planar camera field-of-view [deg]
N_{pv}	8	Number of planar views
L_{VS}	50	VS maximum range [m]
-	0.5	Grid cell size [m]

Table 5.2: Description of parameters used as settings for the planar camera views, sector size and the probability maps.

θ=45° there is no overlap in the local maps belonging to the same position. Using θ=30° it was intended to see how the system works when only the centre of the virtual sensor's field-of-view is used and the border parts are neglected.

One can note that P(build|VS=build) was selected to be always proportionally larger than P(build|VS=¬build). There are several reasons that motivate this asymmetry. The main reason is that the virtual sensor for building classification produces substantially more false negatives than false positives (see Section 4.5.1). A second reason is that it is more common with visible nature in front of buildings than vice versa. A third reason discovered during the experiments is that open views often were classified as nonbuildings. This affects the result since as the robot drives along a straight road, both the forward and backward looking views were classified as nature. The problem with this was that parts of the building close to the road were included in these sectors giving many false updates. Accordingly, we tend to have little confidence in classifications as nonbuilding, expressed by a value of P(build|VS=¬build) close to 0.5.

For each parameter combination the following measures were calculated:

- The true positive building detection rate, Φ_{TP}. Number of cells correctly classified as building / number of building cells included in the sectors.
- The true negative detection rate, Φ_{TN}. Number of cells correctly classified as nature / number of nature cells included in the sectors.

The measures were calculated based on the final global semantic map and use the sum of the true rates ($\Sigma_T = \Phi_{TP} + \Phi_{TN}$) as the primary selection criterion. Combination 6 (θ = 56°, P(build|VS=build) = 0.9, P(build|VS=¬build) = 0.45)

Test	θ [°]	P(b|VS=b)	P(b|VS=¬b)	Φ_{TP} [%]	Φ_{TN} [%]	Σ_T [%]
1	56	0.80	0.30	33.1	96.0	129.1
2	56	0.80	0.40	63.8	91.0	154.8
3	56	0.80	0.45	76.8	88.2	165.0
4	56	0.90	0.30	52.0	94.2	146.2
5	56	0.90	0.40	71.2	90.3	161.5
6	**56**	**0.90**	**0.45**	80.2	86.6	**166.8**
7	56	0.95	0.48	90.1	73.4	163.5
8	45	0.80	0.30	26.5	97.1	122.6
9	45	0.80	0.40	54.9	91.8	146.7
10	45	0.80	0.45	71.3	88.0	159.3
11	45	0.90	0.30	44.5	94.8	139.3
12	45	0.90	0.40	65.0	90.5	155.5
13	**45**	**0.90**	**0.45**	80.8	87.2	**168.0**
14	45	0.95	0.48	88.3	76.7	165.0
15	30	0.80	0.30	22.9	95.6	118.5
16	30	0.80	0.40	46.5	91.4	137.9
17	30	0.80	0.45	73.8	89.5	163.3
18	30	0.90	0.30	35.9	93.6	129.5
19	30	0.90	0.40	69.1	90.0	159.1
20	30	0.90	0.45	75.4	88.6	164.0
21	30	0.95	0.48	81.1	79.4	160.5

Table 5.3: Parameters and detection rates for Set 1. The letter b in P(b|VS=b) is short for building.

and combination 13 (θ=45°, P(build|VS=build) = 0.9, P(build|VS=¬build) = 0.45) resulted in the highest detection rate, see Table 5.3, and hence, these combinations result in the lowest total false rate.

5.4 Result

We use Set 2 to evaluate the semantic map and present the result for the two best parameter settings (combination 6 and 13) as found in the previous section.

5.4.1 Evaluation of the Handmade Map

The semantic map based on the handmade occupancy map and data from Set 2 (combination 6) is presented in Figure 5.6. It can be noted that most of the outlines of the connected regions were correctly labelled. Small parts that are not correct are the rightmost part of the building (marked with 'a') and a part of a grove close to the left building (marked with 'b'). At 'a' there is a conifer in front of the building and at 'b' the view shows a building behind some tree trunks. Both these two problems originate from scenes including a mixture of building and vegetation.

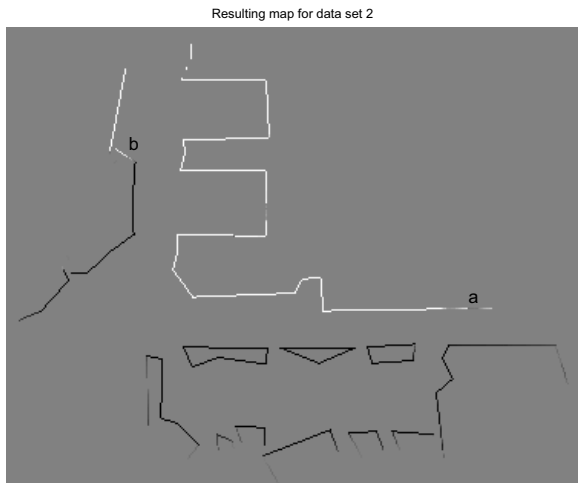


Figure 5.6: The resulting map using dataset 2. The outlines of both building and nature regions are correct to a large extent.

Table 5.4 presents the detection rates for Set 2. The first row shows the result for combination 6 and the second row for combination 13. The evaluation was performed based on all cells in the grid map that are not equal to 0.5. The true detection rates are all equal to or higher than 96.9% and combination 6 gives a slightly better result than combination 13 (it was the other way around for Set 1).

5.4.2 Evaluation of the Laser-Based Maps

The semantic maps created using the occupancy maps obtained from the 2D and 3D laser range measurements are shown in Figures 5.7 and 5.8. From a qualitative visual inspection, one can note that the main problem in the 2D-version is that the occupancy map contains connected regions originating from

Test	Φ_{TP} [%]	Φ_{TN} [%]	Φ_{FP} [%]	Φ_{FN} [%]
Handmade (6)	98.3	98.7	1.3	1.7
Handmade (13)	96.9	98.4	1.6	3.1
Laser 2D (6)	87.1	73.1	26.9	12.9
Laser 2D (13)	88.5	72.8	27.2	11.5
Laser 3D (6)	94.1	95.0	5.0	5.9
Laser 3D (13)	96.4	97.3	2.7	3.6

Table 5.4: Results for parameter combination 6 and 13 using dataset 2 with 3 different occupancy maps. The columns contain the true positive, the true negative, the false positive, and the false negative rates.

objects with low height, indicated by the ellipses in Figure 5.7. Since the images from the camera include also buildings behind these objects, which are mainly bushes, the output from the virtual sensor indicates buildings. These bushes have therefore been classified as buildings in the semantic map, see the marked areas in Figure 5.7. These problems do not occur in the 3D-version. In the 2D map also some single small trees have been classified as buildings where there are buildings in the background of the image. This again shows the problem of mixtures of different classes in the image, in this case between small objects in front of a building.

Quantitatively, the results in Table 5.4 also show that the input of a 2D-laser based occupancy map does not deliver as high true positive rates as the handmade map and the 3D-laser based map. The trajectories only partly cover the same area, so the results are not fully comparable. Still, it can be noted that the false positive rate is around 27% for the test using the 2D-laser based map, while it is below 5% for the test using the 3D-laser based map. This difference originates from the low vegetation mentioned above.

5.4.3 Robustness Test

To evaluate the robustness of the system, two different Monte Carlo simulations [Metropolis and Ulam, 1949] were performed using the handmade occupancy map. First, the sensitivity to changes in robot pose was tested (pose noise) and second, the dependency on variations in the detection rate of the virtual sensor was evaluated (classification noise). The uncertainty is modelled with zero mean Gaussian noise defined by the standard deviation σ for the position, $\sigma_{pos} = 2$ m, and direction, $\sigma_{dir} = 5°$. The position uncertainty is approximately the accuracy of standard GPS. Table 5.5 shows the result for Monte Carlo simulations with 20 runs per test. The first two rows contain results after introducing additional pose uncertainty. The detection rates are

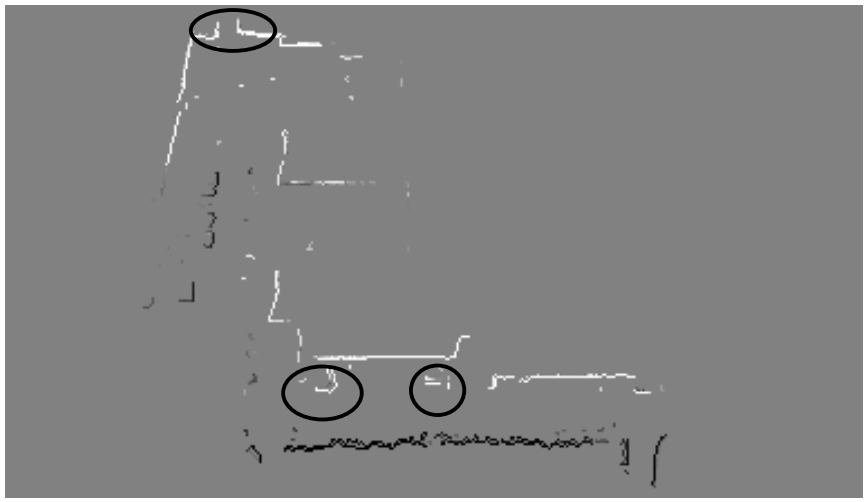


Figure 5.7: The semantic map based on the 2D-laser occupancy map. The ellipses mark bushes classified as buildings.

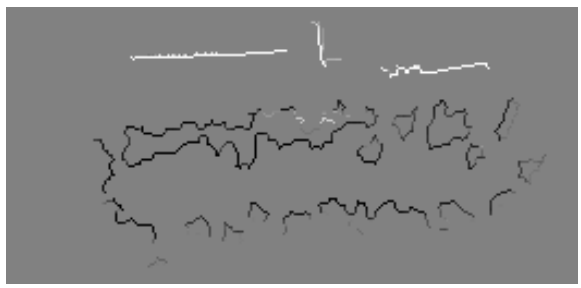


Figure 5.8: The semantic map based on the 2D-laser occupancy map.

slightly lower than the comparable ones presented in Table 5.4 (and repeated in Table 5.5). The total average detection rate[2] has decreased from 98.1% to 96.3%. A degradation is expected when errors are introduced and in this case, the pose errors resulted in a small degradation showing that the system have a robust behaviour.

The second two rows contain the result with classification noise. Here 5% of the classifications obtained from the virtual sensor were randomly changed (building to nature and vice versa). One can note that the result for building detection is close to the nominal case (average 97.0% compared to 97.6%), but

[2] Average of Φ_{TP} and Φ_{TN} for parameter combinations 6 and 13.

that nature detection is clearly affected by the changed detection rates of the virtual sensor (average 81.7% compared to 98.5%). This indicates that the probabilities P(build|VS=build) and P(build|VS=¬build) are crucial for the system. Introducing an additional error of 5% to the nature classifications results in a considerable drop of the correct nature classifications if this additional error is not reflected in the probability P(build|VS=¬build).

To demonstrate better handling of false nature classifications the tests with classification noise were repeated for parameter combinations 3 and 10. With the lower building probability (0.8) the Φ_{FP}-rates were halved, from 15% and 21% to 7% and 9% respectively.

Test	Φ_{TP} [%]	Φ_{TN} [%]	Φ_{FP} [%]	Φ_{FN} [%]
6 (pose noise)	95.8±3.6	97.5±1.0	2.5±1.0	4.2±3.6
13 (pose noise)	94.6±2.8	97.2±1.2	2.8±1.2	5.4±2.8
6 (classification noise)	97.6±1.2	84.7±3.7	15.3±3.7	2.4±1.2
13 (classification noise)	96.5±1.0	78.7±6.2	21.3±6.2	3.5±1.0
6 (nominal case)	98.3	98.7	1.3	1.7
13 (nominal case)	96.9	98.4	1.6	3.1
3 (classification noise)	95.8 ± 1.6	92.6 ± 4.2	7.4 ± 4.2	4.2 ± 1.6
10 (classification noise)	93.3 ± 2.5	90.5 ± 3.6	9.5 ± 3.6	6.7 ± 2.5

Table 5.5: Results for dataset 2. The first two rows show the results with pose uncertainty and the second two rows contain the result with classification noise. The results are presented with the standard deviation over 20 runs per test. The third two rows contain the nominal values taken from Table 5.4 and the fourth two rows include data for comparison with a second building probability.

5.5 Summary and Conclusions

In this chapter it was shown how a virtual sensor for pointing out buildings along the trajectory of a mobile robot can be used in the process of building a probabilistic semantic map of an outdoor environment. The presented results show that with the probabilistic mapping algorithm the uncertainty in the semantic labelling can be reduced. The method handles the wide field-of-view of the planar camera (56°) including objects that can belong to different classes. Despite the fact that the location of the classified object in the image is not known, an almost correct semantic map is produced. The map produced was found to be very robust in the presence of pose uncertainty both for buildings and nonbuildings. The experiments where classification noise was added

showed that a correct selection of prior probabilities is essential for the overall performance.

From the experiments described in this chapter, several benefits of using the virtual sensor with its good generalisation properties have been noted. The virtual sensor produces useful results even though:

1. The training set was quite limited. In total only 80 low resolution images with side length 240 pixels were used.

2. The resolution of the planar images (320×320 pixels) was different compared to the one used in the training phase.

3. The training was performed with images taken with a standard digital camera, but the images used during the experiment were planar images extracted from the omni-directional vision system onboard the mobile robot.

In the evaluation of the semantic map it was noted that one type of occupancy map did not produce as good results as the others. This was the occupancy grid map that was based on 2D-laser readings. Such a map, where the data have been collected in a horizontal plane close to the ground, is not optimal for finding building outlines. It is therefore recommended that objects represented by the occupancy grid map should have a minimum height, especially when the objective is to find objects of a particular height, such as buildings.

An extension to the semantic map presented here would be to refine the virtual sensor so that it can point out the parts of an image that are mainly responsible for the classification. This will further improve the separation between different objects. The problem with the separation was most evident in the case with the occupancy map based on 2D-laser, since that map included connected regions representing objects with a low height. For the handmade occupancy map and the 3D-laser-based map, where only readings above a height of 2 m were considered, this problem was not observed. To refine the virtual sensor, classification of sub-images could be used. An example of such a technique is described in [Morita et al., 2005] where small squares of the upper half of images are classified as tree, building and uniform regions. The result from the classifier was used for localization in previously visited areas.

This chapter described the construction of a probabilistic semantic map. Wide angle views have been used and the results were good. The objects observed in the experiments have a certain size which can explain why the wide angle views produced good results. If classes of smaller objects should be included in a semantic map, views with a smaller field-of-view should be used in order to better localize the objects of interest. An example of such a map would be to add information from the virtual sensor for windows.

Part III

Overhead-Based Semantic Mapping

Chapter 6
Building Detection in Aerial Imagery

6.1 Introduction

Detection of man-made structures, such as buildings, roads and vehicles, in aerial or satellite images has been an active research topic for many years. Aerial images, with their highly detailed contents, are an important source of information for applications including GIS, surveillance, etc.

This chapter discusses the extraction of man-made objects (buildings) from aerial imagery and gives examples of existing systems for automatic building detection. Without aiming at an exhaustive overview, the purpose is to give a background on extracting information from aerial images and to exemplify the problems when only monocular aerial images are used. The complexity of a system handling multiple view or stereo images is beyond our interest. Monocular images can be accessed more and more easily through the Internet and avoid complications that would arise from using stereo or multiple view images.

The content has been restricted to the fields that are of interest for our research, which include models, strategies, technologies and algorithms for extraction of buildings. Automatic detection of man-made structures is not yet a fully mature subject. Presented systems are often limited to certain types of images giving a strong dependence on a specific source of input data to obtain good performance.

Extraction of man-made structures from aerial images is a difficult task due to many reasons. Aerial images have a high level of structured and unstructured contents. Images differ in scale (resolution), sensor type, orientation, quality, dynamic range, light conditions and due to different weather and seasons. Buildings may have rather complicated structures and can be occluded by other buildings or vegetation. Together this makes building detection a challenging problem.

6.1.1 Outline

The chapter starts by introducing digital aerial imagery in Section 6.2. This includes aspects of image sensors, carriers, and manual feature detection, intended to facilitate the understanding of the extraction procedure and give ideas for improvement of existing implementations. Section 6.3 reviews methods for building detection in aerial images and the use of additional information, e.g., maps, to improve the performance is described. The chapter is concluded in Section 6.4.

6.2 Digital Aerial Imagery

The main information source in this work is a single digital aerial image from which buildings and other relevant structures should be extracted. Digital images can be divided in different ways, e.g., (a) panchromatic (monochromatic) and multi-chromatic, (b) low, medium and high resolution, and (c) stereo and mono.

Aerial images can be captured by cameras onboard satellites, spacecraft, aircraft and unmanned aerial vehicles (UAVs). Images may be of multi-band type including both the infrared (IR) and the visual wavebands. The use of aerial images for mapping is interesting for several reasons:

- Satellites cover and can photograph most interesting areas on the earth, even though there are some areas that are always cloudy and cities that are covered in smog most often.
- Satellite images are regularly updated.
- Aerial photos, for instance taken by UAVs, can give real-time coverage of an area.

6.2.1 Sensors

Our main interest is in aerial images taken with daylight cameras[1]. The reason for this is that daylight cameras are probably the most common type of image-based sensors. There are other image-based sensors that can provide more suitable information but these are not so common. In scientific applications sensors are often combined to enhance the operational use. Some properties of four types of sensors used for remote sensing onboard aerial vehicles are listed below:

Daylight cameras, usually monochrome or colour CCD cameras, give high resolution and high update rate. Monochrome cameras used to be most

[1]Daylight cameras should be understood as cameras that capture the visual wavebands.

common in existing systems since they traditionally gave higher resolution, but colour cameras are more frequently used in recent systems. Drawbacks of daylight cameras are their sensitivity to light conditions and reduced visibility.

IR cameras are often used in remote sensing tasks, for instance in detection of green vegetation. With wavelengths of typically 7–12 μm IR sensors operate independently of the light conditions and are well suited for, e.g., search operations.

Image radar, SAR (Synthetic Aperture Radar), can give high resolution 3D images and has all-weather capacity, but low update frequency.

Lidar, light detection and ranging, uses laser to give high resolution 3D images independent of light conditions. It gives high accuracy in depth measurement and airborne Lidar systems can cover large areas (altitudes from 1500 m up to 3500 m are possible) and are useful for distinction between buildings and the ground surface.

For frequent updating of aerial images, aircraft and UAVs are typically used as sensor carriers. The four types of image sensors listed above are available today for use onboard UAVs.

Additional information that can aid the extraction process may be provided by, for instance, elevation data, city maps, and GIS. Airborne laser range scanners have been used to build accurate 3D-models [Söderman et al., 2004] and SAR images, multi-spectral images, and high resolution satellite images have been used in building detection [Tupin and Roux, 2003, Xiao et al., 1998].

6.2.2 Resolution

The resolution of aerial and satellite images is often divided into 'low', 'medium', and 'high', where the resolution in meters per pixel for satellite images is: 30 m for low resolution, 10 m for medium resolution and 1 m for high resolution, and for aerial images: high resolution is below 0.2 m and low resolution is above 1 m pixel size. These numbers differ in the literature but the values given above indicate their normal sizes.

6.2.3 Manual Feature Extraction

When identifying features in an aerial image one has to consider that the photograph is taken from above with the result that the objects do not look familiar. Below, a number of factors [US Army, 2001] are listed that can be considered during the recognition phase. For identification of an object several of these factors have to be used.

Size The dimension of an object is important in the identification process. The size, either measured from the image scaling or by comparison with known objects in the image, can help to classify different types of objects.

Shape Shape can be used to separate man-made structures from natural ones. Man-made structures often have straight or smooth curved lines while natural structures are typically more irregular. For example, compare a canal and a river.

Shadows Shadows can be very helpful since they show a familiar view of the object, e.g., an antenna tower may be hard to detect in an image but its shadow is revealing. Shadows can also be used in height estimation of objects.

Shade Shade refers to the grey tone or texture (surface) of objects as they appear in the image and can, e.g., be used to find areas with surfacing in an even tone, such as asphalt.

Site An object can be recognized by its location in the image and its relation to surrounding objects. For example, the spacing between objects can reveal man-made influence in a natural environment.

From available papers on automatic extraction, shape, as defined by edges, seems to be the most popular feature in building detection. Techniques for use of shading are difficult to apply to real images [Lin, 1996].

6.3 Automatic Building Detection in Aerial Images

A survey by Mayer [Mayer, 1999] focuses on extraction of buildings. Seven systems developed between 1984 and 1998 were assessed according to a number of criteria. The author concentrated on models and strategies in this survey and the survey concludes that scale, context and 3D structure were the three most important features to consider for object extraction in aerial images.

In the following subsections a few systems that represent different approaches to automatic building detection are presented. They are divided into three groups based on the type of information that is used. The first group deals with systems that use monocular images (2D), the second group are systems with access to elevation information (3D) and the third group make use of maps or GIS for the detection of buildings in the aerial images.

6.3.1 Using 2D Information

A basic system for detection of buildings in monocular images taken from a nadir[2] view may include i) edge detection in a greyscale image ii) line determi-

[2] An image taken from a nadir view is taken from a zenith position. The opposite is an oblique view where the image is taken at an angle.

nation, and iii) search for buildings represented as ortho-polygonal lines [Cardoso, 1999].

In [Persson et al., 2005] we describe a system for automatic detection of buildings in aerial images, also taken from a nadir view. Our system builds two types of independent hypotheses based on the image content. A colour based segmentation process implemented with ESOM, an Ensemble of Self Organizing Maps, is trained and used to create a segmented image showing different types of roofs, vegetation and sea. A second type of hypotheses is based on an edge image produced from the aerial photo. Here, a line extraction process uses the edge image as input to find straight line segments that represent edges. From these edges, corners and rectangles that represent buildings are constructed. A classification process uses the information from both hypotheses to determine whether the rectangles belong to buildings, unsure buildings or unknown objects.

In the PhD thesis by C. Lin [Lin, 1996], generic 3D rectilinear models are used to model building parts. The system uses both wall and shadow information to verify hypotheses about modelled buildings. In a nadir image the shadow information is preferable to use while walls are hardly detectable, and for an oblique image the walls are more easily found while the shadows may be hard to use. The image angles therefore control how the verification process of the hypotheses weights the extracted information. Edges are identified with a direction, where the direction is dependent on the brightness on the respective side of the edge. In this way, parallel edges belonging to the same object can be connected. Shadows are detected by their darker appearance. Lin's system work well on images of scale models while real images pose problems due to the existence of vegetation, roads, and parking lots. The system for building detection from aerial images has been further improved with a user interface for interaction with the automatic system [Nevatia et al., 1997].

Fuzzy techniques have also been used in building detection. The FuzzBuRS [Levitt and Aghdasi, 2000], Fuzzy Building Recognition System, uses fuzzy variables to describe average region intensity, region size, average edge lengths and building likelihood. Roofs are often split into two halves due to different intensity and the system is limited to straight lines (as with many other systems) meaning that only polygon shaped buildings can be detected. The benefits of the system, compared with the authors' previous system working on crisp values, BuRS, are that the new system has a more compact representation and is easily understandable.

6.3.2 Using 3D Information

Matthieu Cord *et al.* presented a method for extraction and modelling of urban buildings [Cord et al., 1999]. By use of high resolution stereo images a Digital

Elevation Model (DEM[3]) is created. The elevation information is then used for classification of the global scene into buildings, ground surface and vegetation. This is followed by detailed modelling of the buildings. The authors believe that altitude is one of the most important sources of information for building detection. This information is necessary to separate buildings from other man-made structures, e.g., parking lots.

Many other authors also use depth information. In [Guo et al., 2001] the depth information is combined with a learning-based second step that corrects false positive detections of buildings. The authors use depth, colour, brightness, texture, and boundary energy[4] in a tree classifier. The depth classifier filters out ground objects and low vegetation. The colour and texture classifier filters out vegetation that has the same height as the buildings. Finally, a gradient field, based on a combination of image intensity and depth, is used to determine the size and orientation of the buildings.

6.3.3 Using Maps or GIS

Methods that use maps or GIS in the extraction process are interesting from our perspective since these have similarities to the approaches presented in Chapter 7 and 8. One example of such a method is the use of knowledge represented in digital topographic databases for improvement of automated image analysis regarding extraction of settlement areas [Schilling and Vögtle, 1997]. With the knowledge a model-driven top-down approach can be integrated into the commonly data-driven bottom-up process of satellite image analysis. Objects of the same class stored in the database are used to learn features in the satellite image, under the assumption that the majority of the objects are correct. These features are then used in a classification process for extraction of settlement areas.

Carroll presents a system for change detection, which is used in an application for information updating [Carroll, 2002]. The presented prototype, HouseDiff, combines GIS and edge detection. An edge detection method based on deformable contours (snakes) is used in the system to find the outline of new buildings. Such a method overcomes limitations due to the assumption of rectilinearity in previous works.

In another approach 3D building hypotheses of dense urban areas are generated using scanned maps[5] and aerial images [Roux and Maître, 1997]. The maps are analysed in order to obtain a structural description of the scene. This information is then used for the analysis of a disparity image generated from a stereo pair of aerial images. Histograms of the disparity image are calculated

[3]DEM include trees, buildings, the ground surface etc.

[4]The boundary energy is used to find an initial size and orientation of a rectangle template model. The best model is the one that maximizes the boundary energy, measured as the average gradient magnitude along model boundaries.

[5]Maps scanned by using an image scanner.

and processed. Buildings are extracted under the assumption that the disparity of one building is included in only one histogram mode. According to the authors, other approaches (not using maps) have been shown to perform well for sparse buildings, but not in dense urban areas. Two reasons that may explain this are the high image complexity and that there may be several different interpretations of the same object in the scene.

6.4 Summary and Conclusions

Automatic detection of man-made structures is not yet a fully mature subject. Developed systems are often limited to certain types of images giving a strong dependence on the type of input data for good performance. Many systems in this field use line and edge detection, form hypotheses, and connect them to 3D models for verification, while colour and texture are not used so often to extract features for the detection phase. Table 6.1 gives a summary of the systems discussed in this chapter.

This chapter has pointed out several important things. First, to be able to truly distinguish buildings from other man-made objects, information about the elevation of the area in the image is needed, which is why a number of the systems use Digital Elevation Models (DEM). The required elevation information can be obtained by airborne mounted sensors such as laser, radar, stereo vision or by systems operating on the ground, e.g., an outdoor mobile robot. Second, multiple view images give different aspect angles which can help an automated system in the detection phase. Third, an experience from our work [Persson et al., 2005] is that colour models of roofs in some cases overlap with other surfaces on the ground in the absence of elevation data. For instance, roads can be mixed up with gray roofs and tennis courts have the same colour as the red roofs.

References	Type of images	Comments
[Lin, 1996], [Nevatia et al., 1997]	Nadir or oblique, monocular	3D-models, works on modelboard images, problems with real images
[Levitt and Aghdasi, 2000]	Nadir, monocular, gray scale	2D-models, fuzzy approach
[Cord et al., 1999]	Nadir, stereo	3D-models, extracted from DEM built by the system
[Guo et al., 2001]	Nadir, stereo, colour	Depth, colour and brightness, texture, and boundary energy are used
[Carroll, 2002]	Nadir, monocular, colour	2D-models, searching for differences (HouseDiff)
[Schilling and Vögtle, 1997]	Nadir, monocular	Combination of GIS and satellite images
[Roux and Maître, 1997]	Nadir, stereo	Combination of GIS in the form of maps and aerial images, generates 3D building hypotheses
[Cardoso, 1999]	Nadir, monocular, gray scale	Extracts 2D building estimates, uses shape in the form of edge detection
[Persson et al., 2005]	Nadir, monocular, colour	Extracts 2D building estimates, uses both colour and shape, assumption of rectangular buildings

Table 6.1: Summary of building detection systems.

Chapter 7
Local Segmentation of Aerial Images

This chapter investigates the use of monocular aerial images to extend the sensory range of a mobile robot for outdoor mapping. The suggested method relates an aerial image to ground-level information using building outlines (wall estimates). This approach addresses two difficulties simultaneously:

1. buildings are hard to detect in monocular aerial images without elevation data and
2. the limitation that only those parts of the environment that are in line-of-sight of the sensors onboard the mobile robot can be perceived.

It is shown how wall estimates found by a mobile robot can compensate for the absence of elevation data in segmentation of aerial images. A virtual sensor for building detection mounted on a mobile robot is used in combination with an occupancy map to obtain wall estimates from a ground perspective. These wall estimates are matched with edges detected in an aerial image. The result is used to direct a region- and boundary-based segmentation algorithm for building detection in the aerial image. Experiments demonstrate that the ground-level based wall estimates can focus the segmentation of the aerial image to buildings and a semantic map, which covers a larger area than the onboard sensors, can be built along the robot trajectory.

7.1 Introduction

A mobile robot has a limited view of its environment. Mapping of the operational area is one way of enhancing this view for visited locations. In this chapter the possibility of using information extracted from aerial images to further improve the mapping process is explored. Semantic information about buildings is used as the link between ground level information and the aerial image. The method can speed up exploration or planning in areas not yet visited by the robot.

Colour image segmentation can be used to extract information about buildings from an aerial image. For directed image segmentation it is necessary to have an input that can point out the image regions (samples) used to train the segmentation algorithm. Examples of segmentation with manually selected samples used for building detection are given in [Dogruer et al., 2007] and in Chapter 6 ([Persson et al., 2005]).

The method presented in this chapter replaces these manually picked training samples with samples directed by the mobile robot. The virtual sensor for building detection described in Chapter 4 is used to determine which parts of an occupancy map belong to a building (wall estimate) resulting in the semantic map described in Chapter 5. Matching of wall estimates found in the semantic map with edges detected in an aerial image followed by colour segmentation is utilized to find building hypotheses. The matching is possible since geo-referenced aerial images are used and an absolute positioning system is installed onboard the robot. The matched lines are then used in region- and boundary-based segmentation of the aerial image for detection of buildings. The purpose is to detect building outlines faster than the mobile robot can explore the area by itself. Using this method the robot can estimate the size of found building regions without actually rounding the building. The method does not assume a perfectly up to date aerial image, in the sense that buildings may exist although they are not present in the aerial image, and vice versa. It is therefore possible to use globally available[1] geo-referenced images.

7.1.1 Outline and Overview

In Section 7.2 a presentation of work related to the use of aerial images in mobile robotics is given. The description of the proposed system is divided into three main parts. The first part, Section 7.3, concerns the estimation of walls by the mobile robot and edge detection in the aerial image. At ground level, wall estimates are extracted from the probabilistic semantic map described in Chapter 5. This map is basically an occupancy map built from range data and labelled using a virtual sensor for building detection (Chapter 4) mounted on the mobile robot. The second part, Section 7.4, describes matching of wall estimates from the mobile robot with the edges found in the aerial image. To determine potential matches between the wall estimates and the roof outlines, geo-referenced aerial images are used and the mobile robot has an onboard absolute positioning system (GPS). The third part, Section 7.5, presents the segmentation of an aerial image based on the matched lines. The matched lines are used in region- and boundary-based segmentation of the aerial image for detection of buildings. This segmentation will be referred to as the local segmentation as opposed to the global segmentation presented in Chapter 8. Descriptions of the experiments performed and the results obtained are found in

[1] E.g. Google Earth, Microsoft Virtual Earth, satellite images from IKONOS and its successors.

Section 7.6. Finally, the chapter is concluded and suggestions for future work are given in Section 7.7.

7.2 Related Work

Overhead images have been used in combination with ground vehicles in a number of applications. Oh *et al.* used map data to bias a robot motion model in a Bayesian filter to areas with higher probability of robot presence [Oh et al., 2004]. It was assumed that probable paths were known in the map. Since mobile robot trajectories are more likely to follow these paths in the map, GPS position errors due to reflections from buildings were compensated using the map priors.

Pictorial information such as aerial photos and city-maps have been used for registration of sub-maps and subsequent loop-closing in SLAM [Chen and Wang, 2006]. Aerial images were used by Früh and Zakhor in Monte Carlo localization of a truck during urban 3D modelling [Früh and Zakhor, 2004].

A method to detect building outlines, also without elevation data, is to fuse SAR (Synthetic Aperture Radar) images and aerial images [Tupin and Roux, 2003]. The building location was established in the overhead SAR image, where walls from one side of buildings can be detected through double reflections on the ground and a wall. The complete building outline was then found using edge detection in the aerial image. Parallel and perpendicular edges were considered and the method belongs to edge-only segmentation approaches. This work is similar to the work presented in this chapter in the sense that it uses a partly found building outline to segment a building from an aerial image.

Combination of edge and region information for segmentation of aerial images has been suggested in several publications. Two papers from which this work took inspiration are [Mueller et al., 2004] and [Freixenet et al., 2002]. Mueller *et al.* presented a method to detect agricultural fields in satellite images. First, the most relevant edges were detected. These were then used to guide both the smoothing of the image and the following segmentation in the form of region growing. Freixenet *et al.* investigated different methods for integrating region- and boundary-based segmentation, and also claim that this combination is the best approach for image segmentation.

Commonly used colour image segmentation methods are reviewed in [Cheng et al., 2001]. Concerning the choice of colour space for segmentation the authors point out that no single colour space surpasses others for all type of images.

From the above discussed references and the references presented in Section 8.2, it can be concluded that aerial images contain information that is useful for mobile robots. For detection of building outlines in aerial images edge information is the most used feature, which is also confirmed by the references

presented in Section 6.3. All together there is considerable motivation for the approach presented in this chapter.

7.3 Wall Candidates

A major problem for building detection in aerial images is to decide which of the edges in the aerial image correspond to building outlines. The idea of our approach is to match wall estimates extracted from two perspectives in order to increase the probability that a correct segmentation is achieved. In this section the process of extracting wall candidates is described, first from the mobile robot's perspective at ground-level and then from aerial images.

7.3.1 Wall Candidates from Ground Perspective

The wall candidates from the ground perspective are extracted from a semantic map acquired by a mobile robot as described in Chapter 5. In Chapter 5 three different occupancy maps (handmade, 2D laser data and 3D laser data) were used to create probabilistic semantic maps with two classes: buildings and nonbuildings. In this chapter, the probabilistic semantic map based on the occupancy grid map built with 2D laser data is used since it is the map built from robot measurements that covers most buildings. This probabilistic semantic map is presented in Figure 7.1.

The lines representing probable building outlines are extracted from the probabilistic semantic map using the same implementation [Kovesi, 2000] as previously used in Section 4.2. The implementation of the line extraction algorithm and the used parameter setting is described in Appendix B.1. The extracted lines representing wall estimates are given in Figure 7.2, which also shows the robot trajectory where data for the probabilistic semantic map were collected.

7.3.2 Wall Candidates in Aerial Images

Edges extracted from an aerial image taken from a nadir view are used as potential building outlines. The edge image is a binary image from which straight lines are extracted to be used as wall candidates in the matching process described in Section 7.4.

Two common categories of colour edge detection methods are output fusion methods and multi-dimensional gradient methods [Ruzon and Tomasi, 1999]. In colour edge detection with output fusion, edge detection is performed separately on the three components of the colour space used and the resulting edges are fused. In multi-dimensional gradient methods, the gradients from the three components are fused and then the edges are defined. Here, the first alternative with the edge detection performed separately on the three RGB-components

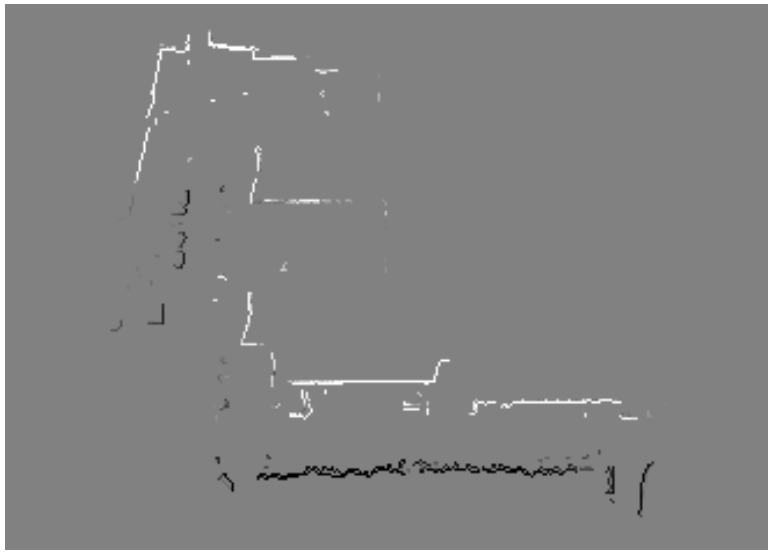


Figure 7.1: The probabilistic semantic map used in the experiments. White cells denote high probability of walls and dark cells show outlines of nonbuilding entities.

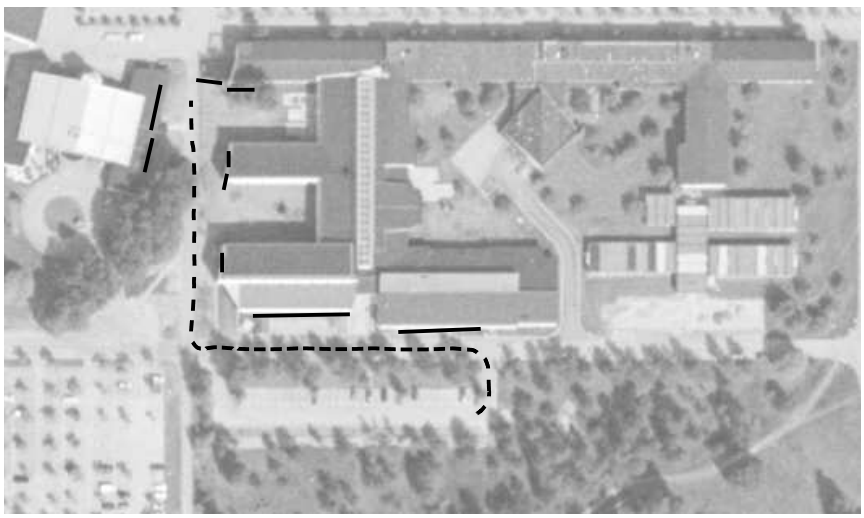


Figure 7.2: The trajectory of the mobile robot (dashed), the ground level wall estimates (solid) and the aerial image used (©Örebro Community Planning Office). The semantic map in Figure 7.1 covers the upper left part of this figure.

using Canny's edge detector [Canny, 1986] is applied. The resulting edge image I_e is calculated by fusing the binary images obtained for the three colour components with a logical OR-function. Finally a thinning operation[2] is performed to remove points that occur when edges appear slightly shifted in the different components. For line extraction in I_e the same implementation and parameters as in Section 7.3.1 were used. The lines extracted from the edges detected in the aerial image in Figure 7.2 are shown in Figure 7.3.

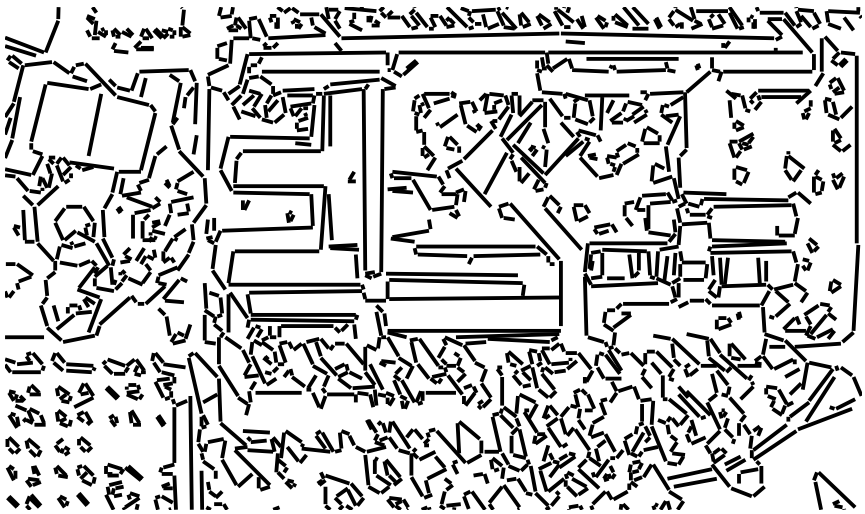


Figure 7.3: The lines extracted from the edge version of the aerial image.

The colour edge detection method is used because it finds more edge points than gray scale edge detection. This is because edges on the border between areas that have different colours but similar intensity are not detected in gray scale versions of the same image. In a test where the two methods had the same segmentation parameters, the colour version produced 19% more edge points resulting in 17% more detected lines for an aerial image of size ca. 800×1300 pixels (400×650 m). Figure 7.4 gives a close-up from that test to show an example of the differences. The calculation time of the colour edge detection is slightly more than three times longer than ordinary gray scale edge detection. This time is still small in comparison to the routines used for detecting lines in the edge images (for the same example as above: ca. 20%).

[2]The Matlab command *bwmorph(im,'thin',Inf)* was used.

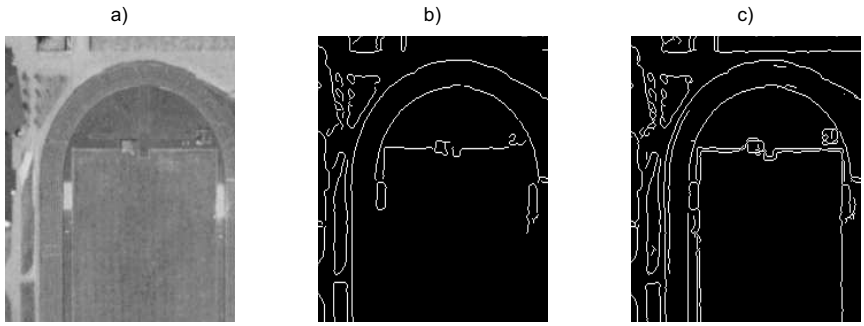


Figure 7.4: Gray scale (b) and colour edge detection (c) in an aerial image (a). In the top the colour version finds additional edges where light green vegetation meets light gray ground, and in the lower part edges are found around the green football field where the grass meets the red running tracks.

7.4 Matching Wall Candidates

The purpose of the wall matching step is to relate wall estimates, obtained at ground level with the mobile robot, to the edges detected in the aerial image. All wall estimates are represented as line segments. A wall estimate found by the mobile robot is denoted as L_g (g indicates ground-level) and the N lines representing the edges found in the aerial image by L_a^i with $i \in \{1, \ldots, N\}$ (a indicates aerial). Both line types are geo-referenced in the same Cartesian coordinate system.

The lines from both the aerial image and the semantic map may be erroneous, especially concerning the line endpoints, due to occlusion, errors in the semantic map, different sensor coverage, etc. A measure for line-to-line distances that can handle partially occluded lines is therefore needed. Hence, the length of the lines is not considered and line matching is based only on the line directions and the distance between two characteristic points, one point on each line. The line matching calculations are performed in two steps described below: determination of the two characteristic points and computation of the distance measure to find the best matches.

7.4.1 Characteristic Points

In this section it is described how the characteristic points on the two lines compared are determined. For L_g the line midpoint, P_g, is used. To cope with the possible errors described above, the point P_a on L_a^i that is closest to P_g is selected as the best candidate to be used in our line distance measure.

To calculate P_a, let e_n be the orthogonal line to L_a^i that intersects L_g in P_g, see Figure 7.5. The intersection between e_n and L_a^i is denoted as ϕ where

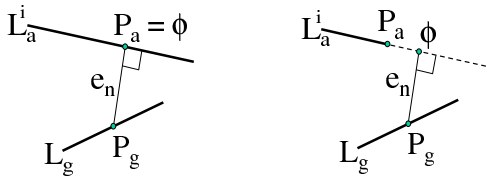


Figure 7.5: Selection of characteristic points for the computation of the distance measure between two lines. The figure shows the line L_g (ground level wall candidate) with its midpoint P_g, the line L_a^i (aerial image wall candidate), and the normal to L_a^i, e_n. To the left, $P_a = \phi$ since ϕ is on L_a^i. To the right, P_a is the endpoint of L_a^i since ϕ is not on L_a^i.

$\phi = e_n \times L_a^i$ (using homogeneous coordinates). The intersection ϕ may be outside the line segment L_a^i, see right part of Figure 7.5. It should therefore be checked if ϕ is within the endpoints and if it is, set $P_a = \phi$. If ϕ is not within the endpoints, then P_a is set to the closest endpoint on L_a^i.

$$P_a = \begin{cases} \phi & \forall \phi \in [m_{min}, m_{max}] \\ \Phi & \forall \phi \notin [m_{min}, m_{max}] \end{cases} \tag{7.1}$$

where

$$\Phi = \begin{cases} L_{a_1}^i & \text{if } \|L_{a_1}^i - \phi\| < \|L_{a_2}^i - \phi\| \\ L_{a_2}^i & \text{if } \|L_{a_1}^i - \phi\| > \|L_{a_2}^i - \phi\| \end{cases} \tag{7.2}$$

and $L_{a_1}^i$ and $L_{a_2}^i$ are the endpoints of L_a^i, $\|\cdot\|$ is the Euclidean distance, and

$$m_{min} = \begin{bmatrix} \min(L_{a_{x1}}^i, L_{a_{x2}}^i) \\ \min(L_{a_{y1}}^i, L_{a_{y2}}^i) \end{bmatrix} \tag{7.3}$$

$$m_{max} = \begin{bmatrix} \max(L_{a_{x1}}^i, L_{a_{x2}}^i) \\ \max(L_{a_{y1}}^i, L_{a_{y2}}^i) \end{bmatrix}. \tag{7.4}$$

$L_{a_{xj}}^i$ and $L_{a_{yj}}^i$ denote the x- and y-coordinate of the endpoints of L_a^i.

7.4.2 Distance Measure

The calculation of the distance measure is inspired by [Guerrero and Sagüés, 2003], which describes geometric line matching in images for stereo matching. The complexity in these calculations has been reduced by exclusion of the line lengths which also results in fewer parameters that need to be determined. Matching is performed using L_g's midpoint P_g, the closest point P_a on L_a^i and the line directions θ_g and θ_a. First, a difference vector is calculated as

$$\mathbf{r}_{\mathbf{\Delta}} = [P_{g_x} - P_{a_x}, P_{g_y} - P_{a_y}, \theta_g - \theta_a]^T. \tag{7.5}$$

Second, the similarity is measured as the Mahalanobis distance

$$d = \mathbf{r_\Delta}^T \mathbf{R}^{-1} \mathbf{r_\Delta} \tag{7.6}$$

where the diagonal covariance matrix $\mathbf{R}$ is defined as

$$\mathbf{R} = \begin{bmatrix} \sigma_{Rx}^2 & 0 & 0 \\ 0 & \sigma_{Ry}^2 & 0 \\ 0 & 0 & \sigma_{R\theta}^2 \end{bmatrix} \tag{7.7}$$

with σ_{Rx}, σ_{Ry}, and $\sigma_{R\theta}$ being the expected standard deviation of the errors between the ground-based and aerial-based wall estimates. Only the relation between the parameters in Equation 7.7 influences the line matching. The important relation is $\sigma_{R\theta}^2/\sigma_{Rx}^2$ and usually $\sigma_{Rx}^2 = \sigma_{Ry}^2$ for symmetry reasons. Note that the distance measure is not strictly a metric in a mathematical sense, due to the non-symmetric method for selecting characteristic points.

7.5 Local Segmentation of Aerial Images

This section describes how local segmentation of the colour aerial image is performed. Generally, segmentation methods can be divided into two groups; edge-based and similarity-based [Gonzales and Woods, 2002]. In our case these approaches are combined by first performing edge based segmentation for detection of closed areas and then colour segmentation based on a small training area to confirm the area's homogeneity. Figure 7.6 gives a short description of the sequence that is performed for each line L_g. The process is stopped, either when a region has been found or when all lines in L_a that are close enough to the present line in L_g to be considered, have been checked. The "close enough" criterion can be implemented using the Euclidean distance between the characteristic points P_g and P_a defined in Section 7.4.1. However, in the current implementation this was not activated during the experiment in order to be able to study whether additional regions were found.

7.5.1 Edge Controlled Segmentation

Based on the edge image I_e constructed from the aerial image, a closed area that is limited by edges is searched for. Since there might be gaps in the edges, small gaps need to be found and filled [Mueller et al., 2004]. Morphological operations are used to first dilate the edge image in order to close gaps and then search for a closed area on the side of the matched line that is opposite to the mobile robot. When this area has been found it is dilated in order to compensate for the previous dilation of the edge image. This procedure is further described by Figures 7.7 and 7.8.

1. Sort the set of lines L_a based on d from Equation 7.6 in increasing order and set $i = 0$.
2. Set $i = i + 1$.
3. Define a start area A_{start} on the side of L_a^i that is opposite to the robot.
4. Check if A_{start} includes edge points (parts of edges in I_e). If yes, return to step 2. This check ensures that a region has a minimum width and depth.
5. Perform edge controlled segmentation, see Section 7.5.1.
6. Perform homogeneity test, see Section 7.5.2.

Figure 7.6: Description of the local segmentation process.

1. Initialize a starting area, A_{start}
2. Dilate the edge map
3. Find the closed area A_{small} that includes the part of A_{start} that is free from edge pixels
4. Calculate A_{final} as the dilation of A_{small}

Figure 7.7: Edge-based algorithm for finding closed areas and filling in small gaps in the edges.

7.5.2 Homogeneity Test

The initial starting area A_{start} is used as a training sample for a colour model and the rest of the region is evaluated based on this colour model. This means that the colour model does not gradually adapt to the growing region, but instead requires a homogeneous region on the complete region that is under investigation. Regions that gradually change colour or intensity, such as curved roofs, might then be partly rejected.

There are different approaches to represent colour models. One approach that is popular for colour segmentation is a Gaussian Mixture Model (GMM). Like Dahlkamp *et al.* [Dahlkamp et al., 2006] we tested both GMM and a model described by the mean and the covariance matrix in RGB colour space. The mean/covariance model was selected since it is faster and it was noted that the mean/covariance model performs approximately equally well as the GMM in our case. A limit O_{lim} is calculated for each model so that 95% of the training

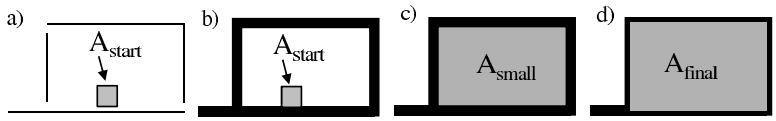


Figure 7.8: Illustration of edge controlled segmentation. a) shows a small part of I_e and A_{start}. In b) I_e has been dilated and in c) A_{small} has been found. Finally, d) shows A_{final} as the dilation of A_{small}.

sample pixels (i.e. pixels in A_{start}) have a Mahalanobis distance smaller than O_{lim}. O_{lim} is then used as the separator limit between pixels belonging to the class and the pixels that do not belong to the class.

7.5.3 Alternative Methods

Above a two step segmentation method to detect homogeneous regions surrounded by edges was presented. There exist a number of different segmentation methods that could have been applied instead. Two such methods are discussed in the following. The conclusions presented below are based on preliminary tests performed on the aerial image used in our experiment. For these tests, the parameters used in the respective algorithms were tuned manually.

The first method tested is a graph-based image segmentation (GBIS) [Felzenszwalb and Huttenlocher, 2004]. GBIS can adapt to the texture and can be set to reject small areas and therefore ignore small-sized disturbances such as shadows from chimneys. For this reason GBIS tends to produce very homogeneous results. A drawback is that GBIS has a tendency to leak and continue to grow outside areas that humans would consider to be closed. Therefore, GBIS does not seem to be an option to replace both steps in our two step method, but it is an alternative to the homogeneity test. In conjunction with the edge controlled segmentation it turns out that GBIS produces similar segmentation results to the mean/covariance model.

The second method tested is a modified flood fill algorithm. The algorithm takes starting pixels from A_{start} and performs region growing limited by colour difference to the starting pixels and local gradient information. Let $\mathbf{C}$ be the mean value vector (RGB) of the starting pixels, $\mathbf{P}_i$ any pixel that has been selected to be inside the region and $\mathbf{P}_n$ a neighbouring pixel (4-connected with $\mathbf{P}_i$) to be tested to see whether it should be included in the region. For each $\mathbf{P}_n$ a local value g_{loc} is calculated as

$$g_{loc} = e^{-\sum_{j=r,g,b}(\mathbf{P}_n(j)-\mathbf{C}(j))^2/\sigma^2_{col}} e^{-\sum_{j=r,g,b}(\mathbf{P}_n(j)-\mathbf{P}_i(j))^2/\sigma^2_{grad}} \quad (7.8)$$

The value of g_{loc} is then compared to a threshold to see if $\mathbf{P}_n$ should be included in the region or not. Due to the use of the local gradient this algorithm performs equally well as the mean/covariance model, both as a replacement for the two steps and when it is used only for the homogeneity check. This modified flood fill algorithm can also leak, like GBIS, but only to areas with colours similar to A_{start}, since $\mathbf{C}$ only depends on the starting pixels.

7.6 Experiments

7.6.1 Data Collection

Data were collected with the mobile robot Tjorven, equipped with differential GPS, a horizontally mounted laser range scanner, cameras and odometry (Section 2.2 gives more details). The robot is equipped with two different types of cameras, an ordinary camera mounted on a PT-head and an omni-directional camera. Here, the omni-directional camera is used. From each omni-image eight planar images (every $45°$), with a horizontal and vertical field-of-view of $56°$, were computed. These planar images are the input to the virtual sensor. The images were taken approximately every 1.5 m along the robot trajectory and were stored together with the corresponding robot pose. The trajectory of the mobile robot is shown in Figure 7.2. Since the ground where the robot was driven during the experiment is mainly flat, inertial sensors were not needed. This can be confirmed by visual inspection of the resulting occupancy map in Figure 7.9.

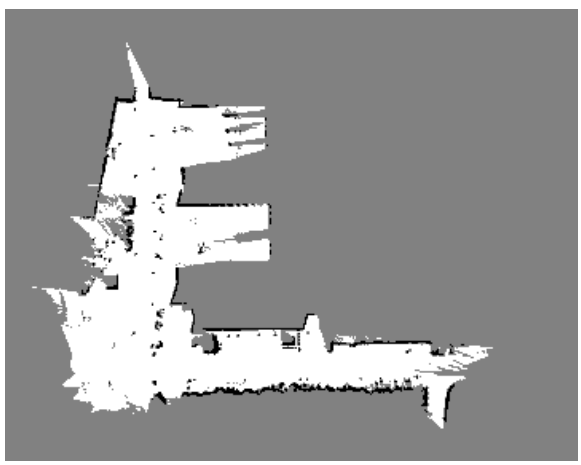


Figure 7.9: Occupancy map used to build the semantic map presented in Figure 7.1.

7.6.2 Tests of Local Segmentation

The occupancy map shown in Figure 7.9 was used for the experiment. This map was built from data measured by the laser range scanner (with 180 degrees field of view) and position data obtained from fusion of odometry and DGPS, see Appendix B.3. The grid cell size was 0.5 m, the range of the data was limited to 40 m and the map was built using the known poses and a standard Bayes update equation as described in [Thrun et al., 1998]. Even though this 2D map works well in our experiments (with exception of the hedge/building mix-up described in Section 5.4.2), one should note that a fixed horizontally mounted 2D laser is not the optimal choice of sensor configuration for detection of building outlines. Alternative methods suitable for capturing large objects in outdoor environments are 3D laser [Surmann et al., 2001], a vertically mounted laser range scanner [Früh and Zakhor, 2004] or (motion) stereo vision [Huber and Graefe, 1994].

The occupied cells in the map (marked in black in Figure 7.9) were labelled by the virtual sensor giving the probabilistic semantic map presented in Figure 7.1. The probabilistic semantic map contains two classes: buildings (values above 0.5) and nonbuildings (values below 0.5). From this semantic map the grid cells with a high probability of being a building[3] (above 0.9) were extracted and converted to the lines L_g^M presented in Figure 7.2. Matching of these lines with the lines extracted from the aerial image L_a^N was then performed (see Figure 7.3). Finally, based on the best line matches, segmentation was performed as described in Section 7.5, where each ground-level line L_g can lead to one extracted region.

The three parameters in $\mathbf{R}$ (Equation 7.7) were set to $\sigma_{Rx} = 1$ m, $\sigma_{Ry} = 1$ m, and $\sigma_{R\theta} = 0.2$ rad. The first two parameters reflect a possible error of 2 pixels between the robot position and the aerial image for the given resolution, and the third parameter allows, for example, each endpoint of a 10 pixel long line to be shifted one pixel (parallel edges in the aerial image do not always result in parallel lines, see roof outline in Figure 7.3). In the tests described in the following paragraph, it will be shown that the matching result is not sensitive to small changes of these parameters. In the experiment, the start area A_{start} (Figure 7.6) was a 8×8 pixels square, equivalent to 4×4 m and a square structuring element of size 3×3 was used for the dilations described in Section 7.5.1.

Two different types of test have been performed. The parameters for these tests are defined in Table 7.1. *Tests 1-3* represent the nominal cases where the collected data are used as they are. These tests intend to show the influence of a changed relation between σ_{Rx}, σ_{Ry} and $\sigma_{R\theta}$ by varying $\sigma_{R\theta}$. In *Test 2* $\sigma_{R\theta}$ is decreased by a factor of 2 and in *Test 3* $\sigma_{R\theta}$ is increased by a factor of 2. In *Tests*

[3]The limit 0.9 was chosen with respect to the probabilities used in the process of building the probabilistic semantic map, see Chapter 5. With this limit at least two positive building readings are needed for a single cell to be used in L_g^M.

4 and *5* additional uncertainty (in addition to the uncertainty already present in L_g^M and L_a^N) was introduced. This uncertainty is in the form of Gaussian noise added to the midpoints (σ_x and σ_y) and directions (σ_θ) of L_g^M and evaluated in Monte Carlo simulation [Metropolis and Ulam, 1949] with 20 runs.

Test	σ_x [m]	σ_y [m]	σ_θ [rad]	$\sigma_{R\theta}$ [rad]	N_{run}
1	0	0	0	0.2	1
2	0	0	0	0.1	1
3	0	0	0	0.4	1
4	1	1	0.1	0.2	20
5	2	2	0.2	0.2	20

Table 7.1: Parameters used in the local segmentation tests.

7.6.3 Result of Local Segmentation

The local segmentation has a limited range and the ground truth area can be beyond this range without affecting the resulting segmentation, e.g. by including new buildings that are not seen by the robot. A traditional quality measure such as the true positive rate is therefore not suitable for these tests, since it depends on the size of the ground truth area. Instead, the *positive predictive value*, *PPV* or *precision*, was used as the quality measure. *PPV* is calculated as

$$PPV = \frac{TP}{TP + FP} \tag{7.9}$$

where *TP* is the number of true positives and *FP* is the number of false positives.

The results of *Test 1* show a high positive predictive value of 96.5%, see Table 7.2. The resulting segmentation is presented in Figure 7.10 where the building regions are found along the robot trajectory. Three deviations from an ideal result can be noted. At *a* and *b* tree tops were obstructing the wall edges in the aerial image and therefore the area opposite to these walls was not detected as a building, and a gap between two regions appears at *c* due to a wall visible in the aerial image. Finally, a false area, to the left of *b*, originates from an error in the semantic map where a low hedge in front of a building was marked as building because the building was the dominating object in the camera view.

The results of *Test 1-3* are very similar, indicating that the algorithm in this case was not particularly sensitive to the changes in $\sigma_{R\theta}$. In *Tests 4* and *5* the scenario of *Test 1* was repeated using a Monte Carlo simulation with introduced pose uncertainty. These results are presented in Table 7.2. One can note that the difference between the nominal case *Test 1* and *Test 4* is very small. In *Test 5* where the additional uncertainties are higher, the positive predictive

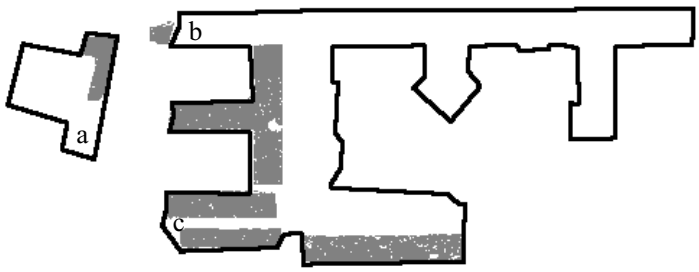


Figure 7.10: The result of the local segmentation of the aerial image, based solely on the few wall estimates shown in Figure 7.2. The ground truth building outlines are drawn in black.

value decreased slightly. Based on these results the approach for local segmentation of the aerial image was found to be very robust.

Test	*PPV* [%]
1	96.5
2	97.0
3	96.5
4	96.8 ± 0.2
5	95.9 ± 1.7

Table 7.2: Results for the tests defined in Table 7.1. The results of Test 4 and 5 are presented with the corresponding standard deviation computed from the 20 Monte Carlo simulation runs.

7.7 Summary and Conclusions

This chapter discusses how aerial images can be used to extend the observation range of a mobile robot. A virtual sensor for building detection on a mobile robot is used to build a ground level probabilistic semantic map. This map is used to link semantic information to a process for building detection in aerial images. The approach addresses two difficulties simultaneously: 1) buildings are hard to detect in aerial images without elevation data and 2) the limitation of the sensors of mobile robots. Concerning the first difficulty the results show a high classification rate and it can therefore be concluded that the semantic information can be used to compensate for the absence of elevation

data in aerial image segmentation. The benefit from the extended range of the robot's view can clearly be noted in the presented example. Even though the roof structure in the example is quite complicated, the outline of large building parts can be extracted although the mobile robot only has seen a minor part of the surrounding walls.

There are a few issues that should be noted:

- It turns out that a complete building outline is seldom segmented due to factors such as different roof materials, different roof inclinations and additions on the roof.
- It is important to check several lines from the aerial image since more edges than expected could have been extracted. For example, roofs can have extensions in other colours, and not only roofs and ground can be seen in the aerial image. When the nadir view is not perfect, walls can appear in the image in addition to the roof outline. Such a wall will produce two edges in the aerial image, one where ground and wall meet and one where wall and roof meet.
- The presented solution performs a local segmentation of the aerial image after each performed matching of a ground-level line with lines in the aerial image. An alternative solution would be to first segment the whole aerial image and then confirm or reject the regions as the mobile robot finds new wall estimates.

An extension to local segmentation elaborated in the next chapter is to use the building estimates as training areas for further colour segmentation in order to make a global search for buildings within the aerial image. This global search can also utilize the robot's knowledge of already traversed areas to recognise other potentially driveable paths.

Chapter 8
Global Segmentation of Aerial Images

8.1 Introduction

Aerial images contain information that can be used to extend the range of mobile robot sensors. This was shown in the previous chapter where segmentation of an aerial image to detect building areas near to the path of a mobile robot (directed by the walls found by the robot) was performed. An extension that will increase the view further is to use the building estimates as training areas for colour segmentation in order to make a global search for buildings within the entire aerial image. Based on the robot's knowledge of already traversed areas, it is also possible to recognise other driveable areas.

In this chapter the approach from Chapter 7 is extended. The extension includes global segmentation of buildings in the aerial image, the introduction of a new semantic class for ground (that is potentially driveable by the robot) and the introduction of the concept and framework of the predictive map. The aim of global segmentation is to build a map that predicts regions such as driveable ground and buildings. To perform global segmentation colour models are used. These colour models are acquired from the aerial image, directed by information from the local segmentation (Chapter 7) and by information collected with a mobile robot. The purpose with the global segmentation is to detect building outlines and driveable paths faster than the mobile robot can explore the area by itself. Using this method, the robot can estimate the outline of found buildings and "see" around one or several corners without actually having visited these areas by itself. As for the local segmentation, this method does not assume a perfectly up-to-date aerial image; buildings may exist although they are missing in the aerial image, and vice versa. It is therefore possible to use globally available geo-referenced images from sources such as Google Earth and Microsoft Virtual Earth, even though up-to-date imagery is preferable.

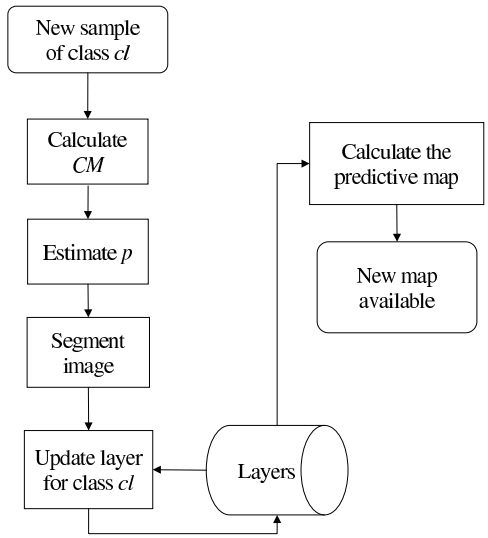


Figure 8.1: Flow chart of the process for calculating the predictive map (PM).

8.1.1 Outline and Overview

In this chapter the range of the robot's view is increased by prediction of the surrounding regions using a process that involves segmentation of aerial images. Trained colour models are used in global segmentation of the entire aerial image. The purpose is to build a map that predicts different types of areas, e.g., ground and buildings, and it is therefore called the predictive map (PM). When the PM includes both driveable ground and obstacles such as buildings, it can serve as an input to a path planning algorithm.

The global segmentation of an aerial image using colour models captures all pixels with the same property as the training sample. For example, if the buildings that were detected by local segmentation are used as training samples, buildings with roofs in similar colours as these buildings will be detected. To complement the detection of buildings an additional class, *ground*, is utilized in the experiments of this chapter.

To calculate the predictive map incrementally two main steps are performed; 1) the aerial image is segmented when a new colour model (CM) is available and 2) the predictive map is recalculated using the result from the latest segmentation. Figure 8.1 shows a flow chart of the updating process.

In Section 8.2 previous work related to mobile robots using aerial images and on detection of driveable ground is described. The segmentation process is presented in Section 8.3. This process includes the extraction of training

samples, calculation of a colour model, and segmentation based on the colour model. The results from the segmentation process are stored in class layers, one layer per class. The predictive map and its calculation from the class layers are described in Section 8.4. In Section 8.5 the combination of the local and global information, the results obtained in the previous chapter and the predictive map respectively, is presented and discussed. The experiments performed and the obtained result are presented in Section 8.6. Finally, the chapter is concluded in Section 8.7 with a summary and a discussion.

8.2 Related Work

In Chapter 6, automatic building detection in aerial images was discussed. In Section 7.2 work related to aerial images and mobile robots relevant for the process described in Chapter 7 was presented. In this section these works are complemented with an overview of works on mobile robots using aerial images and works on detection of driveable ground.

Information from aerial images can be used to support long range navigation of autonomous vehicles. Silver *et al.* discuss a method to produce cost maps from aerial surveys [Silver et al., 2006]. The aerial surveys give heterogeneous data (e.g. data recorded with different sampling density) that are registered and used in classification of ground surface. The used features are image based, both from visible light and near-infrared light, elevation based (rasterized elevation data available from multiple sources) and point cloud based (from airborne laser range scanners).

In a similar example heterogeneous data from maps and aerial surveys are used to construct a world model with semantic labels [Scrapper et al., 2003]. This model was compared with data from sensors mounted on a ground vehicle. The system uses the semantic labelling to focus a search for particular features in the vehicle neighbourhood, allowing a fast scene interpretation by targeting the sensor data processing to regions that are predicted to be most interesting.

Manually extracted training samples have been used to segment a satellite image for use in mobile robot localization and navigation [Dogruer et al., 2007]. HSI colour space was chosen in order to reduce the sensitivity to shades in the classification. Evaluation with a mobile robot has not yet been performed.

In this chapter a class for ground is used, but for the mobile robot it would be of benefit to know whether the ground is driveable or not. There is a possibility to extract information about the traversability of ground based on measurements performed by an unmanned vehicle. Models of driveable ground can be extracted in different ways. Laser range scanners are popular and efficient when it comes to observe the ground close to the vehicle, but for long ranges it seems that vision is more often used. In the following described works, the

intention has been to find areas of driveable ground not only as opposed to obstacles, but with the purpose of obtaining a forecast of the conditions ahead of the vehicle using vision to extrapolate driveable areas.

The area traversed by a mobile robot has been used as the indicator of obstacle free ground [Ulrich and Nourbakhsh, 2000]. When the robot has driven over a specific area it knows that this area is driveable. A forward looking camera registers the area that the robot will pass and models in HSI colour space are calculated from image histograms. The system extrapolates the information forward and uses this to separate free ground from obstacles.

One can also assume that when a vehicle already is on a road, a forward looking camera has the road in view (the part of the image showing the ground closest to the vehicle) [Song et al., 2006]. Song *et al.* used this to extract information for a motion planning system used on ill-structured roads during DARPA Grand Challenge.

Gaussian Mixture Models (GMM) are popular for colour segmentation. GMM were used in detection of drivable areas in desert terrain [Dahlkamp et al., 2006] for DARPA Grand Challenge. An algorithm that extends the range of a laser range scanner by the use of vision was developed. The laser scans the area close to the vehicle and the obstacle free area is mapped to the camera image where it is used for training the GMM. The rest of the image can then be classified as drivable or nondriveable areas. It turned out that the authors could reduce the complexity of the GMM to only one model described by the mean and the covariance matrix with only a slight decrease of performance.

AdaBoost was used by Guo *et al.* to train a system to detect roads in difficult situations [Guo et al., 2006]. They used two types of weak classifiers, one based on colour samples that are compared to colour histograms and the other on rectangular features (reminiscent of Haar basis functions) introduced for face detection [Viola and Jones, 2004]. The final system detects roads in front of a vehicle and handles both shadowed regions and water pools.

A recent example to forecast terrain traversability is given by Karlsen and Witus. They used vision and extracted texture features based on standard deviation and entropy [Karlsen and Witus, 2007]. The features observed during training were clustered using fuzzy c-means. The system is able to forecast vehicle-terrain interaction in upcoming terrain.

Based on the above referenced work it can be concluded that several systems for finding driveable terrain exist. It can therefore be assumed that also with the omni-directional camera onboard our mobile robot driveable terrain could be identified and used as a class complementary to the ground class currently used.

8.3 Segmentation

The segmentation of the aerial image is based on colour models. In the example used in this chapter, models are calculated for the two classes *building* and

ground. Figure 8.1 shows a flow chart of the updating process. In this section the first part, up to the image segmentation, is explained. The algorithm is adapted to work also in an on-line situation. When a *New sample* belonging to class *cl* is available, a new colour model *CM* is calculated. Based on the quality of *CM*, a measure $p, 0 \leq p \leq 1$ should be estimated. The last step is to classify the pixels in the aerial image with the trained colour model. In summary the algorithm is as follows:

1. Take a sample (a number of pixels from an area that is expected to belong to class *cl*).
2. Train a colour model in RGB colour space.
3. *Estimate* p.
4. Perform a classification using the colour model.

In the following paragraphs the training samples (step 1), the colour models (step 2), and the classification (step 4) are described. The parameter p should reflect the certainty of the classification performed with the colour model and could also be influenced by the probabilities of the connected regions in the probabilistic semantic map. Estimation of the parameter p is still an unsolved issue and left for future work. In our experiments $p = 0.7$ was used.

8.3.1 Training Samples

The system needs to define training samples of the classes that shall be segmented. Two classes are of interest here, buildings and ground. The method presented in the previous chapter was used to find local estimates of buildings. Therefore, to define the colour models for the building class, the building estimates found by local segmentation are used as training areas.

To extract colour models that represent the different ground areas, the occupancy grid map and the edge version of the aerial image, I_e, are combined. The free cells in the occupancy grid map define the regions in I_e that represent ground. The combination of I_e and the occupancy grid map can be done either under the assumption that the navigation is precise giving a perfect registration or by reduction of the area of free cells with the estimated size of the navigation error. Only the latter was used in our work. Assuming that the DGPS gave rather accurate positions and an accurate heading angle could be calculated from the trajectory, the latter approach was used to remove the small navigation errors. The area of free cells in the occupancy grid map was reduced by morphological erosion with a square structuring element of size 5×5 pixels to compensate for errors up to 2 pixels (1 m) in all directions, resulting in a ground region. An example of the combination of the ground region with the edges in I_e is shown in Figure 8.2. Next, edge controlled segmentation of the

Figure 8.2: The combined binary image of free points (reduced using morphological erosion with a square structuring element of size 5×5 pixels) and edges in I_e.

ground region is performed, as described in Section 7.5.1, to find the individual ground areas with boundaries defined by the I_e. The largest areas[1] found in the edge controlled segmentation, point out samples in the aerial image that are used to train mean/covariance models, the same type of colour models as in Section 7.5.2.

The free space found in the occupancy grid map can be considered to be driveable ground assuming that there are no negative obstacles or other features, which cannot be sensed with the horizontally mounted 2D laser scanner and prevent the robot from driving safely. However, since it cannot be guaranteed that the free space in the occupancy grid map in fact corresponds to a driveable area, the new class is called *ground*.

The result from the local building segmentation (an example was presented in Figure 7.10) and the ground information from the occupancy grid map are referred to as the *local information*, since it results from direct observation by the mobile robot.

8.3.2 Colour Models and Classification

Both the colour models and the classification algorithm follows the same ideas as the procedure for the homogeneity test used in the local segmentation presented in Section 7.5.2. The training samples are described in a

[1]The limit was set to 50 pixels (12.5 m^2) in order to avoid small areas that could represent movable objects such as cars and small trucks.

mean/covariance model in RGB space and the segmentation is based on the Mahalanobis distance to classify the aerial image:

1. Set a limit O_{lim} based on how many outliers that are expected in the model (here O_{lim} represents 95% of the training sample pixels).
2. Calculate the Mahalanobis distances between the colour model and the pixels in the sample. Set d_{lim} to the distance that represent O_{lim}.
3. Use the Mahalanobis distance in the segmentation of the aerial image, with d_{lim} as the separator between pixels belonging to *cl* and the pixels that don't.

8.4 The Predictive Map

This section describes the layout of the predictive map (PM) and the method used to calculate incremental updates of it.

The PM is designed to handle multiclass problems and updating this map can be performed incrementally. The PM is a grid map of the same size as the aerial image that is segmented. For each of the n classes, a separate layer l_i, with $i \in \{1, \ldots, n\}$, is used to store the accumulated segmentation results. These layers also have the same size as the aerial image. The colour models used to segment the aerial image define a subset of a class through a binary classifier.

To calculate the predictive map incrementally two main steps are performed:

1. When the aerial image has been segmented with a new colour model, the layer of that class is updated.
2. The predictive map is recalculated using the result from the updated layers.

From the segmentation of the aerial image a temporary layer for *class* is obtained. The old layer of *class*, l_{cl}, is fused with the temporary layer using a max function. Alternative methods to fuse the layers, such as a Bayesian method, should be evaluated when a method to calculate the parameter p has been decided.

8.4.1 Calculating the PM

The predictive map is based on voting from separate layers l_i for the n classes, one layer for each class. The voting is performed on the layers cell by cell using IF-THEN rules biased with c_{ij}:

$$\text{IF } l_i^{xy} > l_j^{xy} + c_{ij} \quad \forall\, j \neq i \text{ THEN } pm^{xy} = class_i \tag{8.1}$$

where l_i^{xy} denotes cell (x, y) in layer i and pm^{xy} denotes cell (x, y) in PM. If the condition cannot be fulfilled due to conflicting information, pm^{xy} is set to *unknown*. To evaluate the similarity between cells, buffer zones are introduced in the voting process. The buffer zones are collected in the off-diagonal elements of a matrix $\mathbf{C}$ where $c_{ij} \geq 0, i \neq j, i = \{1, 2, \ldots, n\}, j = \{1, 2, \ldots, n\}$. Introducing the buffer zones makes it possible to adjust the sensitivity of the voting individually for all classes. If $c_{ij} = 0$ the rules in Equation 8.1 will turn into ordinary voting where the largest value wins and where ties give *unknown*.

During the experiments presented in this chapter two layers were used ($n = 2$); one building layer and one ground layer, and $\mathbf{C}$ was set to

$$\mathbf{C} = \begin{bmatrix} - & 0.1 \\ 0.1 & - \end{bmatrix} \tag{8.2}$$

(the values of the diagonal elements are not used).

All in all, the PM contains information about $n + 2$ categories. First there are the n different classes, then the *unknown* cells due to ambiguous class values and finally the unexplored cells that represent the remaining pixels, which cannot be explained by any of the trained colour models.

8.5 Combination of Local and Global Segmentation

The approaches described above and in the previous chapter result in two sets of information. The first is the local information that has been confirmed by the mobile robot and the second is stored in the PM. Where these sets overlap they can be fused into one final estimate. Since the local information has been confirmed by the mobile robot it is reasonable to let the local information have precedence over the PM by giving it a higher probability p. Fusion of the PM and the local information is carried out using the method described in Section 8.4.1 .

8.6 Experiments

8.6.1 Experiment Set-Up

The experiment reported in this chapter makes use of the same set-up as was described in Section 7.6.1. The additional information used is the ground estimation obtained from the occupancy grid map as described in Section 8.3.1.

8.6.2 Result of Global Segmentation

The result of the global segmentation is shown in Figures 8.3 and 8.4. Visual inspection of the result illustrates the potential of the approach. The PM based

on ground colour models from regions in Figure 8.2 and building colour models from the regions in Figure 7.10 is presented in Figures 8.3(a) (cells classified as ground and buildings) and 8.3(b) (unexplored and unknown cells).

Compared with the aerial image in Figure 7.2 the result is promising. One can now follow the outline of the main building and most of the paths, including paved paths, roads and beaten tracks, have been found. The main problem experienced during the work is caused by shadowed ground areas that look very similar to dark roofs resulting in the major part of the *unknown* cells.

If areas representing the unknown cells have already been classified by the mobile robot, as in Figures 7.9 and 7.10, that result has precedence over the PM (see discussion in Section 8.5). The final result is obtained when the PM is combined with the local information. For these pixels p is set to 0.9 and another update of the PM (using the method described in Section 8.4) is performed resulting in the map shown in Figure 8.4.

A formal evaluation of the ground class is hard to perform. Ground truth for buildings can be manually extracted from the aerial image, but it is hard to specify in detail the area that belongs to ground. Based on the ground truth of buildings and an approximation of the ground truth of ground as the non-building cells, statistics of the result are presented in Table 8.1. In the table all values in the right column, where the results from the combined PM and local information are shown, are better than those in the middle column (only PM).

As in Chapter 7, the *positive predictive value*, PPV or *precision*, has been used as the quality measure. PPV is calculated as

$$PPV = \frac{TP}{TP + FP} \tag{8.3}$$

where TP are the number of true positives and FP are the number of false positives. Since the PPV depends on the actual presence of the different classes in the aerial image, normalized values are also presented. The normalized values are calculated as

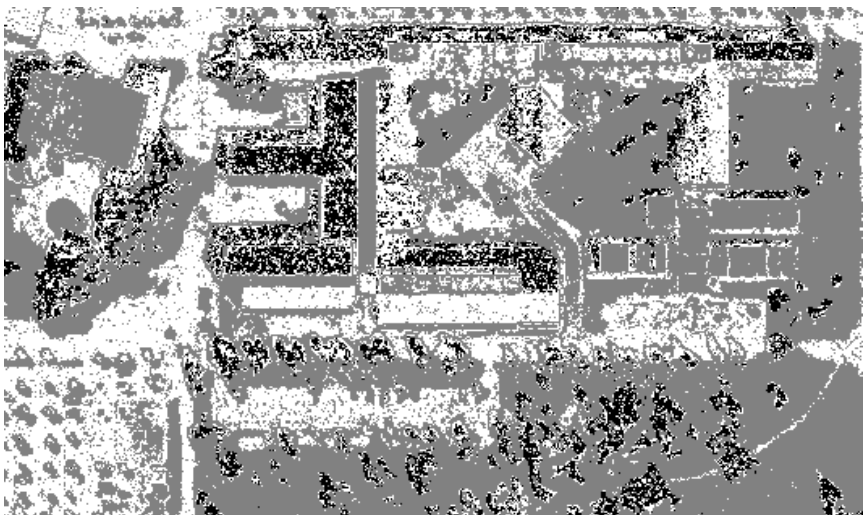
$$PPV_{norm} = \frac{TP_{cl}}{TP_{cl} + FP_{cl}\frac{GT_{cl}}{NGT_{cl}}} \tag{8.4}$$

where TP_{cl} and FP_{cl} are the numbers of true and false positives of class *cl* respectively. GT_{cl} is the number of ground truth cells of class *cl* and NGT_{cl} (not ground truth) is the difference between the total number of cells in the PM and GT_{cl}. The area covered by buildings is smaller than the ground area giving an increase in the normalized PPV for buildings and a decrease for ground, compared to the nominal PPV.

Descriptions	PM (Fig. 8.3) [%]	PM + local (Fig. 8.4) [%]
PPV buildings (norm)	66.6 (88.6)	73.0 (91.3)
PPV ground (norm)	96.8 (88.6)	97.3 (90.4)
Building cells	12.3	13.8
Ground cells	21.7	25.8
Unclassified cells	55.5	52.4
Unknown cells (ties)	10.5	8.1

Table 8.1: Results of the evaluation of the predictive map, PM, displayed in Figure 8.3 and the fusion of PM and the local information in Figure 8.4. The last four rows show the actual proportions of the cells.

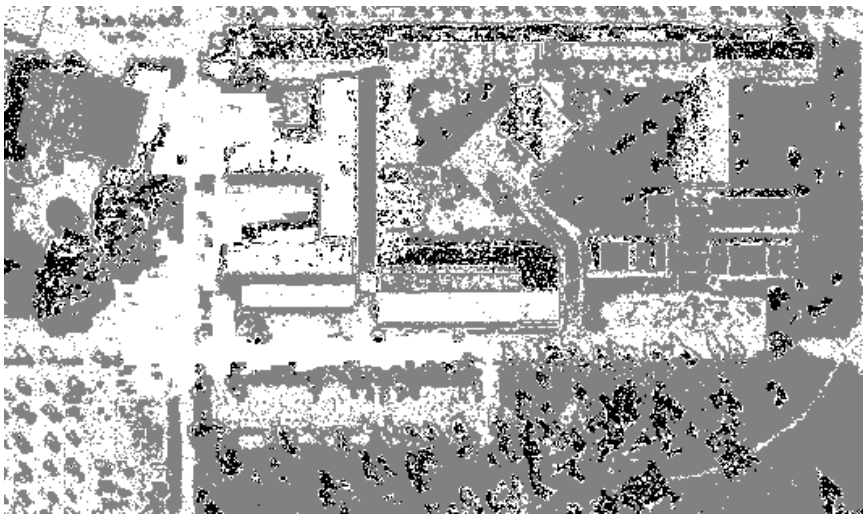
(a) Ground (gray) and building (black) estimates. The white cells are unexplored or unknown.



(b) Ties or unknown cells (black), not classified cells (gray), and classified cells (white).

Figure 8.3: The result of the global segmentation of the aerial image (see Section 8.4) using both ground and building models.

(a) Ground (gray) and building (black) estimates. The white cells are unexplored or unknown.



(b) Ties or unknown cells (black), not classified cells (gray), and classified cells (white).

Figure 8.4: The PM combined with the local information (see Section 8.5).

8.7 Summary and Conclusions

This chapter presented a method to segment aerial images with the purpose of extending the observation range of a mobile robot. The inputs to the method are information about ground taken from an occupancy grid map and building samples obtained from the local segmentation described in the previous chapter. The benefit from the extended range of the robot's view can clearly be noted in the presented example:

- The outline of the main building and additional building parts have been found.
- Most of the paths and roads have been found.
- The PM clearly shows the regions where more information needs to be collected in order to build a complete map.

In the local segmentation step it was noted that it can be hard to extract a complete building outline due to factors such as different roof materials, different roof inclinations and additions on the roof, specifically when the robot has only seen a small portion of the building outline. The global segmentation is a powerful extension here. Even though the roof structure in the example is quite complicated, the outline of a large building could be extracted based on the limited view of the mobile robot, which had only seen a minor part of surrounding walls.

The introduction of ground as a new class confirms the potential of using information from aerial images for planning tasks. Good estimates of where ground can be expected have been achieved and it is believed that planning algorithms can take advantage of this information to improve navigation in unknown environments.

8.7.1 Discussion

Oh *et al.* assumed that probable paths were known in a map and used this information to bias a robot motion model towards areas with higher probability of robot presence [Oh et al., 2004]. Using the approach suggested in this chapter these areas could be automatically found from aerial images.

Son *et al.* derived building structures from blueprints and correlate the information with satellite images [Son et al., 2007]. Dogruer *et al.* used manually extracted training samples to segment a satellite image for use in mobile robot localization and navigation [Dogruer et al., 2007]. Both these works are examples that could make use of our approach where the needed information, instead of being extracted from blueprints and manual samples, would be collected automatically by a mobile robot.

With the presented method, changes in the environment compared to an aerial image that is not perfectly up-to-date are handled to a certain degree. Assume that a building, present in the aerial image, has been removed after the image was taken. It may therefore be classified as a building in the PM if it had a roof colour similar to a building already detected by the mobile robot. When the robot approaches the area where the building was situated, the building will not be detected. If the mobile robot classifies the area as ground, the PM will turn into *unknown* (of course depending on c_{ij} and p), not only for that specific area but also globally, with the exception of areas where local information exists.

What about the other way around? Assume that a new building is erected and this is not yet reflected in the aerial image. If the wall matching indicates an edge as a wall this can, of course, introduce errors. However, there are several cases where it would not be a problem. When the area is cluttered, e.g., a forest, several close edges will be found and no segmentation is therefore performed. The same result is obtained if the building is erected in a smooth area, for example an open field, since there are no edges to be found. The result of these cases is that the building will only be present in the probabilistic semantic map in the form of a possible wall.

The uncertainties in the robot pose and association problems between the probabilistic semantic map and the aerial image have to be handled. The presented work requires that the robot information can be associated (registered) with the aerial image. This was solved by using global positioning provided by GPS. An alternative method for registration is to use multi-line matching.

Multi-line matching, in comparison to the single line matching used here, can relax the need for accurate localization of the mobile robot. An example of successful matching between ground readings and aerial image for localization is given in [Früh and Zakhor, 2004] and for matching of building outlines in [Beveridge and Riseman, 1997] and [Zhang et al., 2005].

Part IV
Conclusions

Chapter 9
Conclusions

This chapter summarizes and concludes the thesis. First, the achievements of the work are summarized. Second, the limitations of the suggested methods are discussed and finally some important directions for future work are given.

9.1 What has been achieved?

The research presented in this thesis deals with semantic mapping of outdoor environments for unmanned ground vehicles. The semantic information is acquired mainly by a vision-based virtual sensor, though laser-based information is also used. Semantic information from the virtual sensor is then combined with an ordinary occupancy grid map to construct a probabilistic semantic map. This map is used in turn as the link to information extracted from aerial images about both major obstacles in the form of buildings and ground that is potentially traversable by the vehicle.

Semantic Information Semantic information is typically used in the mobile robotics community in the context of human robot interaction (HRI). In HRI it is obvious that the semantics of a scene are crucial for successful operations. In order to be compatible with humans, the robots have to transform their sensor readings and relate them to human spatial concepts. Even though HRI is in the focus of attention, there are a number of other applications where semantics can play an important role, for example;

- semantic mapping [Wolf and Sukhatme, 2007, Calisi et al., 2007, Ross et al., 2006],
- execution monitoring to find problems in the execution of a plan [Bouguerra et al., 2007],
- localization through connection of human spatial concepts to particular locations [Galindo et al., 2005, Mozos et al., 2007],

- path following in conjunction with aerial images [Oh et al., 2004],
- improving 3D models of indoor environments [Nüchter et al., 2003, Weingarten and Siegwart, 2006, Nüchter et al., 2005], and
- model validation by forming the link between measurements and ground truth data [Wulf et al., 2007].

This list of applications shows the importance of semantic concepts in mobile robotics. Our work falls into the semantic mapping category and provides tools and algorithmic approaches that can be used also in the other application domains listed above.

Semantic Information Extraction To extract semantic information the concept of a virtual sensor based on visual input was developed. This virtual sensor is a generic approach that utilizes machine learning to adapt to given concepts. The virtual sensor makes use of a set of features extracted from gray scale images. The AdaBoost algorithm is used to select features and learn a classifier. The experiments showed that virtual sensors can be learned for several classes of objects using the same feature set. Virtual sensors for buildings, windows, trucks and nature were evaluated and the virtual sensor for buildings was used for building a probabilistic semantic map. The feature set can be extended to further improve the classifier and to handle more concepts.

The presented experiments demonstrated that the selected feature set handles variations in seasons and is robust towards the choice of camera. The suggested method using machine learning and generic image features makes it possible to extend virtual sensors to a range of other important human spatial concepts.

Probabilistic Semantic Mapping In Chapter 5, a method to build probabilistic semantic maps based on semantic and occupancy information is presented. A virtual sensor for pointing out buildings along a mobile robot's track is used together with information from an ordinary occupancy grid map. The method handles the wide field-of-view of the planar camera ($56°$), which may contain different sized objects that can belong to different classes. Despite the large uncertainty about the location of the classified object in the image, a very accurate semantic map is produced.

From the experiments described in Chapter 5, several benefits of using the virtual sensor with its good generalisation properties are noted. The approach was found to be very robust even though

- the training set was quite small,
- different resolutions were used in the training phase and in the map building phase, and

- different cameras were used in the training phase and in the map building phase.

Aerial Image Segmentation The results from the probabilistic semantic map are used as the link to building outlines in aerial images. Both local and global segmentation in the aerial image are performed in order to detect buildings and ground. Fusion with ground information, which is found from an occupancy grid map, then results in a local information map and a predictive map (see description of the relations between the semantic maps below).

With this approach two difficulties are addressed simultaneously: 1) buildings are hard to detect in aerial images without elevation data and 2) the limitation in range of the sensors onboard the mobile robots. Concerning the first difficulty the results show a high classification rate and we can therefore conclude that the ground-level semantic information can be used to compensate for the absence of elevation data in aerial image segmentation. Second, the information gained from the aerial image segmentation results in an extended range of the robot's view into areas that have not been visited by the robot. The benefits from this extended range include

- detection of building outlines and areas that could be obstacles,
- detection of paths and roads that are potentially driveable and therefore important from a planning perspective, and
- clear indication of where more information needs to be collected in order to complete the maps.

Relation Between the Semantic Maps Three different types of grid maps containing semantic information have been defined in this thesis:

The probabilistic semantic map is a grid map where cell values in the interval $[0, 1]$ represent the probability that the cell belongs to a particular class. The probabilistic semantic map is based on occupancy information and it is only the outline of objects that are represented. The probabilistic semantic map is presented in Chapter 5.

The local information map is a grid map where occupied cells represent regions belonging to a semantic class. By contrast with the probabilistic semantic map, the cells in the local information map represent the estimated area of objects and not only their partial outlines. All classified object regions are spatially connected to the input information used in the creation of the map. The local information map is described in Chapter 7.

The predictive map, PM, is a global semantic grid map that predicts the presence of different classes in an entire aerial image. The classified regions are therefore not necessarily spatially connected with the input information, but rather found via the colour models utilized in the segmentation process. The PM is described in Chapter 8.

The probabilistic semantic map is only based on ground level information collected by the mobile robot while the local information map shows objects of particular classes close to the robot's path, based also on information extracted from aerial images. The PM predicts the surroundings of the mobile robot in a much larger area. The quality of these predictions depends on how homogeneous the aerial image is, i.e., if the colour models manage to separate one class from the other classes. The PM is based on classified regions in the local information map, which in turn is based on information from the probabilistic semantic map. The local information map therefore has precedence over the PM in the sense that in the case of conflicting entries, the local information map is treated as being more credible.

9.2 Limitations

The key objectives of the work performed were outdoor semantic mapping and to show how semantic information can be used in conjunction with aerial imagery to extend the robot's understanding of the surrounding world. The presented system has deliberately been divided into three modules with the purpose of making parts of the system exchangeable. In this way three objectives are addressed. First of all, the system can be adapted to utilize other sensor modalities. Second, the system can be easily improved through improvement of the individual modules. Third, the modules may be used individually for situations when only parts of the system are needed. With this said, the limitations of the system can be discussed and it is my belief that these shortcomings can be handled by future research and work.

Starting with the virtual sensor, the main limitation is that the classification is performed on the entire image. Even though the results from using this virtual sensor in the probabilistic semantic map were promising, improved and more robust results can be expected if the virtual sensor was extended so that it could define which parts of an image belong to the specified class. Methods to overcome this issue include windowing techniques, where windows of varying size divide the image into sub-images that are individually classified. By this the location of the object in the image could be estimated more accurately. The virtual sensor has showed resolution independence within a certain range, which indicates that it can be used on sub-images.

The probabilistic semantic map currently handles two classes. One can foresee that in the future a need for more classes will arise. It should be possible

to handle more classes by building one semantic map for each class of interest utilizing the virtual sensors for the respective classes, and letting these maps represent separate layers. The different maps can then be fused into one single semantic map, e.g., by the use of the same fusing technique as for the PM.

In the evaluation of the probabilistic semantic map it was noted that the occupancy grid map where the data have been collected in a horizontal plane close to the ground is not optimal for finding building outlines. Therefore, to build accurate maps that include objects of a certain vertical scale, it is recommended that the sensor readings should be able to reflect objects within that scale. Alternative solutions include the use of 3D-lasers, e.g., vertically mounted laser range scanners, or vision only solutions using stereo cameras or motion stereo.

The derived semantic maps are limited to two dimensions, i.e., they are 2D maps with the 3D-world information projected onto a layer at ground level. Situations where the 2D-information is not enough to describe the environment may therefore occur, for instance when information from several overlapping spatial layers is of interest. Consider a situation where the mobile robot detects driveable ground under trees, while the aerial image only shows the trees, since only the top layer is visible from above. If the robot detects that objects such as tree tops may obstruct the view of the ground in the aerial image, training samples for colour models of ground cannot be taken in these areas. If the forest is considered to be dense at tree top level, colour models of trees can be extracted. One way to represent this type of overlapping information is to introduce layered maps.

9.3 Future Work

Based on the limitations of the system presented and the suggestions mentioned earlier in the respective chapters, the most promising directions for future work are discussed for each module as follows.

Virtual Sensor The introduction of a method to refine the virtual sensor so that it can point out which parts of an image are mainly responsible for the classification could be used to further improve the separation of different classes in the probabilistic semantic map. To refine the virtual sensor, classification of sub-images could be performed, exploiting the robustness of the virtual sensor to changes in resolution. An example of a sub-image technique is described in [Morita et al., 2005] where small squares of the upper half of images are classified as tree, building and uniform regions.

Probabilistic Semantic Mapping A natural extension of the work on the probabilistic semantic map is to introduce other object classes and refined ground

classification. An example would be drivable areas that can be detected using the onboard sensor system.

Aerial Image Segmentation The segmentation to produce the predictive map is pixel based. It is therefore likely that post-processing of the PM, e.g., with filters taking neighbouring cells into account, could improve the results. Post-processing with Hidden Markov Models and Markov Random Field has been used in similar applications [Mozos et al., 2006, Wolf and Sukhatme, 2007].

It is further expected that shadow detection, which merges shadowed areas with corresponding areas in the sun, can reduce the number of false positive building pixels from the segmentation and decrease unknown areas caused by ties.

Multi-line matching should be evaluated as an alternative method for registration. Multi-line matching can relax the demands on accurate global navigation of the mobile robot. An example of successful matching between ground readings and aerial image for localization is given in [Früh and Zakhor, 2004] and for matching of building outlines in [Beveridge and Riseman, 1997].

The accuracy of the PM can probably be further improved by using a measure of the colour model quality to assign a value to the parameter p (the quality parameter for the layers in the PM), see Chapter 8. Also the probabilities from the semantic map from which the ground wall estimates are extracted and the certainty of the virtual sensor could be used in the calculation of p. When the parameter p is in use, alternatives to the max-function used to update the individual layers should be investigated.

The use of automatic systems in different forms will be more and more common in future. The area of mobile robotics is expected to grow fast and the trends in the automotive industry aim at more automatic functions and will probably result in driverless cars. If these types of systems can understand human concepts their usability will be greatly improved and it is therefore expected that utilization and extraction of semantic information will play an important role in the future.

Aerial images are nowadays globally available via the Internet and several types of systems can make benefit of the rich information they contain. Utilization of overhead information in the form of aerial images brings, in combination with semantic knowledge, a new dimension to mobile robot mapping. The examples given in this thesis show the potential of semantic maps that can be used for planning and exploration.

In this sense, the ideas and work presented in this thesis take us one step further towards systems that use multimodal inputs and transform their internal representations into human spatial concepts.

Part V
Appendices

Appendix A
Notation and Parameters

A.1 Abbreviations

Abbreviations used in the thesis are explained in the following.

AdaBoost	Adaptive Boosting
BIRON	Bielefeld Robot Companion
BOC	Bayes Optimal Classifier
CCD	Charge-Coupled Device (electronic light sensor)
CM	Colour Model
DGPS	Differential GPS
DEM	Digital Elevation Model
ESOM	Ensemble of Self-Organizing Maps
GBIS	Graph-Based Image Segmentation
GIS	Geographical Information System
GMM	Gaussian Mixture Model
GPS	Global Positioning System
HMM	Hidden Markov Model
HRI	Human-Robot Interaction
HSI	Hue, Saturation, Intensity (colour space)
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
INS	Integrated Navigation System
IR	Infra-Red
JPEG	Joint Photographic Experts Group (still image compression standard)
LIDAR	Light Detection and Ranging

MCL	Monte Carlo Localization
MDC	Minimum Distance Classifier
MRF	Markov Random Fields
PCA	Principal Components Analysis
PM	Predictive Map
PPV	Positive Predictive Value
PT	Pan-Tilt
RFCH	Receptive Field Cooccurence Histograms
RGB	Red, Green, Blue (colour space)
SAR	Synthetic Aperture Radar
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SLR	Single-Lens Reflex (camera)
SSH	Spatial Semantic Hierarchy
SVM	Support Vector Machine
TIFF	Tagged Image File Format
UAV	Unmanned Aerial Vehicle
VS	Virtual Sensor
WGS	World Geodetic System, WGS84 is the latest revision

A.2 Parameters

Below is a list of parameters and their notation used in equations etc. throughout the thesis. The values of the parameters are presented where appropriate.

α_i	covering angle of object i in a sector	
β_{dev}	corner tolerance [deg]	± 20
Δ	planar camera field-of-view [deg]	56
Φ_{TP}	true positive rate	
Φ_{TN}	true negative rate	
Φ_{TP}	false positive rate	
Φ_{FN}	false negative rate	
θ	sector opening angle [deg]	30-56
$\vec{\theta}$	edge orientations	
A_{start}	start area [pixels]	8×8

C_{45}	number of corners with direction of $90n + 45$ degrees, $n \in 1, \ldots, 4$	
d	Mahalanobis distance	
$\boldsymbol{D}_t$	distribution that weight training samples in AdaBoost	
f_x	feature number x	
h_t	the best weak classifier from iteration t in AdaBoost	
H	the strong classifier (AdaBoost)	
$\vec{H}_x$	histogram with x bins	
I_e	edge image	
L_{VS}	VS maximum range [m]	50
n	number of objects in a view	
N	number of instances	
N_{pv}	number of planar views	8
N_{run}	number of Monte Carlo runs	
P(build|VS=build)	building probability given that the VS indicates building	> 0.5
P(build|VS=$\neg$build)	1 - nonbuilding probability given that the VS indicates nonbuilding	< 0.5
P_{cell}	value of a cell in a grid map	
s	sensor reading	
t	iteration (AdaBoost)	
T	max number of iterations (AdaBoost)	30
T_s	number of sensor readings	

Appendix B
Implementation Details

B.1 Line Extraction

This section gives a short overview of the line extraction implementation used to find straight line segments in a binary image. This binary image may be the result of an edge detection operation but can also originate from other sources.

For line extraction, Matlab functions implemented by Peter Kovesi [Kovesi, 2000] have been used. These functions are implemented in m-files described in the following:

edgelink.m This function links edge points in a binary image into chains. If an edge diverges at a junction the function tracks one of the branches. Broken branches can be remerged by *mergeseg.m*.

lineseg.m This function forms straight line segments from the edge list calculated by *edgelink.m*. The function breaks down each array of edgepoints in the edge list to straight lines that fulfill the tolerance TOL. An edge merging phase is performed to connect lines that may have been separated in the edge linking phase.

mergeseg.m Function used by *lineseg.m*. The function scans through the list of line segments to check if any segments can be merged. Segments are merged if the orientation difference is less than ANGTOL and if the ends of the segments are within LINKRAD of each other.

The parameter setting for the line extraction, as used in the experiments, is described in Table B.1.

Name	Value	Description
TOL	2 pixels	Maximum deviation from a straight line before a segment is broken in two
ANGTOL	0.05 rad	Angle tolerance used when attempting to merge line segments
LINKRAD	2 pixels	Maximum distance between end points of line segments for segments to be eligible for linking

Table B.1: Parameters used for line extraction.

B.2 Geodetic Coordinate Transformation

The GPS receiver outputs position data in the form of longitude and latitude coordinates represented in WGS84. WGS84 is the world geodetic system dating from 1984. It is the reference system used by GPS and uses a reference ellipsoid representing the earth.

The WGS84 coordinates are appropriate for a system that covers the whole earth. In smaller areas a metric system is preferable. The aerial images are registered in local coordinate systems, closely connected to the Swedish coordinate system RT90 2.5 gon V 0:-15. The WGS position data therefore have to be transformed. For this a transformation function from SWEREF 99 to RT90 has been used. WGS 84 and SWEREF 99 are, in principle, interchangeable[1]. The difference is in the order of 0.1 m but these systems are slowly diverging due to the motion of the European plate.

The transformation makes use of Gauss conformal projection according to [Lantmäteriet, 2005]. The parameters used by the transformation are listed in Table B.2.

B.3 Localization

The robot was localised using DGPS and odometry. Since the trajectory is close to buildings, the positions from the DGPS suffer from multipath signals resulting in errors from a few meters up to 100 m. When these errors occur intermittently they can be filtered out using a motion model of the mobile robot. The robot can only move with a certain velocity and therefore many of the erroneous positions can be removed. The main problem occurs when the multipath signals result in a constant shift of the position. If the start of the shift is not detected it is no longer possible to remove the false positions.

[1] National land survey of Sweden, www.lantmateriet.se

Parameter	Value
Type of projection	Transverse Mercator (Gauss-Krüger)
Reference ellipsoid	GRS 80
Semi-major Axis (a)	6378137
Inverse flattening (1/f)	298.257222101
Central meridian	15°48'22".624306 East Greenwich
Latitude of origin	0°
Scale on central meridian	1.00000561024
False Northing, x_0	-667.711 m
False Easting, y_0	1500064.274 m

Table B.2: Parameters used for transformation from SWEREF99 to RT90.

The pose data used in the experiments of this thesis have been manually calibrated in the following way. Using only five parameters, the odometry data have been tuned to fit with the DGPS positions. As has been discussed above odometry suffers from drift. However, it is possible to achieve accurate positions by calibration of the odometry as long as the properties of the ground surface are constant and the trajectory length is limited. The five parameters that were used in the calibration are:

- Start position in X (north)
- Start position in Y (east)
- Initial heading
- Length scale
- Turn scale

The first three parameters are used to set the position and orientation of the robot at the first GPS-fix. The last two parameters handle the wheel sizes (length scale) and the difference in left and right wheel radii (turn scale). The length scale is used as a factor multiplied to the odometry increments before integration. The turn scale parameter is multiplied with the current velocity and added to the heading increment before integration. For the performed experiments this gives good pose estimates of the robot.

The robot heading is calculated from the obtained trajectory. The calibrated pose data are linearly interpolated to obtain the pose for each image. This is not a limitation since the images are taken either during forward motion or when the robot does not move.

Bibliography

Iyad Abuhadrous, Samer Ammoun, Fawzi Nashashibi, François Goulette, and Claude Laurgeau. Digitizing and 3D modeling of urban environments and roads using vehicle-borne laser scanner system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 76–81, Sendai, Japan, September 28 – October 2 2004.

Peter Allen, Ioannis Stamos, Atanas Gueorguiev, Ethan Gold, and Paul Blaer. AVENUE: Automated site modeling in urban environments. In *Proceedings of 3rd Conference on Digital Imaging and Modeling*", pages 357–364, Quebec City, Canada, May 2001.

Drago Anguelov, Rahul Biswas, Daphne Koller, Benson Limketkai, Scott Sanner, and Sebastian Thrun. Learning hierarchical object maps of non-stationary environments with mobile robots. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 10–17, Edmonton, Canada, August 2002.

Drago Anguelov, Daphne Koller, Evan Parker, and Sebastian Thrun. Detecting and modeling doors with mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3777–3784, New Orleans, LA, USA, April 26-May 1 2004.

Ashtech. *G12 GPS OEM board & sensor reference manual*. Ashtech, Sepember 2000.

Patrick Beeson, Nicholas K. Jong, and Benjamin Kuipers. Towards autonomous topological place detection using the extended voronoi graph. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4384–4390, Barcelona, Spain, April, 18–22 2005.

Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):239–256, February 1992. ISSN 0162-8828.

J. Ross Beveridge and Edward M. Riseman. How easy is matching 2D line models using local search? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):564–579, 1997.

Johan Bos, Ewan Klein, and Tetsushi Oka. Meaningful conversation with a mobile robot. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 71–74, Budapest, Hungary, 2003. Association for Computational Linguistics. ISBN 1-111-56789-0.

Abdelbaki Bouguerra, Lars Karlsson, and Alessandro Saffiotti. Semantic knowledge-based execution monitoring for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3693–3698, Rome, Italy, April 10 2007.

Pär Buschka and Alessandro Saffiotti. A virtual sensor for room detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 637–642, Lausanne, CH, 2002.

Pär Buschka. *An Investigation of Hybrid Maps for Mobile Robots*. PhD thesis, Dept. of Technology, Örebro University, Sweden, February 2006.

Daniele Calisi, Alessandro Farinelli, Giorgio Grisetti, Luca Iocchi, Daniele Nardi, Stefano Pellegrini, Diego Tipaldi, and Vittorio Amos Ziparo. Contextualization in mobile robots. In *Proceedings of the ICRA 2007 Workshop on Semantic Information in Robotics*, Rome, Italy, April 10 2007.

John Canny. A computational approach for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):279–98, November 1986.

Luiz Alberto Cardoso. Computer aided recognition of man-made structures in aerial photographs. Master's thesis, Naval postgraduate school, Monterey, California, 1999. URL `http://www.cs.nps.navy.mil/people/faculty/rowe/cardosothesis.htm`.

Robert W. Carroll. Detecting building changes through imagery and automatic feature processing. In *URISA 2002 GIS & CAMA Conference Proceedings*, Reno, Nevada, USA, April 7–10 2002.

Cheng Chen and Han Wang. Large-scale loop-closing with pictorial matching. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1194–1199, Orlando, Florida, May 2006.

Heng-Da Cheng, Xihua Jiang, Y. Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259–2281, 2001.

Kok Seng Chong and Lindsay Kleeman. Sonar based map building for a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1700–1705, Albuquerque, New Mexico, USA, April 1997.

Matthieu Cord, Michel Jordan, Jean-Pierre Cocquerez, and Nicolas Paparoditis. Automatic extraction and modelling of urban buildings from high resolution aerial images. In *Proceedings of ISPRS Automatic Extraction of GIS Objects from Digital Imagery*, pages 187–192, Munich, Germany, Sepember 1999.

Crossbow. *DMU User's manual*. Crossbow Technology, Inc., 2001.

Hendrik Dahlkamp, Adrian Kaehler, David Stavens, Sebastian Thrun, and Gary Bradski. Self-supervised monocular road detection in desert terrain. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 16–19 2006.

Frank Dellaert and David Bruemmer. Semantic SLAM for collaborative cognitive workspaces. In *AAAI Fall Symposium Series 2004: Workshop on The Interaction of Cognitive Science and Robotics: From Interfaces to Intelligence*, 2004.

Can Ulas Dogruer, Bugra Koku, and Melik Dolen. A novel soft-computing technique to segment satellite images for mobile robot localization and navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2077–2082, San Diego, CA, USA, October 29 – November 2 2007.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, second edition, 2001.

Staffan Ekvall, Patric Jensfelt, and Danica Kragic. Integrating active mobile robot object recognition and SLAM in natural environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5792–5797, Beijing, China, October 9–15 2006.

Ahmed El-Rabbany. *Introduction to GPS: the Global Positioning System*. Artech House Publishers, Boston, London, 2002.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM (Association for Computing Machinery)*, 24(6):381–395, June 1981.

Jordi Freixenet, Xavier Munoz, David Raba, Joan Marti, and Xavier Cufi. Yet another survey on image segmentation: Region and boundary information integration. In *European Conference on Computer Vision*, volume 3, pages 408–422, Copenhagen, Denmark, May 2002.

Christian Freksa, Reinhard Moratz, and Thomas Barkowsky. Schematic maps for robot navigation. In C. Freksa, W. Brauer, C. Habel, and K. Wender, editors, *Spatial Cognition II: Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications*, pages 100–114. Berlin: Springer, 2000.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Stephen Friedman, Hanna Pasula, and Dieter Fox. Voronoi random fields: Extracting the topological structure of indoor environments via place labeling. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2109–2114, Hyderabad, India, January 6–12 2007.

Christian Früh and Avideh Zakhor. An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision*, 60(1):5–24, 2004.

Christian Früh and Avideh Zakhor. Constructing 3D city models by merging aerial and ground views. *IEEE Computer Graphics and Applications*, 23(6): 52–61, Nov/Dec 2003.

Christian Früh and Avideh Zakhor. Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images. In *Proceedings of First Symposium on 3D Data Processing, Visualization and Transmission*, pages 834–847, Padua, Italy, June 2002.

Christian Früh and Avideh Zakhor. Fast 3D model generation in urban environments. In *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 165–170, Baden-Baden, Germany, August 2001.

Cipriano Galindo, Alessandro Saffiotti, Silvia Coradeschi, Pär Buschka, Juan-Antonio Fernández-Madrigal, and Javier González. Multi-hierarchical semantic maps for mobile robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3492–3497, Edmonton, Alberta, Canada, 2005.

Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. Using semantic information for improving efficiency of robot task planning. In *Proceedings of the ICRA 2007 Workshop on Semantic Information in Robotics*, Rome, Italy, April 10 2007.

M. García-Alegre, Angela Ribeiro, Lia García-Pérez, Rene Martínez, Domingo Guinea, and Ana Pozo-Ruz. Autonomous robot in agriculture tasks. In *3rd European Conference on Precision Agriculture*, pages 25–30, Montpellier, 2001.

Atanas Georgiev and Peter K Allen. Localization methods for a mobile robot in urban environments. *IEEE Transactions on Robotics*, 20(5):851–864, 2004.

Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing*. Prentice-Hall, 2002. ISBN 0-201-50803-6.

Jose J. Guerrero and Carlos Sagüés. Robust line matching and estimate of homographies simultaneously. In *Pattern Recognition and Image Analysis: First Iberian Conference, IbPRIA 2003*, pages 297–307, Puerto de Andratx, Mallorca, Spain, June 2003. ISBN 3-540-40217-9.

Yanlin Guo, Harpreet Sawhney, Rakesh Kumar, and Steve Hsu. Learning-based building outline detection from multiple aerial images. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 545–552, Los Alamitos, CA, USA, 2001.

Ying Guo, Vadim Gerasimov, and Geoff Poulton. Vision-based drivable surface detection in autonomous ground vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3273–3278, Beijing, China, October 9–15 2006.

Richard I. Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

Andrew Howard, Denis F. Wolf, and Gaurav S. Sukhatme. Towards 3D mapping in large urban environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1258–1263, Sendai, Japan, August 2004.

Johannes Huber and Volker Graefe. Motion stereo for mobile robots. *IEEE Transactions on Industrial Electronics*, 41(4):378–383, 1994.

Patric Jensfelt. *Approaches to Mobile Robot Localization in Indoor Environments*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2001.

Seungdo Jeong, Jounghoon Lim, Il Hong Suh, and Byung-Uk Choi. Vision-based semantic-map building and localization. In Bogdan Gabrys, Robert J. Howlett, and Lakhmi C. Jain, editors, *KES (1)*, volume 4251 of *Lecture Notes in Computer Science*, pages 559–568. Springer, 2006. ISBN 3-540-46535-9.

Robert E. Karlsen and Gary Witus. Terrain understanding for robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 895–900, San Diego, CA, USA, October 2007.

Yong-Shik Kim, Bong Keun Kim, Kohtaro Ohba, and Akihisa Ohya. A data integration for localization with different-type sensors. In *The 13th International Conference on Advanced Robotics (ICAR)*, pages 773–778, Jeju, Korea, August 21–24 2007.

Ron Kohavi. A study of cross-validation and bootstrap for accuracy esimation and model selection. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1145, Montreal, Canada, 1995.

Peter D. Kovesi. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia, 2000. Last checked November 2007. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

Benjamin Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119: 191–233, May 2000.

KVH. *C100 Compass Engine, Technical manual*. KVH Industries, Inc., 1998.

John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289, Williamstown, MA, USA, June 28 – July 1 2001. ISBN 1-55860-778-1.

Lantmäteriet. Gauss conformal projection (transverse mercator), Krügers formulas. <http://www.lantmateriet.se>, August 31 2005.

Stephen Levitt and Farzin Aghdasi. Fuzzy representation and grouping in building detection. In *Proceedings 2000 International Conference on Image Processing*, volume 3, pages 324–327, Vancouver, BC, Canada, September 10–13 2000.

Yi Li and Linda G. Shapiro. Consistent line clusters for building recognition in CBIR. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pages 952–957, Quebec City, Quebec, Canada, August 2002.

Benson Limketkai, Lin Liao, and Dieter Fox. Relational object maps for mobile robots. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1471–1476, Edinburgh, Scotland, UK, July 30 – August 5 2005.

Chung-An Lin. *Perception of 3-D Objects from an Intensity Image Using Simple Geometric Models*. PhD thesis, Faculty of the Graduate School, University of Southern California, USA, December 1996.

Jovanka Malobabic, Herve Le Borgne, Noel Murphy, and Noel O'Connor. Detecting the presence of large buildings in natural images. In *Proceedings of the 4th International Workshop on Content-Based Multimedia Indexing*, Riga, Latvia, June 21–23 2005.

Helmut Mayer. Automatic object extraction from aerial imagery – a survey focusing on buildings. *Computer Vision and Image Understanding*, 74(2): 138–149, May 1999.

Nicholas Metropolis and Stanislaw Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, September 1949.

Hans P. Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 116–121, St. Louis, Missouri, USA, 1985.

Hideo Morita, Michael Hild, Jun Miura, and Yoshiaki Shirai. View-based localization in outdoor environments based on support vector learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3083–3088, Edmonton, Alberta, Canada, 2005. IEEE.

Hideo Morita, Michael Hild, Jun Miura, and Yoshiaki Shirai. Panoramic view-based navigation in outdoor environments based on support vector learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2302–2307, Beijing, China, October 9–15 2006. IEEE.

Óscar Martínez Mozos. Supervised learning of places from range data using AdaBoost. Master's thesis, University of Freiburg, Freiburg, Germany, 2004.

Óscar Martínez Mozos, Cyrill Stachniss, and Wolfram Burgard. Supervised learning of places from range data using AdaBoost. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1742–1747, Barcelona, Spain, April 2005.

Óscar Martínez Mozos, Axel Rottmann, Rudolph Triebel, Patric Jensfelt, and Wolfram Burgard. Semantic labeling of places using information extracted from laser and vision sensor data. In *Proceedings of the IROS 2006 Workshop: From Sensors to Human Spatial Concepts*, pages 33–39, Beijing, China, October 10 2006.

Óscar Martínez Mozos, Patric Jensfelt, Hendrik Zender, Geert-Jan M. Kruijff, and Wolfram Burgard. From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots. In *Proceedings of the ICRA 2007 Workshop on Semantic Information in Robotics*, Rome, Italy, April 10 2007.

Marina Mueller, Karl Segl, and Hermann Kaufmann. Edge- and region-based segmentation technique for the extraction of large, man-made objects in high-resolution satellite imagery. *Pattern Recognition*, 37:1621–1628, 2004.

Ramakant Nevatia, Chungan Lin, and Andres Huertas. A system for building detection from aerial images. In A. Gruen, E. P. Baltsavias, and O. Henricsson, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 77–86. Birkhäuser Verlag, 1997.

Curtis W. Nielsen, Bob Ricks, Michael A. Goodrich, David Bruemmer, Doug Few, and Miles Walton. Snapshots for semantic maps. In *Proceedings of the 2004 IEEE Conference on Systems, Man, and Cybernetics*, volume 3, pages 2853–2858, The Hague, The Netherlands, October 10-13 2004.

Andreas Nüchter, Hartmut Surmann, Kai Lingemann, and Joachim Hertzberg. Semantic scene analysis of scanned 3D indoor environments. In *Proceedings of the 8th International Fall Workshop Vision, Modeling, and Visualization 2003*, pages 215–222, Munich, Germany, November 2003. IOS Press. ISBN 3-89838-048-3.

Andreas Nüchter, Oliver Wulf, Kai Lingemann, Joachim Hertzberg, Bernardo Wagner, and Hartmut Surmann. 3D mapping with semantic knowledge. In *Proceedings of the RoboCup International Symposium 2005*, pages 335–346, Osaka, Japan, July 2005.

Sang Min Oh, Sarah Tariq, Bruce N. Walker, and Frank Dellaert. Map-based priors for localization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, pages 2179–2184, Sendai, Japan, 2004.

Kazunori Ohno, Takashi Tsubouchi, Bunji Shigematsu, and Shin'ichi Yuta. Differential GPS and odometry-based outdoor navigation of a mobile robot. *Advanced Robotics*, 18(6):611–635, 2004.

Martin Persson, Mats Sandvall, and Tom Duckett. Automatic building detection from aerial images for mobile robot mapping. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 273–278, Espoo, Finland, June 2005. ISBN 0-7803-9355-4.

Fiora Pirri. Indoor environment classification and perceptual matching. In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 73–84, Whistler, Canada, June 2–5 2004. AAAI Press. ISBN 1-57735-199-1.

Fabio T. Ramos, Ben Upcroft, Suresh Kumar, and Hugh F. Durrant-Whyte. Recognising and segmenting objects in natural environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5866–5871, Beijing, China, October 9–15 2006. IEEE.

Robert J. Ross, Christian Mandel, John A. Bateman, Shi Hui, and Udo Frese. Towards stratified spatial modeling for communication & navigation. In *Proceedings of the IROS 2006 Workshop: From Sensors to Human Spatial Concepts*, pages 41–46, Beijing, China, October 10 2006.

Axel Rottmann, Óscar Martínez Mozos, Cyrill Stachniss, and Wolfram Burgard. Place classification of indoor environments with mobile robots using boosting. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1306–1311, Pittsburgh, PA, USA, 2005.

Michel Roux and Henri Maître. Three-dimensional description of dense urban areas using maps and aerial images. In A. Gruen and H. Li, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, pages 311–322. Birkhäuser Verlag, 1997.

Mark A. Ruzon and Carlo Tomasi. Color edge detection with the compass operator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 160–166, Ft. Collins, CO, USA, June 23–25 1999.

Robert E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1401–1406, Stockholm, Sweden, July 31 – August 6 1999.

Klaus-Jürgen Schilling and Thomas Vögtle. An approach for the extraction of settlement areas. In A. Gruen and H. Li, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, pages 333–342. Birkhäuser Verlag, 1997.

Chris Scrapper, Ayako Takeuchi, Tommy Chang, Tsai Hong, and Michael Shneier. Using a priori data for prediction and object recognition in an autonomous mobile vehicle. In Grant R. Gerhart, Charles M. Shoemaker, and Douglas W. Gage, editors, *Unmanned Ground Vehicle Technology V. Proceedings of the SPIE, Volume 5083*, pages 414–418, September 2003. doi: 10.1117/12.485917.

David Silver, Boris Sofman, Nicolas Vandapel, J. Andrew Bagnell, and Anthony Stentz. Experimental analysis of overhead data processing to support long range navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2443–2450, Beijing, China, October 9–15 2006.

Marjorie Skubic, Pascal Matsakis, George Chronis, and James Keller. Generating multi-level linguistic spatial descriptions from range sensor readings using the histogram of forces. *Autonomous Robots*, 14(1):51–69, January 2003.

Ulf Söderman, Simon Ahlberg, Åsa Persson, and Magnus Elmqvist. Towards rapid 3D modelling of urban areas. In *Proceeding of the Second Swedish-American Workshop on Modeling and Simulation, (SAWMAS-2004)*, Cocoa Beach, FL, USA, February 2004.

Kil-ho Son, Ji-hwan Woo, and In-So Kweon. Automatic 3D urban modeling from satellite image. In *The 13th International Conference on Advanced Robotics (ICAR)*, pages 936–940, Jeju, Korea, August 21–24 2007.

Dezhen Song, Hyun Nam Lee, Jingang Yi, and Anthony Levandowski. Vision-based motion planning for an autonomous motorcycle on ill-structured road. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3279–3286, Beijing, China, October 9–15 2006.

Thorsten Spexard, Shuyin Li, Britta Wrede, Jannik Fritsch, Gerhard Sagerer, Olaf Booij, Zoran Zivkovic, Bas Terwijn, and Ben Kröse. BIRON, where are you? - Enabling a robot to learn new places in a real home environment by integrating spoken dialog and visual localization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 934–940, Beijing, China, October 9–15 2006. IEEE.

Cyrill Stachniss. *Exploration and Mapping with Mobile Robots*. PhD thesis, University of Freiburg, Department of Computer Science, Freiburg, Germany, April 2006.

Cyrill Stachniss, Óscar Martínez Mozos, and Wolfram Burgard. Speeding-up multi-robot exploration by considering semantic place information. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1692–1697, Orlando, FL, USA, May 2006.

Hartmut Surmann, Kai Lingemann, Andreas Nüchter, and Joachim Hertzberg. A 3D laser range finder for autonomous mobile robots. In *Proceedings of the 32nd International Symposium on Robotics (ISR)*, pages 153–158, April 19–21 2001.

The MathWorks. Matlab 7.0, including Image Processing Toolbox 5.0. <http://www.mathworks.com>.

Christian Theobalt, Johan Bos, Tim Chapman, Arturo Espinosa-Romero, Mark Fraser, Gillian Hayes, Ewan Klein, Tetsushi Oka, and Richard Reeve. Talking to Godot: Dialogue with a mobile robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1338–1343, Lausanne, Switzerland, September 30 – October 4 2002.

Sebastian Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

Sebastian Thrun, Arno Bücken, Wolfram Burgard, Dieter Fox, Thorsten Fröhlinghaus, Daniel Henning, Thomas Hofmann, Michael Krell, and Timo Schmidt. Map learning and high-speed navigation in RHINO. In David Kortenkamp, R. Peter Bonasso, and Robin Murphy, editors, *Artificial intelligence and mobile robots: case studies of successful robot systems*, pages 21–52. AAAI Press / The MIT Press, 1998.

Kinh Tieu and Paul Viola. Boosting image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 228–235, Hilton Head Island, South Carolina, USA, June 2000.

Ricardo A. Téllez and Cecilio Angulo. Acquisition of meaning through distributed robot control. In *Proceedings of the ICRA 2007 Workshop on Semantic Information in Robotics*, Rome, Italy, April 10 2007.

Elin A. Topp and Henrik I. Christensen. Topological modelling for human augmented mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2257–2263, Beijing, China, October 9–15 2006.

Antonio Torralba, Kevin P. Murphy, William T. Freeman, and Mark A. Rubin. Context-based vision system for place and object recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 273–280, Nice, France, October 13–16 2003.

André Treptow, Andreas Masselli, and Andreas Zell. Real-time object tracking for soccer-robots without color information. In *European Conference on Mobile Robotics (ECMR)*, pages 33–38, Radziejowice, Poland, 2003.

Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proceedings of the 2006*

IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2276–2282, Beijing, China, October 9–15 2006.

Florence Tupin and Michel Roux. Detection of building outlines based on the fusion of SAR and optical features. *ISPRS Journal of Photogrammetry & Remote Sensing*, 58:71–82, 2003.

Iwan Ulrich and Illah Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *AAAI National Conference on Artificial Intelligence*, pages 866–871, Austin, TX, USA, July 30 – August 3 2000.

US Army. Map reading and land navigation. Online version of field manual No. 3-25.26, US Army, July 2001. Last checked February 2008. Available from: <http://www.globalsecurity.org/military/library/policy/army/fm/3-25-26/index.html>.

Dominique v. Zwynsvoorde, Thierry Siméon, and Rachid Alami. Incremental topological modeling using local Voronoi-like graphs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 897–902, Takamatsu, Japan, October 30 – November 5 2000.

Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

Jan Weingarten and Roland Siegwart. 3D SLAM using planar segments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3062–3067, Beijing, China, October 9–15 2006. IEEE.

Denis F. Wolf and Gaurav S. Sukhatme. Semantic mapping using mobile robots. *Accepted for publication in the IEEE Transactions on Robotics*, 2007.

Oliver Wulf, Andreas Nüster, Joachim Hertzberg, and Bernardo Wagner. Ground truth evaluation of large urban 6D SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 650–657, San Diego, CA, USA, October 29 – November 2 2007.

Rongrui Xiao, Chris Lesher, and Bob Wilson. Building detection and localization using a fusion of interferometric synthetic aperture radar and multispectral image. In *Proceedings of the ARPA Image Understanding Workshop*, pages 583–588, Monterey, CA, USA, November 20–23 1998.

Aiwu Zhang, Shaoxing Hu, Xin Jin, and Weidong Sun. A method of merging aerial images and ground laser scans. In *Proceedings of the 2005 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 222–225, Seoul, Korea, July 25–29 2005.

Publikationer *i serien* Örebro Studies in Technology

1. Bergsten, Pontus (2001) *Observers and Controllers for Takagi – Sugeno Fuzzy Systems*. Doctoral Dissertation.
2. Iliev, Boyko (2002) *Minimum-time Sliding Mode Control of Robot Manipulators*. Licentiate Thesis.
3. Spännar, Jan (2002) *Grey box modelling for temperature estimation*. Licentiate Thesis.
4. Persson, Martin (2002) *A simulation environment for visual servoing*. Licentiate Thesis.
5. Boustedt, Katarina (2002) *Flip Chip for High Volume and Low Cost – Materials and Production Technology*. Licentiate Thesis.
6. Biel, Lena (2002) *Modeling of Perceptual Systems – A Sensor Fusion Model with Active Perception*. Licentiate Thesis.
7. Otterskog, Magnus (2002) *Produktionstest av mobiltelefonantenner i mod-växlande kammare*. Licentiate Thesis.
8. Tolt, Gustav (2003) *Fuzzy-Similarity-Based Low-level Image Processing*. Licentiate Thesis.
9. Loutfi, Amy (2003) *Communicating Perceptions: Grounding Symbols to Artificial Olfactory Signals*. Licentiate Thesis.
10. Iliev, Boyko (2004) *Minimum-time Sliding Mode Control of Robot Manipulators*. Doctoral Dissertation.
11. Pettersson, Ola (2004) *Model-Free Execution Monitoring in Behavior-Based Mobile Robotics*. Doctoral Dissertation.
12. Överstam, Henrik (2004) *The Interdependence of Plastic Behaviour and Final Properties of Steel Wire, Analysed by the Finite Element Metod*. Doctoral Dissertation.
13. Jennergren, Lars (2004) *Flexible Assembly of Ready-to-eat Meals*. Licentiate Thesis.
14. Jun, Li (2004) *Towards Online Learning of Reactive Behaviors in Mobile Robotics*. Licentiate Thesis.
15. Lindquist, Malin (2004) *Electronic Tongue for Water Quality Assessment*. Licentiate Thesis.
16. Wasik, Zbigniew (2005) *A Behavior-Based Control System for Mobile Manipulation*. Doctoral Dissertation.

17. Berntsson, Tomas (2005) *Replacement of Lead Baths with Environment Friendly Alternative Heat Treatment Processes in Steel Wire Production*. Licentiate Thesis.

18. Tolt, Gustav (2005) *Fuzzy Similarity-based Image Processing*. Doctoral Dissertation.

19. Munkevik, Per (2005) *"Artificial sensory evaluation – appearance-based analysis of ready meals"*. Licentiate Thesis.

20. Buschka, Pär (2005) *An Investigation of Hybrid Maps for Mobile Robots*. Doctoral Dissertation.

21. Loutfi, Amy (2006) *Odour Recognition using Electronic Noses in Robotic and Intelligent Systems*. Doctoral Dissertation.

22. Gillström, Peter (2006) *Alternatives to Pickling; Preparation of Carbon and Low Alloyed Steel Wire Rod*. Doctoral Dissertation.

23. Li, Jun (2006) *Learning Reactive Behaviors with Constructive Neural Networks in Mobile Robotics*. Doctoral Dissertation.

24. Otterskog, Magnus (2006) *Propagation Environment Modeling Using Scattered Field Chamber*. Doctoral Dissertation.

25. Lindquist, Malin (2007) *Electronic Tongue for Water Quality Assessment*. Doctoral Dissertation.

26. Cielniak, Grzegorz (2007) *People Tracking by Mobile Robots using Thermal and Colour Vision*. Doctoral Dissertation.

27. Boustedt, Katarina (2007) *Flip Chip for High Frequency Applications – Materials Aspects*. Doctoral Dissertation.

28. Soron, Mikael (2007) *Robot System for Flexible 3D Friction Stir Welding*. Doctoral Dissertation.

29. Larsson, Sören (2008) *An industrial robot as carrier of a laser profile scanner. – Motion control, data capturing and path planning*. Doctoral Dissertation.

30. Persson, Martin (2008) *Semantic Mapping Using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle*. Doctoral Dissertation.