Faculty of Science and Technology
University of Coimbra

Automated Landmine Detection by means of a Mobile Robot

Svetlana Larionova

2007

Automated Landmine Detection by means of a Mobile Robot

Dissertation submitted to University of Coimbra

by

Svetlana Larionova

as a partial requirement for obtaining the degree

Doctor in Computer Engineering

Department of Electrical and Computer Engineering

Faculty of Science and Technology

University of Coimbra

2007

Thesis performed under the supervision of:

Professor Doctor Aníbal Traça de Almeida

Full Professor of the

Faculty of Science and Technology, University of Coimbra

and

Professor Doctor Urbano Carreira Nunes

Associated Professor of the

Faculty of Science and Technology, University of Coimbra

Abstract

Millions of antipersonnel landmines are left in the ground after past war conflicts across many countries. Being functional for more than 50 years they provide a lot of humanitarian and economical problems long after the conflict is finished. Cleaning the existing minefields, called humanitarian demining, is required in order to return the large areas of the land to normal use and save the local civilians from the danger. Currently, the only fully trustable solution for this problem is the manual clearance which is itself a very dangerous and slow procedure. Automation of the humanitarian demining may change the situation providing a faster approach which eliminates the participation of humans on the minefield. This work is a part of the effort toward the development of such approach.

Automation of the humanitarian demining meets a lot of technical problems which currently do not have effective solutions. This work covers the ones related to the automatic detection of antipersonnel landmines assisted by a mobile scanning platform which carries the landmine detection sensors.

The landmine detection approach developed in this work assumes the employing of several nonselective sensors most widely used for landmine detection which include metal detectors, infrared sensors, and ground penetrating radar. The approach has a multi-stage structure and is based on feature-level sensor fusion strategy. This process is understood as a step-by-step reduction of the false alarm rate depending on the quality of the available sensor data. During the first stage the sensor data are processed in order to distinguish all objects suspected to be landmines against the background. For this purpose a novel online algorithm was developed. It allows to detect the object during the robot movement and is hardly sensitive to the quality of the sensor data. The consequent stages are performed in order to recognize the landmines among the detected suspicious objects. A number of new classification features were developed in order to perform this recognition. Based on the feature analysis a concept of *selective training* specially suited for the landmine recognition task was developed. This technique allows to account for the high overlapping of the classes and multimodal distributions of the classification features. Finally, a concept of *dominant class* was introduced in order to provide high levels of detection rates even in case of poorly separated classes. Being specially designed for the specifics of landmine detection the proposed algorithms allow

to improve the results.

In order to assist the gathering of the sensor data, the problems related to the effective sensor data gathering,path planning and localization of the platform are also addressed. The developed solutions are implemented on the previously created pneumatic scanning platform acting as a prototype demining robot. A number of practical solutions improving the platform localization were developed. The positioning of the robot is based on its odometry, compass and a novel vision system which are combined together by means of a Kalman filter. The vision system employees a simple CCD camera and is guided by a novel algorithm for the detection and association of natural landmarks found on the ground surface.

Finally, considering the landmine detection problem in the scale of minefield the problem of the field exploration is addressed. Assuming a general case in which the minefield may be populated with some obstacles in unknown positions an algorithm for online unknown area coverage was developed. The algorithm guarantees regularity of the robot path necessary for the mapping of sensor data and the safety of the robot by planning its path only inside already covered area.

The developed algorithms were implemented in a form of control software for the real platform. Testing of the proposed ideas in simulation and in real conditions (on a test minefield) provided promising results showing the perspective of the developed concepts. Based on the experimental results the recommendations for future work are formulated.

The automatic landmine detection task raises a number of challenging problems which have connections to other areas of robotics, pattern recognition and control. In this regards the development of the methods proposed in this work was considered in a more broad sense. Thus, the results of this work can be used in the adjacent fields of robotics: automatic subsurface sensing with online reaction to the found target, pattern recognition in case of poorly distinguished classes, and online unknown area coverage required for cleaning, grass cutting, agriculture, etc.

Acknowledgements

This work would not be completed without help, support and encouragement of several persons.

I would like to thank my supervisors Aníbal Traça de Almeida, Urbano Carreira Nunes and Lino José Forte Marques for a very interesting research topic, helpful ideas, discussions and critics.

I am very grateful to Nuno Luís Lopes de Almeida for his support in all kind of problems faced by me in Portugal, for useful advises, discussions and support concerning my work. Many thanks also to other colleagues of Institute of Systems and Robotics (ISR) and specially the Embedded Systems Laboratory, ISR, Coimbra for technical and emotional support. Specially, to Machmood Tavakoli for objective discussions about mechanics related issues, and to Rui Manuel Silva Costa for willing help.

I will also give special thanks to Dr. Mikhail Zheludkevich, University of Aveiro, Portugal for inspiring and encouraging, and also for the close look on the thesis, a lot of advises on its structure and careful reading of preliminary and final versions.

The landmine detection experiments performed in this work were carried out on the test fields located in Meerdaal Bomb Disposal Unit, Belgium. I would like to say many thanks to the staff of the unit for hospitality and well-timed help, specially to Marnik Jacobs. Many thanks to Daniela Doroftei, Marc Acheroy and Prof. Yvan Baudoin from Royal Military Academy, Brussels, Belgium for the help in organization of the experiments.

I am also very grateful to my friends from Saint-Petersburg in Russia, Coimbra and Aveiro in Portugal for emotional support and relaxing atmosphere. Many thanks to my family for patience and belief in my success.

Financial support has been provided by Fundação para a Ciência e a Tecnologia, Portugal with the research grant SFRH/BD/9164/2002 and under project DEMINE. I thank also the Institute of Systems and Robotics, Coimbra, Portugal where my work was performed.

Contents

List of Acronyms

APL antipersonnel landmine

MD metal detector

EM electromagnetic

PMD pulsed metal detector

CMD continuous metal detector

IR infrared

GPR ground penetrating radar

ROI Regions-Of-Interest

ICBL International Campaign to Ban Landmines

GICHD Geneva International Center for Humanitarian Demining

RPC Radio Packet Controller

MSMS Multi-Sensor Mine Signature

JRC Joint Research Center

SM Segmented Map

OA Object Area

DM Data Map

DR detection rate

FA false alarm

FAR false alarm rate

PCA principal component analysis

ROC Receiver-Operating Characteristic

SLAM Simultaneous Localization and Mapping

FSR Force Sensing Resistor

LADERO LAndmine DEtection RObot

ATMID All Terrain Mine Detector

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides an introduction to the problem of humanitarian demining and possible solutions for it. The landmines problem in described in Section 1.1, and Section 1.2 is dedicated particularly to the specifics of humanitarian demining. The solutions currently used for humanitarian demining and the ones being in development are reviewed in Section 1.3. A special attention is paid to the sensor technologies and to the algorithms available for landmine detection in Section 1.4 because this is one of the main topics of this work. Finally, the goals of this research are formulated in Section 1.5 followed by an outline of the main contributions (Section 1.6) and an overview of the thesis (Section 1.7).

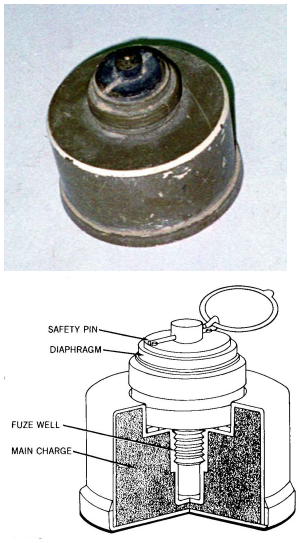
1.1 The Landmines Problem

Only two things are infinite, the
universe and human stupidity,
and I'm not sure about the former.

Albert Einstein (1879 - 1955)

Landmines are used as a tactical weapon in time of war. This is a self-contained explosive device which is placed onto or into the ground, designed to explode when triggered by a vehicle, a person or an animal. Landmines are usually used to restrict enemy movements and to secure disputed borders. Unfortunately, the effect of landmines last much longer than the conflict, affecting not only the army forces but mostly the civilian population after the conflict ending.

Among all types of landmines antipersonnel landmines produce the largest problem because they are specially designed to kill or maim people. An antipersonnel landmine (APL) can be a very simple device which costs on average only 3$. Normally, its size is not larger than 15 cm and it triggers from a pressure of several kilograms (see,



(a) Belgian antipersonnel landmine M35BG

(b) A child on a minefield in Cambodia (image produced by The Lutheran World Federation, `http://www.lutheranworld.org`)

(c) Falklands, Argentine

Figure 1.1:

for example, Figure 1.1(a)). To achieve their purpose during war-time the antipersonnel landmines are hidden in the ground and the minefields are not marked (specially the minefields created by terrorists). The problem is that the same situation remains after the conflict and the minefields are abandoned and left to the competence of local authorities. Most landmines can be functional for 50 years and some last much longer. The first things which can be done in this case are marking the minefield and educating the local civilians not to enter it. However, this does not completely stop the accidents when people return to their country and discover that their houses are now on a minefield, as happens for example, in Cambodia, Figure 1.1(b). Often a minefield is discovered only after some casualties have happened on it. There is a large economical impact of landmines on the civilian life; large areas cannot be used for agriculture or other purposes. Obviously, the effectiveness of antipersonnel landmines during the war cannot excuse the danger they provide to civilians for many years after.

As it often happens in human history, the danger of landmines was only recognized long time after the millions of them were laid in the ground. Since that time large effort was made to change the situation. Very important results were achieved by International Campaign to Ban Landmines (ICBL) `http://www.icbl.org/` which was awarded for this with the Nobel peace prize in 1997. ICBL provided the 1997 Convention on the *Prohibition of the Use, Stockpiling, Production and Transfer of Anti-Personnel Mines and on Their Destruction* [6] which is currently signed by 154 countries. This document

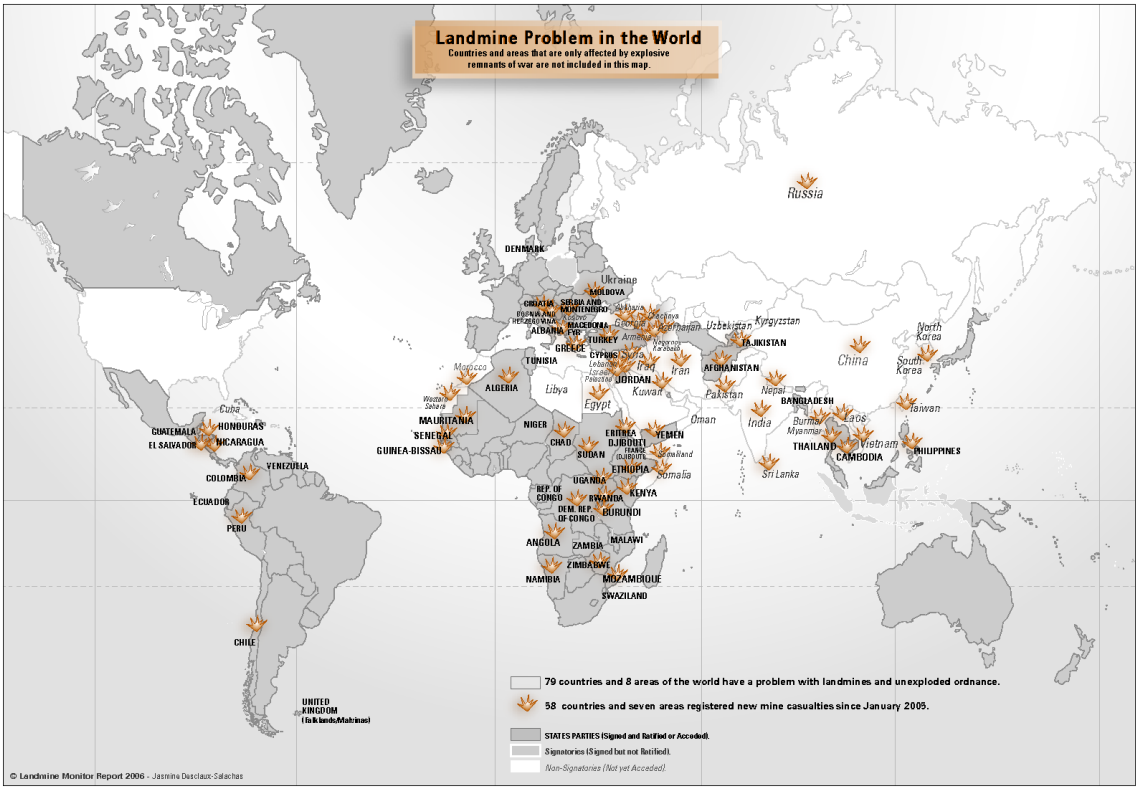


Figure 1.2: Global landmine problem [1]

significantly changed the politics of the member States according to use and production of landmines but it is not signed yet by two large countries which still produce and use antipersonnel landmines: USA and Russia. ICBL also maintains a comprehensive report about the world landmine problem. This document, called *Landmine Monitor*, is published every year and presents to the public an update of the situation in the countries related to the landmine problem (information on landmine use, production, trade, stockpiling, demining, casualties and survivor assistance in 126 countries and areas). According to the Landmine Monitor 2006 [1] currently:

- At least 78 nations are affected to some degree by landmines;
- Antipersonnel landmines are still produced by 13 countries and are used by 10 countries (including official use by 3 governments);
- There are over 160 million antipersonnel mines stockpiled;
- There are 15,000-20,000 new casualties from landmines and explosive remnants of war each year (recorded in 58 countries during 2005-2006), there are approximately 350,000 to 400,000 mine survivors in the world today which need long-term care;

Figure 1.2 shows the global situation about landmine distribution and accidents.

Along with maiming civilian people minefields affect the environment as well. A landmine left in the ground leaks heavy metals and explosives which can poison water and soil. However, there is also an opposite effect when a minefield protects animals from people providing more comfortable conditions for them. For example, penguins on Falkland Islands freely mate on a minefield Figure 1.1(c). This situation is probably looking more positive for ecologists but there are still no doubts that the minefields should be cleared and the land returned to the normal use.

Important activities toward the solution of the global landmine problem include *mine risk education* and *survivor assistance* which are needed while minefields exist. The clearance of a minefield is a very dangerous and complex procedure called *humanitarian demining*.

1.2 Humanitarian Demining

Following the ICBL there are many activities around the world related to *humanitarian demining*. They are mainly coordinated by the standards of Geneva International Center for Humanitarian Demining (GICHD) `http://www.gichd.ch/`. According to the definition of GICHD [7]:

> Humanitarian demining includes activities which lead to the removal of mine and UXO (Unexploded Ordnance) hazards, including technical survey, mapping, clearance, marking, post-clearance documentation and handover of cleared land.

The process of *humanitarian demining* consists in several steps:

1. *Assessment*: determining if a mine action is necessary and feasible;
2. *Mine clearance surveys*: technical and social survey of the area to be cleared, determining the depth of clearance, local soil conditions, vegetation and marking the area;
3. *Mine clearance techniques*
4. *Post-clearance inspection*: reports, marking;
5. *Community notification*

In contrast to *military demining* the *humanitarian demining* should lead to a complete clearance of the specified area, so it can be safely used for civilian purposes. The current standard of the clearance maintained by GICHD recommends considering the area to be clean if all mines and UXO hazards are removed and destroyed [8]. By other words, a clearance rate of 100% is required. This situation is further complicated by the following:

(a) Sudan (b) Afghanistan (c) Lebanon

Figure 1.3: Sample minefields (images produced by UNMAS, `http://www.mineaction.org`)

Figure 1.4: Sample antipersonnel landmines

- Minefields are located in very different environmental conditions (see, for example, Figure 1.3), often in hardly accessible areas;
- There are a lot of types of landmines worldwide. They differ in size, shape, used materials (see, for example, Figure 1.4) which complicates the development of standard detection methods.
- The landmine problem affects a lot those countries where the cost is often the main concern. This limits the use of new and developing technologies.

Human inventiveness produced hundreds of different antipersonnel landmines (see, for example, the ORDATA database [9]). However, it seems that currently the only effective manner to remove and destroy a landmine is a manual technique. The basic technique used during manual demining is probing the soil in front of the deminer centimeter by centimeter. The landmines can be discovered as any other rigid object by the force which the prober returns to the deminer's hand (Figure 1.5(b)). This is a very dangerous and slow procedure which requires good personnel skills of the deminer. In most cases the process can be made more efficiently by using a handheld contactless mine detector which is swept over the ground. Landmines usually contain some amount of metal, thus a widely used tool for manual demining is a metal detector (Figure 1.5(a)). However, only few types of landmine have metallic cases, and others may include only some grams of metal (a fuse) which makes them undistinguishable from metal debris left in the ground after the conflict. In this situation the data from a metal detector is ambiguous providing a lot of false alarms and returning the situation to the case of

(a) Probing the area with a metal detector (image produced by Luke Powell, `http://www.un.org.pk`)

(b) Manual drilling and probing the soil (image produced by MACC-SL, `http://www.mineaction.org`)

Figure 1.5: Manual technique of mine clearance

careful probing.

The manual procedure is the basic tool of humanitarian demining nowadays. Even so, other more advanced techniques exist; they are seldom used because their cost is not affordable in most situations. The most common helpers of the deminers are dogs which have great olfaction senses allowing them to detect the explosives leaking from the landmines. However, a specially trained demining dog is still too expensive. From the above information it can be seen that the technology of humanitarian demining needs to be advanced in terms of speed and safety to make the complete clearance of the existing minefields an affordable task. It was estimated that one deminer is killed for every 5000 landmines removed [10].

1.3 Automation of Humanitarian Demining

The problems related to the manual humanitarian demining have provoked many attempts to automate this process. There are two main objectives for this effort:

- Decreasing the time of mine clearance
- Eliminating the human participation on the minefield

Following the pointed goals an automated system must still provide the same quality of mine clearance archived by the manual technique. Automation of humanitarian demining should be understood here in a broad sense meaning any tool which allows to diminish or completely eliminate participation of humans on the minefield. It is important to mention that there is no automated demining system which can provide the required clearance rate yet. The research in this area is ongoing in several directions which are presented below.


Figure 1.6: Sample machines for mechanical mine clearance

1.3.1 Mechanical mine clearance

A naive technique for mine clearance consists in a safe activating of the landmines situated on a minefield. This is particularly useful in the case of antipersonnel landmines because their destroying force is not sufficient to damage a relatively simple vehicle. The field of mechanical mine clearance is the most developed one and there are commercially available vehicles (Figure 1.6). There are a number of different ideas used for the mechanical design:

- Mine Clearing Flails: this machine hits and churns the soil with long chains to detonate and break apart mines.
- Earth Tillers: these machines till the ground to a pre-set depth, detonating or breaking up mines.
- Wheel Shovel: a shovel fitted with a mesh basket. When the rubble is shaken out, landmines and large ordnance remain.
- AP Mine Sifter: the contaminated soil is picked up and sifted and the remaining debris can be inspected.
- Mine Protected Vehicle: vehicles designed to detonate mines and resist mine explosions.

However, these machines cannot guarantee a complete clearance, thus they have to be used in combination with manual techniques. Moreover, the cost is still a limitation for their wide use.

Along with using of brutal force to deal with a minefield there is some development in the direction of mechanical systems which include some intelligence. For example, in the framework of project ELADIN [11] it is proposed to use high pressure water for the purpose of landmine detection and deactivation [12].

Figure 1.7: A teleoperated robot and its operator panel at Meerdaal Bomb Disposal Unit

1.3.2 Teleoperated vehicles

One step toward an automatic system is the teleoperated one which allows substituting the human by a robot in the dangerous conditions. An example of such system can be seen in Figure 1.7. A teleoperated vehicle is controlled by the operator from a safe distance using video cameras. Changeable manipulators are used to perform the necessary operations which include, for example, target deactivation by high pressure air. Such machines are available in the market and are widely used by bomb disposal units to assist in their missions. They are mostly suitable for short operations when the location of the dangerous object is more or less known. However, teleoperated robots for the humanitarian demining in a large scale are still in the state of development.

Several new designs of semi-autonomous robots are proposed in [2]. These vehicles can move to a specified location and then scan a small area using an advanced manipulator (Figure 1.8).

The advantage of the teleoperated concept is the ability to develop a relatively simple vehicle and give the human operator with the complex task of decision making. Then, the robot itself can be made very reliable which is important because the reliability is one of the main concerns on a minefield.

(a) Mine Hunter (b) Gryphon

Figure 1.8: Semiautonomous vehicles for landmine detection [2]

1.3.3 Autonomous vehicles

The largest degree of automation for humanitarian demining can be implemented in a fully automated system which should be able to perform all of its stages. However, this is a very complex and probably unaffordable task. So, currently the research is mostly concentrated in the direction of automated mine clearance because it is the most dangerous stage of humanitarian demining. An automated vehicle has to perform The following main tasks in order to assist the mine clearance task:

- Landmine detection
- Removal or marking (for further removal by humans) of the detected landmines
- Planning a safe path through the minefield to ensure its full exploration
- Avoiding eventual obstacles on the way

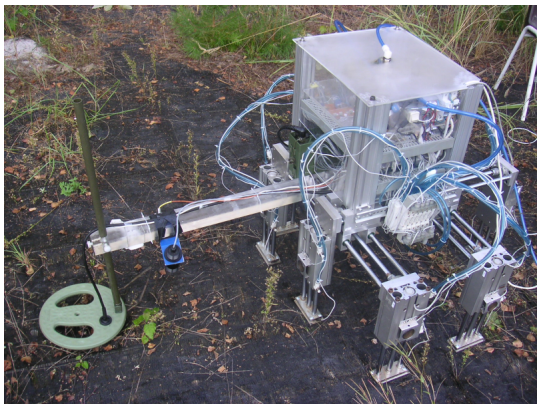
In the area of automated mine clearance there are several platforms being developed. Most of the research projects are dedicated to the mechanical design of the platform to make it reliable in the harsh outside conditions. Few projects also incorporate the landmine detection and the path planning tasks. There are also few developments in the area of automated landmine removal.

A quadruped walking mechanical structure for the demining robot is proposed in [16, 13, 17]. The implemented TITAN-IX robot shown in Figure 1.9(a) is able to change working tools in the end of one leg allowing it to operate as a landmine detector (specifically, a metal detector). The working principle of the robot consists in scanning of a small area while staying stable and then moving to the next location.

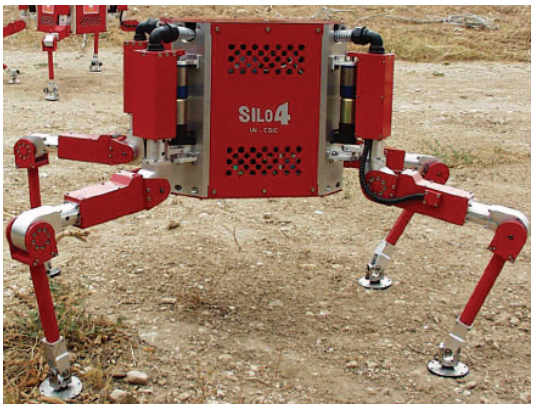
Another spider-like robot SILO4 is proposed to be used for humanitarian demining in [15]. The robot has a good adaptability for outdoor terrain Figure 1.9(d). The developments also include an adaptable manipulator for a landmine detector.

(a) TITAN-IX [13]

(b) Finder [14]



(c) LADERO



(d) SILO4 [15]

Figure 1.9: Sample mobile demining robots

Figure 1.9(b) shows a simple wheeled platform for testing of demining path planning algorithms [14]. This project is concentrated mostly on the path planning and navigation algorithms. The proposed unknown coverage algorithm is based on the Morse decomposition of the area [18, 19]. The main disadvantage of this algorithm is its high dependence on the precise localization of the robot which requires improvements in the mobile platform localization sensors. There are also developments of probabilistic path planning algorithms specially designed for fast landmine search [20, 21] appropriate for the military but not for the humanitarian demining.

A pneumatic cartesian platform is proposed for landmine detection in [22, 4] (see Figure 1.9(c)). The platform has a simple structure and is scanning the area during movement. Several sensors can be installed in front of the robot to perform the landmine detection. The last platform is used in the presented work for the development of the algorithms for landmine detection and path planning.

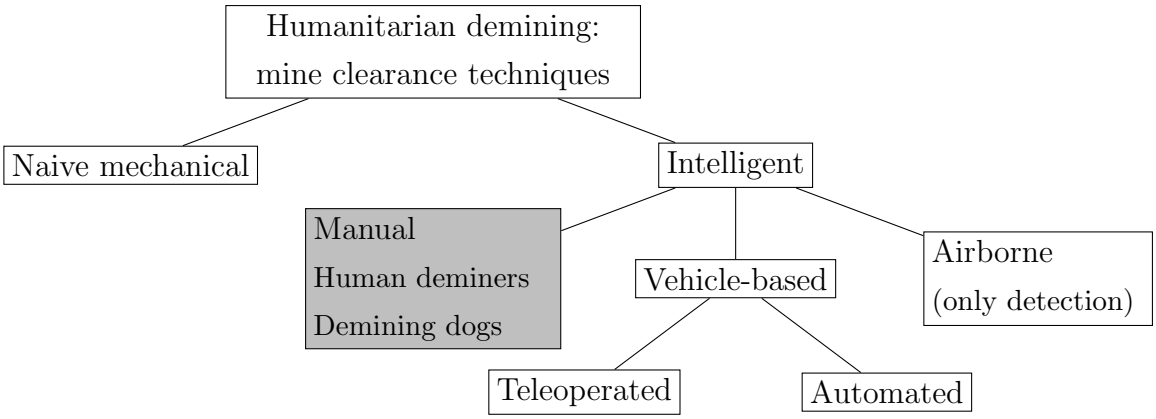


Figure 1.10: Approaches for clearance techniques which can be used for humanitarian demining. The only widely accepted technology is the manual demining (highlighted with gray)

1.3.4 Summary

The automated techniques described in this section can be classified as shown in Figure 1.10. Currently, the most widely used technologies are mechanical mine clearance and the teleoperated vehicles. There is also some work done in the area for airborn assistance of landmine detection and minefield reduction [23, 24]. However, the domain of automated mobile platforms for landmine detection (and, eventually, removal) is developing and can be considered the most promising for future applications.

1.4 Sensors and Algorithms for Landmine Detection

This section examines the existing techniques for landmine detection from the point of view of an automatic demining system. Some of these approaches are widely used for the manual humanitarian demining as well (metal detectors), but most of them are currently in the stage of research and cannot be yet accepted for real operation.

If a landmine was recently placed into the ground, the place where it was buried in can usually be well seen by an experienced deminer. An automatic system can use vision sensors and pattern recognition techniques to perform the same operation and discover the landmines [25]. However, in most cases landmines rest in the ground for many years after which there may be no visible characteristics available (usually several levels of soil and vegetation cover the minefield). Thus, in general it is logical to consider that the landmine sensors should not rely on the visual information. Such sensors are related to the domains of subsurface sensing and nondestructive control.

An antipersonnel landmine is a complex object composed from several heterogeneous

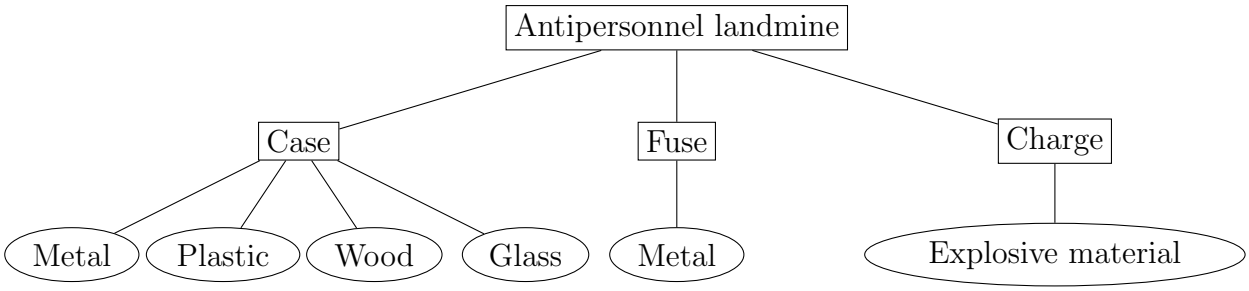


Figure 1.11: Characteristics of an antipersonnel landmine which could be sensed

parts: explosive charge, fuse system and case (see Figure 1.11). These parts have physically different properties, thus there is no general "landmine sensor" which would allow to sense a landmine as a whole object. However, the parts of the landmine can be sensed separately, for example, as shown in Figure 1.11. Each of the characteristics provides a disturbance of some physical property in relation to the surrounding soil. Any of these characteristics can be used for the detection of the whole landmine if it is unique for landmines in comparison to other objects located on the minefield. It can be seen later in this section that, in general, all the characteristics shown in Figure 1.11 except the *Explosive material* are not selective. From the other side, sensors which are suitable for sensing of the explosive materials do not have high selectivity themselves. In this situation, it is usually not possible to provide the required quality of landmine detection by sensing only one of the landmine characteristics. Thus, there are developments of sensing devices using different physical principles (appropriate to the landmine characteristics) [26] in order to provide as more information as possible. A combination of the characteristic and the sensing method defines a type of *signature* which a specific detector acquires from the landmine. *Landmine signature* is a general term which can be used both in the case of manual and automated landmine detection: it is a set of signals provided by the detector together with their relation to some spatial information. It should be noticed that the degree of nonselectivity of a sensing method also depends on the approach used for the discrimination between landmines and other objects. For example, the amplitude of the signal may not be a selective property, but the shape of its spatial signature may provide enough information for the discrimination.

The quality of the sensing method is affected both by the selectivity of the sensor and by the selectivity of the sensed property. It must be also mentioned that the final result of landmine detection is a binary decision taken by some algorithm using the sensor data. Thus, the parameters of the data processing algorithms also affect the detection quality. The whole process of landmine detection is usually characterized statistically using the notations of the Signal Detection Theory [1]. As applied to the case of landmine

[1]Signal Detection Theory was introduced in 1966 by John A. Swets and David M. Green. It is widely accepted for the analysis of signal detection algorithms in presence of noise

detection the following terms can be defined:

- N_M - number of landmines (or, more generally, dangerous objects) to be examined, a number of landmines in a specified area S
- $N_{\overline{M}}$ - number of nonhazardous objects in S
- *False alarm* - a decision that a landmine is present in its absence
- *Probability of detection* or *detection rate* is the ratio of the number of correctly identified landmines to the total number of existing landmines

$$P_d = \frac{N_{M+}}{N_M}, \tag{1.1}$$

  where N_{M+} is the number of correctly identified landmines
- *Probability of false alarm* or *false alarm rate* is the ratio of the number of false alarms to the number of nonhazardous object

$$P_{fa} = \frac{N_{\overline{M}+}}{N_{\overline{M}}}, \tag{1.2}$$

  where $N_{\overline{M}+}$ is the number of false alarms. This measure is convenient for estimating the quality of the detection algorithms. However, it is not very meaningful for representing the quality of the landmine detection in overall because every false alarm causes a costly operation of the landmine removal. Thus, not only the number but also the area of false alarms is important. A widely used measure in this case is the *number of false alarms per square meter*

$$N_{fa/m^2} = \frac{N_{\overline{M}+}}{S}, \tag{1.3}$$

  where every *false alarm* is allowed to cover an area not larger than a specified size (otherwise it is considered as multiple false alarms).

1.4.1 Metal detection

Among all sensing technologies for landmine detection the detection of metal has the oldest history and is the most developed. The contactless detection of metal is a relatively simple task because all metals are conductive and thus they disturb electromagnetic (EM) waves in some manner [2]. There are two approaches which can be used: *active* and *passive*. Active sensors produce a EM field in the direction of the investigated object and measure the disturbance of the field the object provided. These are the most widely

[2]The first metal detector was invented by Gerhard Fischer in 1930's after he noticed that his radio-based navigation system worked abnormally in the areas which contain ore-bearing rocks

(a) Shiebel ATMID Metal Detector (b) FEREX Magnetometer

Figure 1.12: Sample metal detector and magnetometer

used detectors, which are in fact called *metal detectors*. The passive metal detection consists in measuring of the Earth's magnetic field which is disturbed by a magnetic metal object. This type of sensors, called *magnetometers* (see Figure 1.12(b), can be used only for detection of relatively large metal objects and thus have limited usage in humanitarian demining.

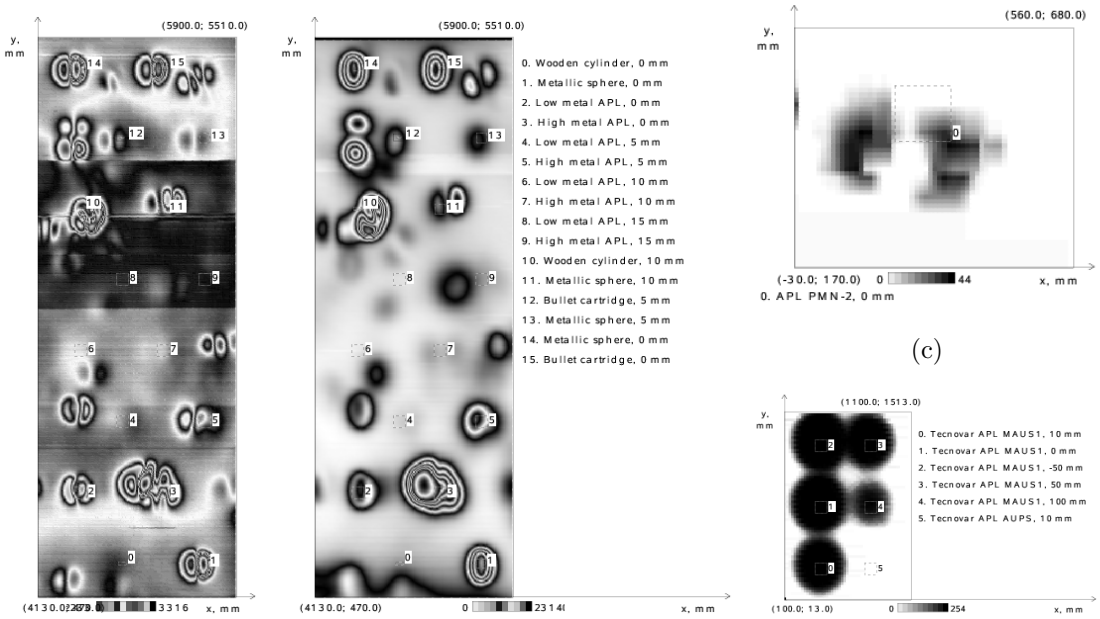
Currently, metal detectors are widely used in manual humanitarian demining to assist the landmine detection [3]. This technology was the only available for a long period when the landmines normally included large amounts of metal (for example, a metal case). Modern landmines may contain only few grams of metal (only the fuse) which significantly complicates the detection: the signal provided by a landmine has the same strength as the signal provided by the metal debris present in the ground or by the soil itself (providing high false alarm rate). This limitation forces the development of other sensing technologies. However, a standard metal detector (MD) is a trusted tool for manual demining as it is assumed that the landmines contain at least some amount of metal [4]. An example of a metal detector for manual demining is shown in Figure 1.12(a). It usually contains a search head, control electronics and headphones which provide to the deminer an audio signal proportional to the measured amount of metal.

Conventional metal detectors employ the principle of electromagnetic induction:

1. An alternating current is applied to the active element of the metal detector (called, *search coil* or *search head*);

[3]Other areas where metal detectors are used include treasure hunting, investigation of pipes, cables etc. in civil engineering, and other related fields

[4]The manufacture of completely nonmetallic antipersonnel landmines is banned by the Protocol II of the United Nations Convention on Conventional Weapons (CCW) of 3 May 1996



(a) (b) (c) (d)

Figure 1.13: Sample data obtained from test minefields by different metal detectors: *a* - pulsed metal detector (Vallon GmbH ML 1620C), *b* - two frequency continuous wave metal detector with phase-sensitive demodulation (Foerster Minex 2FD 4.500), *c* - continuous metal detector (Shiebel ATMID), the shape of the search head influenced the output signature, *d* - pulsed metal detector (EBEX 420 PB Ebinger) using the audio frequency as the output signal (the maximum is cut off)

2. The search coil current produces an alternating EM field around the coil (primary EM field);

3. The primary EM field induces (alternating) eddy currents in the metal object and, in case of a magnetic metal, magnetizes it;

4. The magnetized object and the eddy currents also produce a EM field (secondary EM field);

5. The secondary EM field is related to the amount and type of metal in the investigated object. It is measured by the detector and compared with the primary EM field;

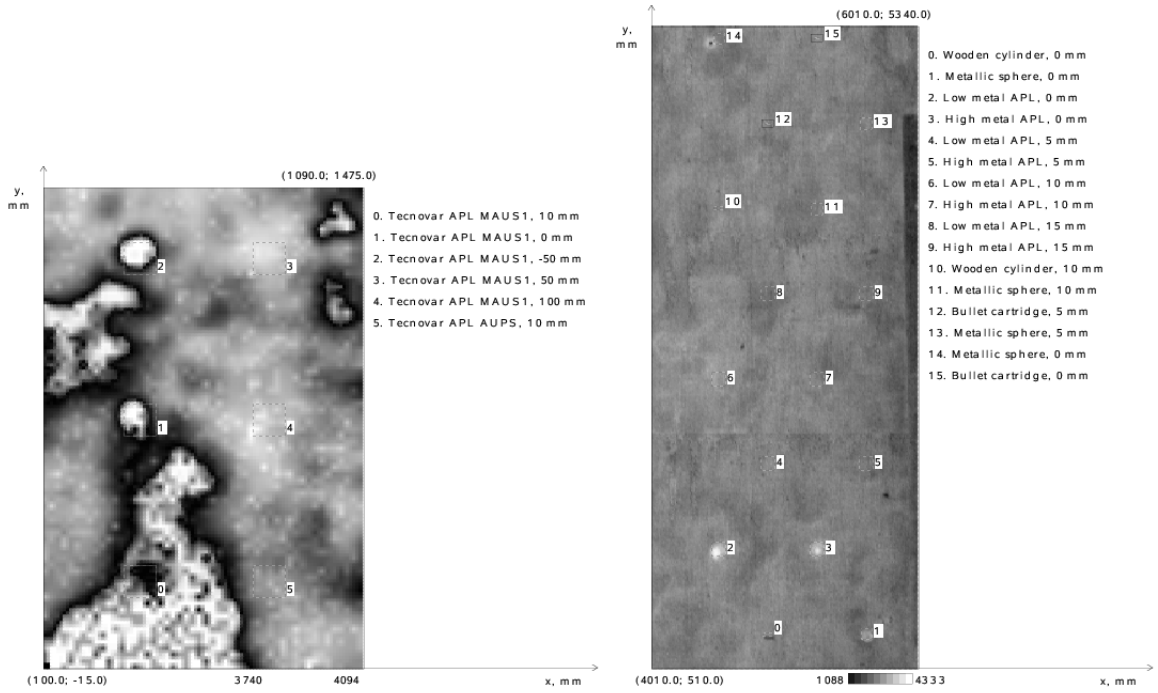
In the modern metal detectors this basic principle is implemented in many different ways. The working conditions and the types of targets to be detected determine the most suitable implementation. A comprehensive overview of metal detectors technology and their use for humanitarian demining can be found in [27].

There are two basic types of metal detectors which differ by the type of EM field they utilize: *pulsed* and *continuous*. The pulsed metal detector (PMD) produces a short

pulse of electromagnetic field and measures its decay time. The presence of a metal object slows down the decay of the pulse. The continuous metal detector (CMD) uses a continuous sinusoidal signal to produce the primary EM field. Then the amplitude and phase of the secondary EM field provide information about the presence of the metal object. The type of the EM field used by the MD is essential for the interpretation of its output signal because it determines the design of control electronics, control principles, the approach for ground compensation, etc. The shape and size of the search coil also influences the output signal pattern of the detector. Moreover, when using a MD in an automated landmine detection system the signal acquisition principle is also important. Metal detectors are usually designed to provide an audio signal of changing frequency which is the most representative form of the output for a human deminer and can be measured by the automated system. An automated system may also utilize a raw output which is not transformed to the conventional audio signal. Without taking into account the operational principles of the used MD and the acquisition principle the data interpretation may provide wrong results. Figure 1.13 illustrates this showing the data obtained from different metal detectors. It can be seen that the signatures of the metal objects are significantly different for different types of metal detectors. For example, the MD used to obtain the data shown in Figure 1.13(b) provides a bipolar signal where the position of the metal object is defined by a zero signal, while the signals from most of the others MDs have maximums at the locations of the objects. In Figure 1.13(c) the shape of the signature is related to the shape of the search coil but not to the shape of the metal object (the MD coil has the shape shown in Figure 1.12(a)). In this case the interpretation of data must incorporate the specifics of the detector. The signatures presented in Figure 1.13(d) are less variable because the used output audio signal does not provide a large variation (it is designed to provide to the deminer an almost discrete signal).

The most intuitive approach for automated landmine detection using a metal detector consists in analyzing the signal amplitude which is proportional to the amount of metal. If it can be assumed that the landmines contain an amount of metal different than the surrounding soil and other objects, then a simple signal thresholding can be used for detection. However, the amount of metal is not a selective property of a landmine. The signatures provided by metal detectors can be very similar for landmines and other artificial and natural objects, like cans, coins, nails, mine and UXO fragments, metal-bore stones, etc. The soil itself may contain magnetic components which provide the same signal as a landmine. So, a simple thresholding is not efficient in most real word cases.

A more sophisticated approach consists in studying and using the physical model of the detector response for landmines and clutter. For example, in [28] it is proposed to utilize properties of two frequency metal detector to discriminate between mines and



(a) (b)

Figure 1.14: Example of passive radiometer data (a) and IR camera data (b)

metal debris. It is assumed that mine targets have certain sizes (non minimum metal mines) which is larger than the size of the clutter. It is shown that the size difference can be obtained by analyzing the phase of the received signal and the "phase threshold" can be used to reject clutter. However, in practice this approach is difficult to implement under non ideal conditions because it is influenced by electrical and magnetic properties and orientation of the object. It is proposed in [29] for a pulsed metal detector to use geometrical particularities of mine signature. Spatial signatures of energy of the measured output signal are considered and it is assumed that metallic mines have symmetrical spatial energy signature in contrast to the clutter. The symmetry is determined by subtracting the values of energy along two orthogonal axes. In [30] the output of the metal detector is modeled based on a weighted sum of decaying exponential functions. The parameters of the model are determined using training data obtained from a test field with buried landmines and clutter. For each type of object an exhaustive search over all possible values is performed to obtain the parameters which provide the best fit of the model to the measured signal. In that way, the approach proposed in [30] incorporates greater amount of statistical learning into the model which makes it more reliable in real conditions.

1.4.2 Detection of thermal radiation

Landmines, as all other objects with nonzero temperature, emit thermal radiation in form of electromagnetic waves of specific frequency. Measuring this radiation can allow to distinguish an object if its temperature is different from the background. The detectors of thermal radiation use for the measurements various frequency bands. The most widely used are *passive microwave radiometers* which work in microwave range (2-15 GHz) and *infrared cameras* for infrared range (1-15 μm). Sample signatures obtained from these sensors can be seen in Figure 1.14. The IR cameras are more convenient because they allow to perform measurements immediately from a large area of surface similar to the usual video cameras. Thus, the infrared cameras were widely investigated for the purpose of landmine detection (the only landmine detection technology which can be used for relatively large areas without scanning). However, there are a number of problems associated with the detection of thermal radiation in general:

- The obtained data are highly ambiguous because the number of clutter objects is very large (any natural or artificial objects, soil heterogeneities and voids may produce thermal signatures)

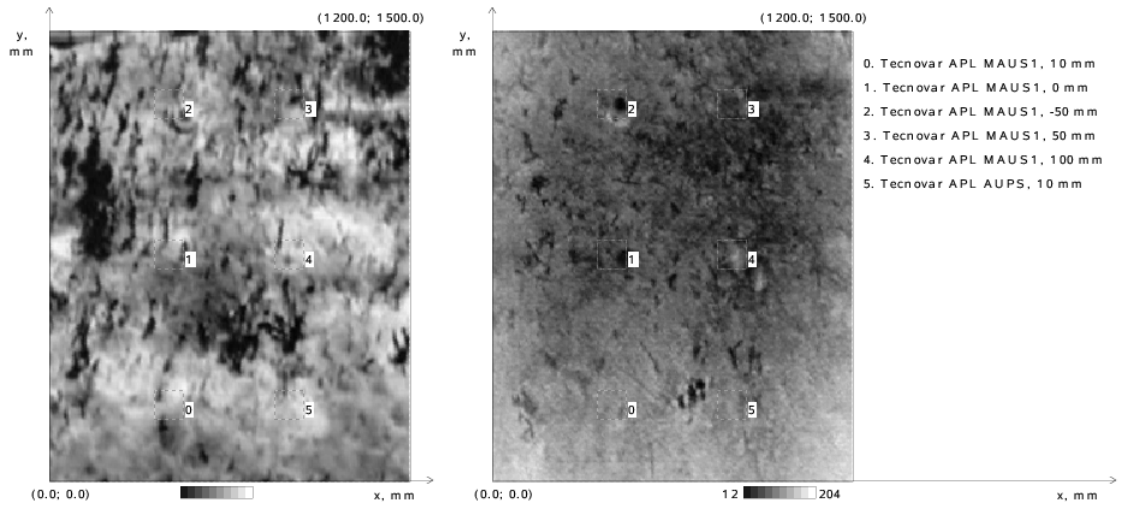
- Thermal radiation has low soil penetration allowing to detect objects not deeper than 5-10 cm

- Thermal signatures highly depend on weather conditions, time of day, season, etc.

An example of such complex data from IR camera can be seen in Figure 1.15: only the surface-laid objects are well seen, the images contain a lot of clutter and highly depend on the time of the day.

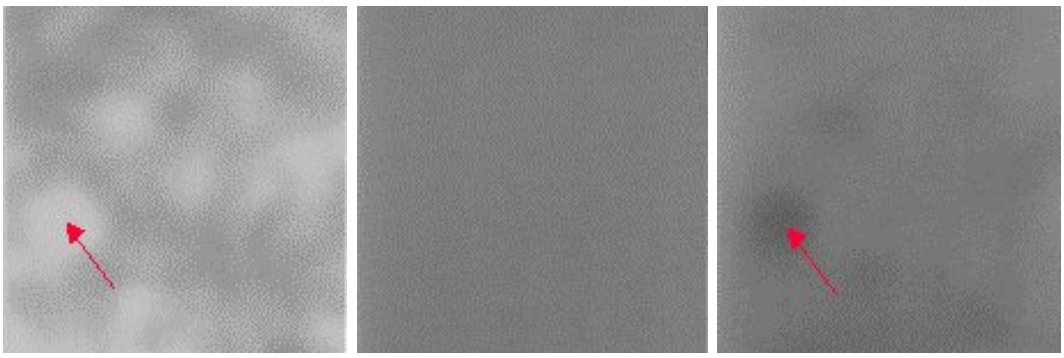
The situation is also complicated by the fact that the landmines buried in the ground for a long time are in temperature equilibrium with the surrounding soil. The temperature difference can only be noticed if the soil is heated or cooled: materials with different thermal conductivity achieve the equilibrium after different times, thus, there is a period of time when a buried landmine is in contrast (warmer or colder) with the background. Naturally the heating happens in the morning after the sunrise, and the cooling - in the evening after the sunset. There is also some time during the day and night when the soil and buried objects are in equilibrium (transition period) and the landmine detection is normally not possible. This process is illustrated in Figure 1.16. Taking into account the mentioned limitations the thermal radiation detectors usually should be coupled with other technologies.

The temperature of the object can be determined from the measured radiation. However, the temperature itself does not represent meaningful information for object recognition, since it depends on a large number of unknown parameters. The difference



(a) 15:25 (b) 20:20

Figure 1.15: Example of data from IR camera obtained at different time of the day



(a) Mid-day, positive contrast (b) 6-8 a.m., no contrast (c) 10 p.m., negative contrast

Figure 1.16: Signature of buried landmine during the day (the red arrow shows the location of the landmine), the experiment was done by HUDEM WorkGroup-2 (HUDEM IR trials at Meerdael, Belgium, April 1-4, 1998) using AGEMA IR camera (3μm-5μm)

between the temperatures of the object and the background is also an ambiguous parameter because it depends on the time of the day. In general, the theoretical modeling of the thermal radiation detectors is usually not used for landmine detection due to unaffordable complexity of real outside environment (an example of microwave radiometer response modeling for ideal conditions can be found in [31]). In this situation the shape of the signature, which usually represents the shape of the object (for example, the case of a landmine), is probably the only parameter which should be used for landmine detection.

Usually the image is analyzed using techniques of computer vision, particularly the approaches for circle detection. For example, Hough transform and Tophat filter are used in [32] and [33], mathematical morphology is used in [34]. These methods are well developed only for circular shapes leading to an important limitation. However,

in many locations most or even all landmines have circular cases making this approach applicable. However, standard methods for circle detection usually do not perform well enough because the signature does not have a sharp edge. A modified approach is proposed in [35]: a set of features which represent radial profile are used. These features are base on the value of mean pixel intensity at given distances from the center of the analyzed region.

There are some attempts aiming to decrease of the IR data ambiguity . One approach is proposed in [35] where data from several IR sensors with frequency bands 3-5 μm and 8-12 μm are combined together. In [36] it is proposed to use additional information from video camera to decrease the influence of grass and other visible clutter those thermal signatures can be very similar to the ones of landmines. *Visual imaginary* provides additional features for object discrimination which improves the detection result.

In addition to simple passive IR measurements a research in the areas of active techniques is ongoing in order to improve the reliability of the method. An approach of *polarized infrared imaging* is specifically designed for detection of surface laid landmines. This method relies on the fact that the surface of a landmine polarizes the reflected IR radiation because of its regular structure (flat surface) while most natural objects do not have this ability. The IR radiation is measured through the polarization filter allowing to obtain images of higher contrast where the landmines are better distinguishable from natural clutter. Some examples of application of this method can be found in [37, 38, 39, 40].

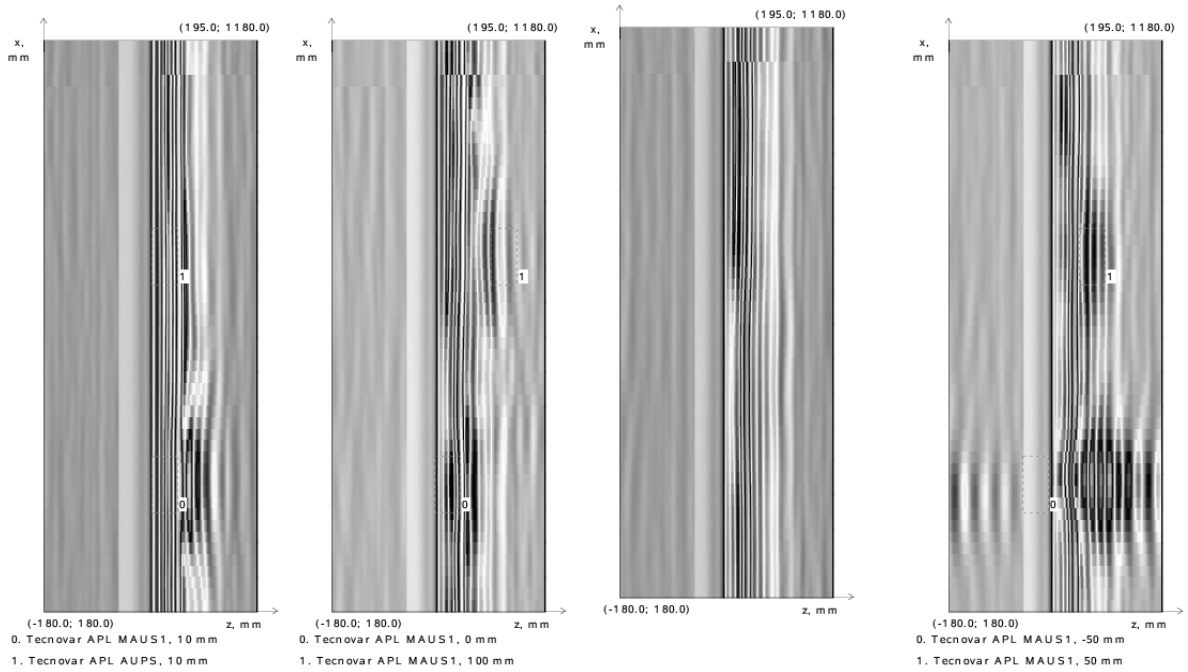
Active infrared imaging intends to decrees the influence of the environmental conditions on the IR measurements. In this case the ground surface is artificially heated usually by using a microwave heater (for example, a heater similar to the one used in microwave ovens) and then the measurements are performed. This approach can operate continuously and provide more predictable results than the passive techniques. However, the presence of the heater significantly increases the weight and power consumption of the system. Moreover, there are some investigations suggesting that the microwave heating may be dangerous because the metallic parts of a landmine may be overheated which can cause its detonation [41].

1.4.3 Ground penetrating radars

GPR is a geophysical technology widely used for subsurface imaging. This method is based on the following principle:

1. A *transmitting antenna* emits short pulses of high-frequency electromagnetic waves.

2. The electromagnetic pulse is reflected from a buried object or a boundary with different dielectric constants. The reflected signals arrive back to the *receiving*



(a) Y=250mm (b) Y=750mm (c) Y=950mm (d) Y=1250mm

Figure 1.17: Sample GPR B-scans obtained by ERA SPRScan during MACADAM experiment on the test scenario 2 (clean agricultural soil) [3]

antenna at different times which depend on the depth of reflection.

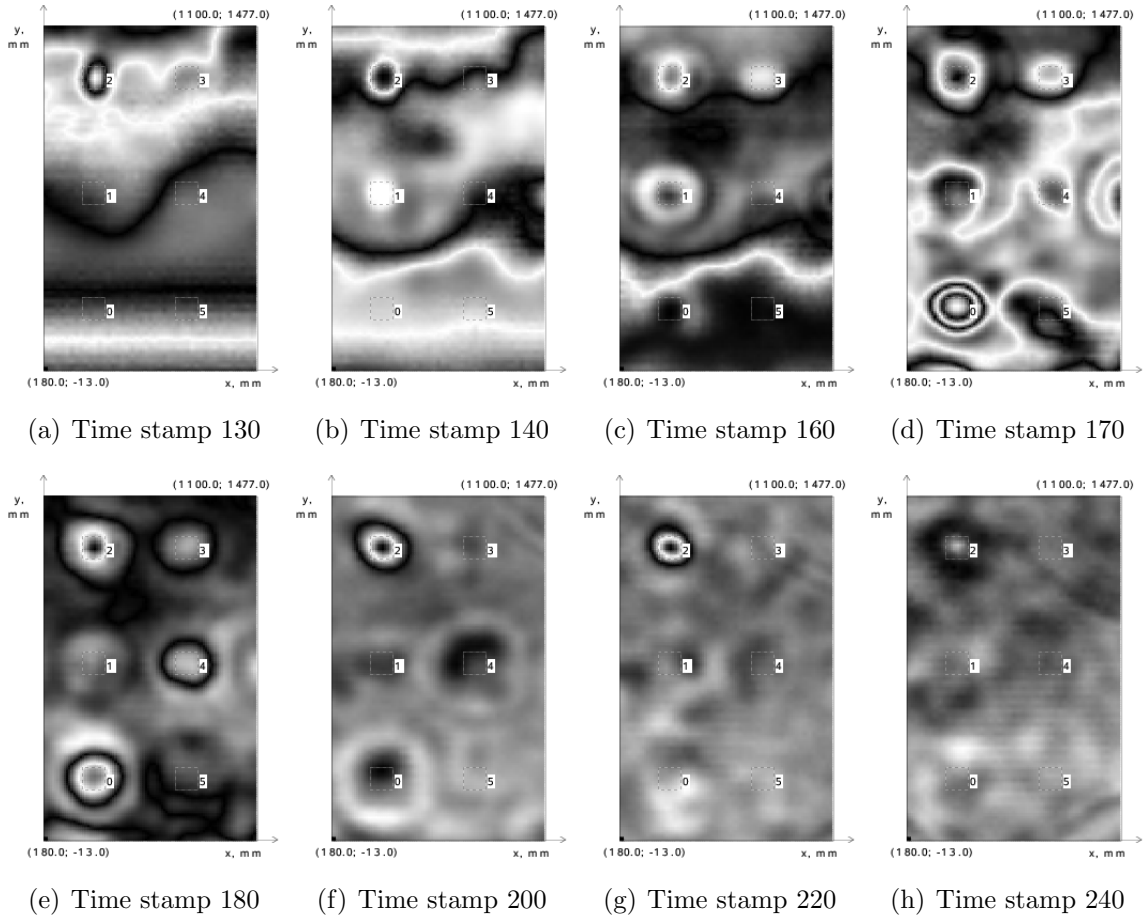
3. The receiving antenna records samples of the time varying reflected signal providing a scan which contains around 250 points, called A-scan.

4. The A-scan provides information about variation of dielectric properties at different times of reflection. The time of reflection roughly represents the depth of reflection. However, there is no exact transformation due to the unknown properties of the materials where the signal is propagating.

The A-scans are difficult to analyze because they do not contain any reference information. Thus, they are usually combined to form B and C-scans:

- B-scan is an image formed by A-scans obtained along a linear scanning path (for example, along axis X) as shown in Figure 1.17. A buried object is usually represented on the B-scan as a set of hyperbolas.

- C-scan is an image formed from the points of A-scans obtained at different X-Y locations at the same depth. Due to the complex transformation between the time and the depth of reflection, C-scans are often formed for a certain time of reflection instead of a certain depth as shown in Figure 1.18. Such images are representative enough if the soil is homogenous. In the ideal case a C-scan can show the real shape of the buried object.



(a) Time stamp 130 (b) Time stamp 140 (c) Time stamp 160 (d) Time stamp 170

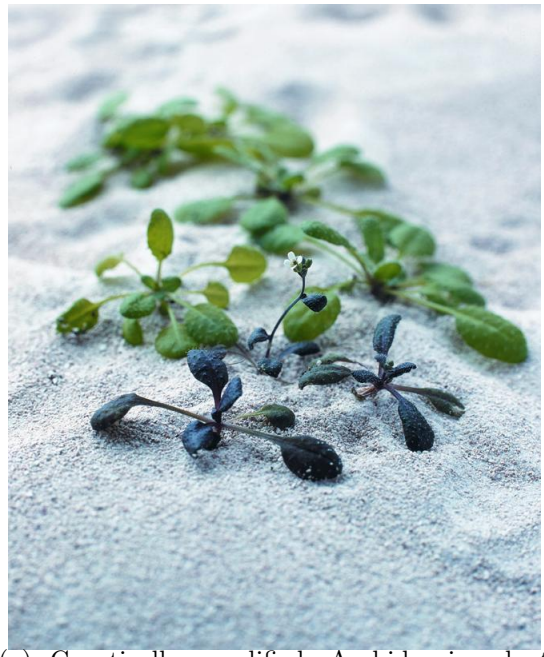
(e) Time stamp 180 (f) Time stamp 200 (g) Time stamp 220 (h) Time stamp 240

Figure 1.18: Sample GPR C-scans obtained by ERA SPRScan during MACADAM experiment on the test scenario 2 (clean agricultural soil) [3]

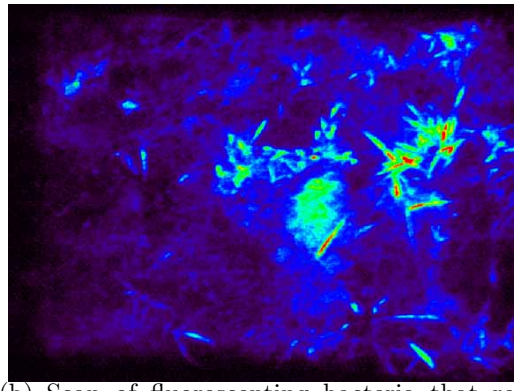
The frequency of the transmitted radio pulse determines its penetration depth. For the purpose of humanitarian demining GPR antennas with frequencies around 1-2 GHz are normally used. This allows to detect antipersonnel landmines up to the depth of around 30 cm. It can be seen from Figures 1.17 and 1.18 that landmines represent clear signatures on the GPR data in case of clean soil. However, GPR is also sensitive to a large amount of clutter objects and any dielectric heterogeneity of the soil. To recognize the landmines in presence of clutter B or C-scans are usually employed. In the case of B-scan the analysis is typically based on models of landmines and clutter [34].C-scans are analyzed using computer vision techniques similar to the ones used in the case of thermal imaging [38].

1.4.4 Other sensing technologies

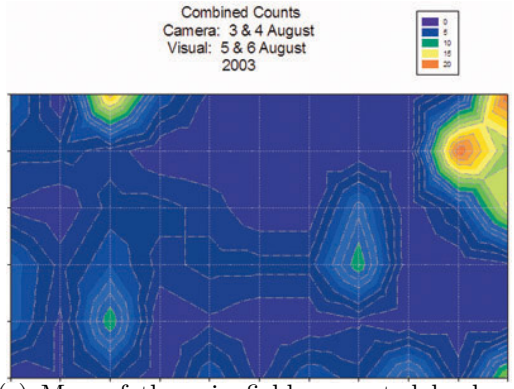
The most developed technologies for landmine detection were considered in the previous three sections. The main problem of these approaches is their low selectivity for the discrimination of landmines and clutter. Thus, there is an ongoing research in other



(a) Genetically modified Arabidopsis plant which changes its color in presence of nitrous oxide (image produced by Aresa Biodetection ApS, Solvgade 14, kld., 1307 Copenhagen K, Denmark, `http://www.aresa.dk/`)



(b) Scan of fluorescenting bacteria that responds to trinitrotoluene (3 by 4 meters area with mine targets within 1 meter of indications), image produced by the demining research team at the University of Western Australia, `http://www.mech.uwa.edu.au/jpt/demining/`



(c) Map of the minefield generated by bees [42]

Figure 1.19: Biotechnologies for landmine detection

areas in order to provide additional detectors. The most important domain is the detection of explosive materials because the presence of explosives is a selective property of a landmine. Trinitrotoluene (TNT) is a common explosive material used in modern landmines. It should be mentioned that the detection of explosives is only possible if it is leaking from the landmine into the surrounding soil.

There are several physicochemical methods for explosive detection which show good results in well controlled laboratory conditions. However, their implementation for a real outside environment meets a lot of obstacles. Such methods are based on the spectroscopic analysis such as mass-spectrometry and nuclear quadrupole resonance [43, 44, 45, 46]. Usually these devices are very expensive, slow and heavy. Thus, in the current stage of development, they should be considered as additional sensing technology which can be used for reduction of false alarm rate provided by faster sensors described

before.

Biological systems provide much better results for explosive detection due to very sensitive capabilities of olfaction. Specially trained demining dogs are widely used to assists the manual humanitarian demining. It is known that some animals have even better olfaction than the dogs. There is also some work done in the area of using rats for explosives detection [47] [48]. Using of highly developed animals like dogs in the automation system for landmine detection is not straightforward, however, other biological technologies may be used as an additional landmine "detector":

- Aresa Biodetection ApS is developing a genetically modified plant which changes the color of its leaves in presence of explosives [49] as shown in Figure 1.19(a)
- Honey bees can be trained to provide a "map" of explosives concentration [42] (see Figure 1.19(c))
- A genetically modified fluorescent bacteria may be also used to detect TNT as shown in Figure 1.19(b) [50], [51]

1.4.5 Algorithms for landmine detection

Each sensor technology has an associated appropriate algorithm to be used for landmine detection. Some of these techniques were mentioned in the previous sections. In this section several basic principles valid for different detectors are summarized.

There are basically two approaches for landmine detection:

- *Physical model based* where it is assumed that the response of the sensor for landmines and clutter can be modeled. Then, the difference between the models is used to discriminate landmines from clutter [30, 52, 31]. A certain implementation of this approach depends on each sensor technology.
- *Learning based* approach does not assume any specific model for the sensor response (the model is hidden). Instead, a set of *classification features* which characterize the response are used. The relations between the features and the detected object are determined by learning the hidden pattern from *training data*. Such approaches are based on the techniques of *pattern recognition* [53]. Then, the landmine detection is usually considered as a classification with two classes: landmines and background. Different classifiers and feature extraction methods were used in previous works: Bayesian classification is used in [33, 52, 54, 55, 56], Support Vector Machines in [57, 52], Dempster-Shafer classification in [58, 59, 60], neural networks in [61, 62, 63], etc. In some works the different methods are compared, however, a definite conclusion about the most suitable approach is not made. This can be explained by the fact that the performance of landmine detection is affected

by many parameters including the techniques for estimation of classification features and preprocessing of the sensor data.

There are also approaches which combine the both mentioned strategies. For example, the parameters of the physical model can be estimated using one of the learning techniques [30]. Analyzing the previous work which uses both approaches it can be concluded that the model based techniques are less appropriate for the detection of landmines in real conditions and at least some amount of learning should be incorporated in the algorithm. Moreover, the physical modeling is well applicable only for some of the sensor technologies (mostly, metal detection) and is usually not appropriate for using of multiple sensors together.

The basic two-class learning-based strategy is usually not effective enough due to the large complexity of the landmine detection task: the division of all the objects into categories landmines and background is not logically sound because of the presence of not-landmine objects which significantly differ from the background. There are few works which pay attention on this issue. A two-step strategy for landmine detection using handheld detector is used in [30]. Several works consider a preprocessing step of Region-Of-Interest extraction [35, 64, 23, 34].

The currently existing algorithms for landmine detection do not achieve the required high quality yet; the research in this area is ongoing in several directions. It is widely accepted that the integration of data from several sensors, called *sensor fusion*, should improve the landmine detection in comparison to the performance of each sensor used alone. This fact was also confirmed experimentally in several previous works [65, 59, 66]. In this situation the development of the learning-based algorithms is an important step forward to develop a reliable landmine detection system.

Sensor fusion

In general, the process of *multisensor integration* can be implemented in different manners [67]:

- Separate operation for each sensor
- Controlling one sensor by the information from another one
- Fusion of data from several sensors

The *sensor fusion* approach is widely accepted as the most promising approach to improve the landmine detection process. The term *sensor fusion* has several definitions, which can be found for instance in [68, 67], and basically means any techniques which assist the integrated usage of several sensor sources to achieve a common goal or decision.

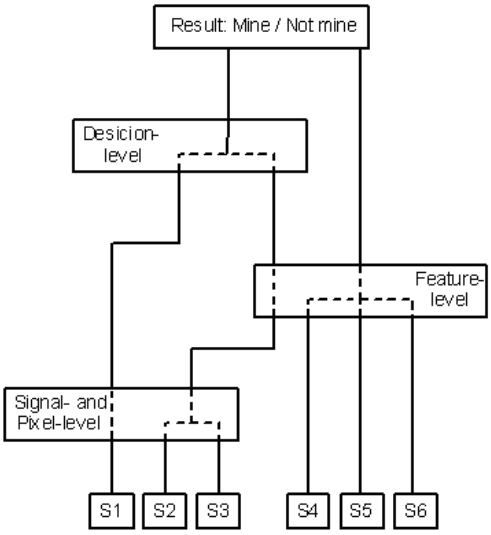


Figure 1.20: Sensor fusion techniques for landmine detection

Four main types of sensor fusion can be distinguished: signal-level, pixel-level, feature-level and decision-level fusion [67]. The signal and pixel-level approaches provide the highest degree of integration. However, they can be used only for sources with equal working principles. While the feature-level fusion can be used for any combination of sensors: the integration is performed using *features* which can be extracted from heterogenous sources but have the same structure themselves. The decision-level fusion represent the highest level of abstraction from the used sources. The integration is performed after the decisions were obtained for each source separately. The structure of the most appropriate method is determined by the specifics of the used sensors and the final goal of sensor fusion. In the case of landmine detection the following properties should be noticed:

- Sensors have different physical principles
- Sensors are not selective for the landmines leading to a high false alarm rate

The sensor fusion techniques which can be used for landmine detection are summarized in Figure 1.20. It can be seen that the final stage of sensor fusion should be always a decision or feature-level algorithm because the goal of the landmine detection is a binary decision about the presence of a landmine. The decision-level sensor fusion combines the decisions made separately by each sensor, while in the feature-level the raw data (classification features) from all sensors are processed together. This processing can be performed on the raw data or on the result provided by a lower level fusion algorithm (signal- and pixel-level fusion).

There is a lot of research made in the domain of sensor fusion for landmine detection which utilizes principles shown in Figure 1.20. Examples of using the feature-level sensor fusion can be found in previous works for different combination of sensor sources:

metal detector and GPR are used in [58, 61], IR camera and GPR are used in [38, 33], and a combination of three sensors (metal detector, IR camera and GPR) are used in [69, 70, 71] An example of pixel-level sensor fusion used for IR cameras with different bands can be found in [72, 73]. The decision-level fusion is the simplest approach in terms of implementation. It is explored for a combination of metal detector and GPR in [30, 61, 74], for IR camera and GPR in [75], for a combination of IR cameras in [65] , and for the three sensors in [59, 76, 77, 69, 60, 78, 66]. There is also some research in the domain of sensor fusion for landmine detection without considering any specific sensors [79, 80] In some works these two concepts are compared [61, 76, 69, 60], but there is no experimentally proven conclusion which strategy is better. However, it is considered that the feature-level fusion is more preferable because it provides a deeper integration of data.

1.5 Problem statement

The present work is focused on development of a sensor fusion approach for landmine detection assisted by autonomous mobile demining robot. The main goal is creation of new sensor data processing algorithms and implementation of them for a prototype scanning platform. It is assumed that the demining robot can be equipped with the available sensor technologies, such as metal detector, infrared sensors, and ground penetrating radar. The algorithms should be implemented on a prototype demining robot which has cartesian mechanical structure and is controlled by pneumatic actuators. This simple scanning platform is in accordance with the idea of simplicity to be used throughout the developed algorithms which should not be computationally expensive to be implemented on ordinary hardware.

Fusion of results obtained from different sensors providing the required quality of landmine detection is a key issue. However, the acquisition of reliable data from the sensors is not possible without correct positioning of the sensors and without an adequate path planning for the robot providing full coverage of the scanned area. Therefore, another important goal in this thesis is development of new algorithms for a robot-assisted sensor data gathering including the path planning and the positioning of the platform.

The combination of the two mentioned tasks and implementation of them in one framework on a demining robot is another target leading to new important step forward in the direction of creation of reliable autonomous mobile demining robot.

1.6 Contributions

The main task of this work, sensor fusion for landmine detection, raised several related problems. That is why the original contributions made in this work belong to three research topics: data processing algorithms for automatic multisensor landmine detection, path planning algorithms for mobile scanning platform to assist the sensor data gathering, and software implementation.

Landmine detection

- A dedicated multi-stage strategy was developed for landmine detection [81, 82]. Following this approach, the raw sensor data are first processed to extract regions which contain objects suspected to be landmines. Then, the detected objects are classified into two groups: man-made objects and natural objects. On the final stage, the man-made objects are classified to be landmines or other objects. The last two steps represent together the *recognition stage* of landmine detection. The advantage of this strategy is the ability to perform landmine detection according to the quality of the available sensor data: the better the sensor data, the larger number of stages can be accomplished. Different types of multi-stage strategies were used in previous works without paying attention on this issue.

- A novel online algorithm for automatic extraction of ROI was developed in order to complete the first stage of landmine detection, detection of suspected objects, [81]. In the previous works a similar problem is usually solved using computer vision techniques including circle detection, contour detection, etc. However, it was found that such techniques are inconvenient or completely inappropriate for the online implementation required in this work. The developed approach is able to detect the region right after it was fully scanned by a scanning device. Other advantages of the algorithm include the low number of parameters which need to be adjusted (two sensor dependent parameters), the general object model which allows to detect objects of different shapes, and an ability to handle changes of environmental conditions.

- Several novel classification features were developed to be used for landmine recognition together with the ones adopted from other fields of pattern recognition or previously utilized for landmine detection [83]. The developed features help to analyze the shape and nature of the objects and to improve the recognition results.

- Taking into account the specificity of the landmine detection task the new concepts of *selective training* and of *dominant class* were developed after the analysis of the classification features. It was revealed that the large number of features

have bimodal distributions for one of the classes. The larger maxima are usually highly overlapped, while the smaller one is better separated. This is interpreted as specifics of the landmine recognition where the signatures of the landmines and other objects can be very similar: the property reflected by the classification feature is almost equally present in both classes of objects (larger maximum), however, a small amount of objects can be separated (smaller maximum).

The developed approaches alter in some way the training of the classifier using only the well-separated maximum of the feature distribution. As a result this improves the recognition .

- In the framework of feature-level sensor fusion implemented in this work a concept of *mixed features* was developed [82]. It implements an idea of signal-level sensor fusion used in the case of heterogeneous landmine detection sensors. This strategy improves the stability of the feature and allows to solve the problem of missed features. However, it can be used only for high-level features related to shape and nature of the object. In previous works the concept of signal-level sensor fusion for landmine detection was used only for the sensors which utilize the same physical principle.

Path planning and positioning

- To improve the reliability of the cartesian pneumatic platform special algorithms for simultaneous movement of parallel cylinders and for smooth landing of the legs were developed.

  Attempting to solve the remaining problems an additional incremental positioning system based on vision was developed. The system utilizes a CCD camera pointed downward to detect located on the ground natural landmarks and their relative movements. For this purpose novel algorithms for detection and association of natural landmarks were developed [84]. In previous works natural landmarks are usually utilized for global localization and the local positioning is performed using more computationally expensive techniques.

- A novel algorithm for *unknown area coverage* to allow the scanning platfom to acquire the sensor data at each point of the specified area [85]. The algorithm satisfies the restrictions applied by the demining task: the coverage path must be regular to allow acquiring data from the sensor and the path of the platform should always lie in the already covered area. An addition advantage consists in its poor dependence on the robot odometry which makes it suitable for other mobile robot applications like cleaning [86].

Software implementation

- The developed algorithms for landmine detection and path planning were implemented in one software framework written in C++. The diversity of the implemented problems allowed to develop a convenient structure of *object abstraction* for mobile robot programming which was not available in the existing software frameworks. For the experimental part of this work two mobile devices with completely different mechanical structures were used: the legged scanning platform and a wheeled differential drive mobile robot. To allow a transparent transition of the algorithms between platforms an appropriate *robot abstraction* was implemented. With this software the developed algorithms were tested on the wheeled robot and then used for the demining platform with no change.

- A graphical operator interface was developed in order to show the state of the robot in real time and to allow manual control (even is the demining robot is developed to be a completely autonomous device). The functions of this software were extended to assist the research in landmine detection allowing to test the developed algorithms offline and maintain a database of landmine signatures. By doing this, it is simulated that the testing is performed in real conditions on the robot avoiding any transitions of the developed algorithms for the real usage.

An important contribution of this work is the combination of landmine detection and path planning algorithms assuming the restrictions of one to another. In previous works only one aspect of the problem is usually considered and often the main attention is paid to the mechanical design of the robot or the landmine detection is considered apart.

1.7 Short Thesis Overview

Chapter 1 provides an introduction to the problem of humanitarian demining. Here an overview of the landmine problem in general and the specifics of humanitarian demining is given. The currently available solutions together with the ones being still in development are reviewed. Then, the discussion is focused on the sensor technologies and algorithm for automated landmine detection. Finally, the goals of this research are also formulated here.

Chapter 2 provides a global overview of the techniques developed in this work and an overview of the used hardware. The prototype demining robot used for the experiments is introduced providing information about its mechanical structure, installed landmine detection sensors, control hardware, and structure of the control software. General strategy used in this work for the development of landmine detection and path planning

algorithms is also presented here together with the tools which were used during this research.

The approach for sensor fusion based landmine detection is described in detail in Chapters 3 and 4. Chapter 3 is dedicated to the first stage of the landmine detection process, the problem of suspected object detection performed on the raw sensor data during scanning. Here, the novel approach developed for Region-Of-Interest extraction is presented together with the results of its experimental testing.

Chapter 4 provides a description of the algorithms developed for the last stages of landmine detection. The classification features used for landmine recognition including the novel features developed in this work are described in detail together with techniques used for the feature analysis. The newly developed concepts of *mixed features*, *selective training* and *dominant class* are presented. The Chapter is finalized by the experimental results which allow to verify the proposed approaches.

The positioning of the mobile scanning platform is described in Chapter 5. The two localization systems are investigated: the odometry of the robot and the additional vision system. The description of the developed vision based localization system includes the novel algorithms for detection and association of natural landmarks and results of their experimental testing. Several techniques are proposed here in order to improve the odometry system of the robot as well.

Chapter 6 provides the description of the online unknown area coverage algorithm with simulation and experimental results of its performance.

Chapter 7 presents the results of experimental tests performed with the prototype demining platform on the test minefields located in Meerdaal bomb disposal unit.

Finally, the conclusions and recommendations are presented in Chapter 8.

Chapter 2

System Organization

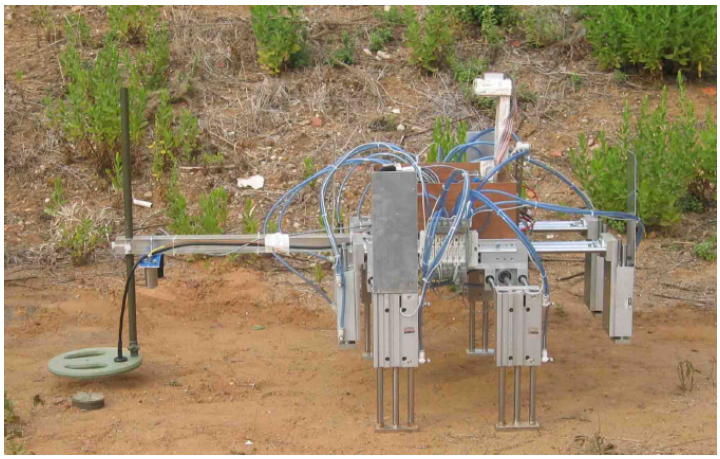
This work is part of the project DEMINE which is being developed in Institute of Systems and Robotics, Coimbra, Portugal. The goal of the project is to develop a mobile robot able to explore a certain area and detect antipersonnel landmines located in it. This chapter describes the system developed in this work as a whole together with the tools used to assist the development. The used prototype demining platform is described in Section 2.1 together with the improvements made in the preset work. Sections 2.2 and 2.4 provide an outlook on the basic strategies which were used during the development of algorithms for landmine detection and path planning. The detailed explanation of these approaches can be fond in the subsequent chapters. The additional differential drive mobile Nomad Super Scout robot used for testing of path planning algorithms is also introduced in Section 2.4.

The experimental environment used for testing of landmine detection algorithms in this research, which was composed of public databases of landmine signature data and experimental test minefields, is presented in Section 2.3.

Finally, the structure of the software in which all the developed algorithms were implemented is described in Section 2.5. The details about implementation of each particular algorithm are located in the section *Implementation* of each chapter.

2.1 Demining Robot Prototype

The main objective of this research is the development of new algorithms for landmine detection and path planning. The practical implementation of these algorithms is handled with assistance of a previously developed prototype demining platform LADERO [22]. It has a cartesian mechanical structure with four axes supported by eight legs (see Figure 2.1). The movements of the robot are performed using discrete pneumatic actuators powered by an external air compressor. The following control and sensing equipment was already installed on the robot at the beginning of the present work [4]:



Dimensions	
width	750 mm
length	750 mm
height without equipment	280 mm
Weight without equipment	48 kg
Maximum obstacle height	150 mm
Air pressure	6 bar
Maximum step length	20 cm

Figure 2.1: Prototype demining robot LADERO before the start of this work and its technical characteristics [4]

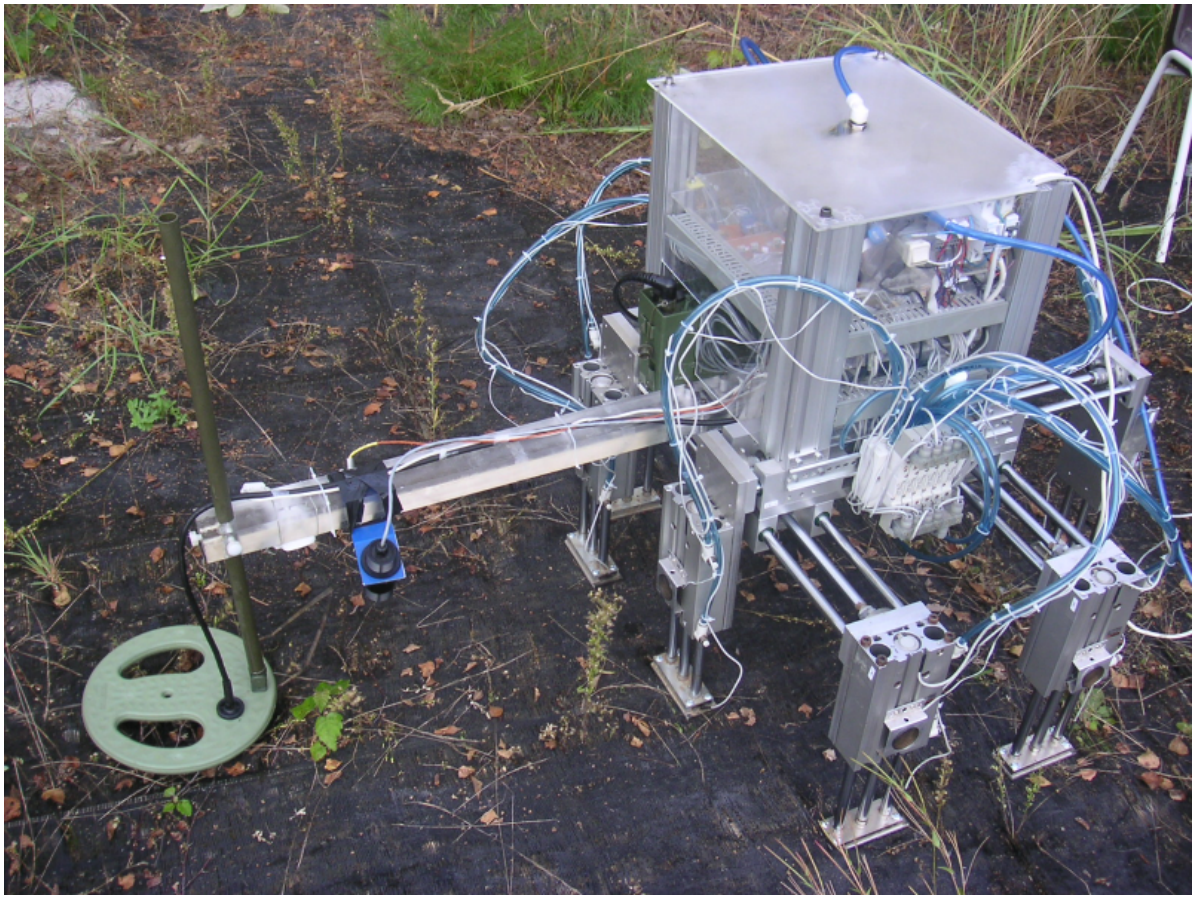


Figure 2.2: The demining robot prototype by the end of this work

- Main control board (Card12 based on Motorola HC12) able to control the actuators and acquire data from sensors via slave boards;
- Slave boards based on Microchip PIC microprocessors to control: 4 ultrasound sensors, metal detector, electronic compass, and end sensors of the cylinders;
- Schiebel All Terrain Mine Detector (ATMID) metal detector with acquisition board providing the audio frequency output signal to the main control board;
- Two infrared sensors with acquisition boards providing analog output signals to the main control board;
- End sensors of the pneumatic cylinders measuring their limit positions;
- Ultrasound sensors to measure positions of the legs relatively to the robot body (odometry system);

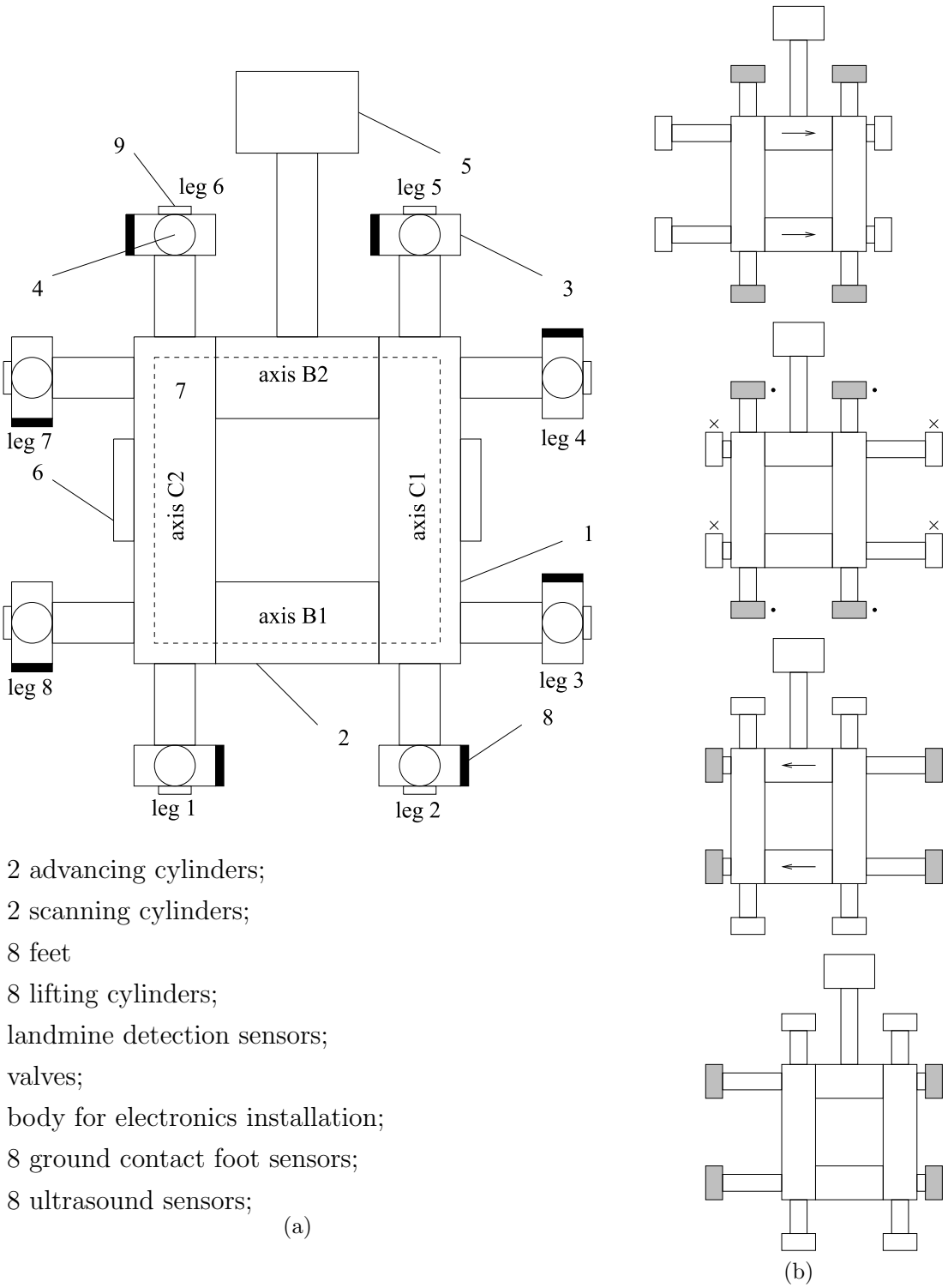
The installed hardware and existing software algorithms did not provide all the required means for a reliable landmine detection. For the purposes of this research the following improvements of the platform hardware were made:

- The audio frequency output used from metal detector was substituted by a more informative analog signal as described in Section 2.1.2;
- The odometry system was changed to a more reliable one based on discrete sensors installed on the cylinders (see Section 5.3);
- Additional equipment was installed: onboard PC, video camera, ground contact foot sensors, ultrasound sensors for obstacle detection, power supply;
- A mechanical structure to support the control electronics was constructed (see Figure 2.2);

Next sections describe in detail the scanning principle of the robot and the installed equipment.

2.1.1 Scanning Principle

The functional structure of the robot is simple as shown in Figure 2.3(a) (additionally there are also end sensors installed on each cylinder and control electronics not shown here). The robot can perform two types of cartesian motions in order to scan a specified area: a *scanning step* performed along X-axis in order to acquire data from the current scanning line, and an *advancing step* along Y-axis in order to move to the next scanning line and start the motion in the opposite direction. The rotations are theoretically

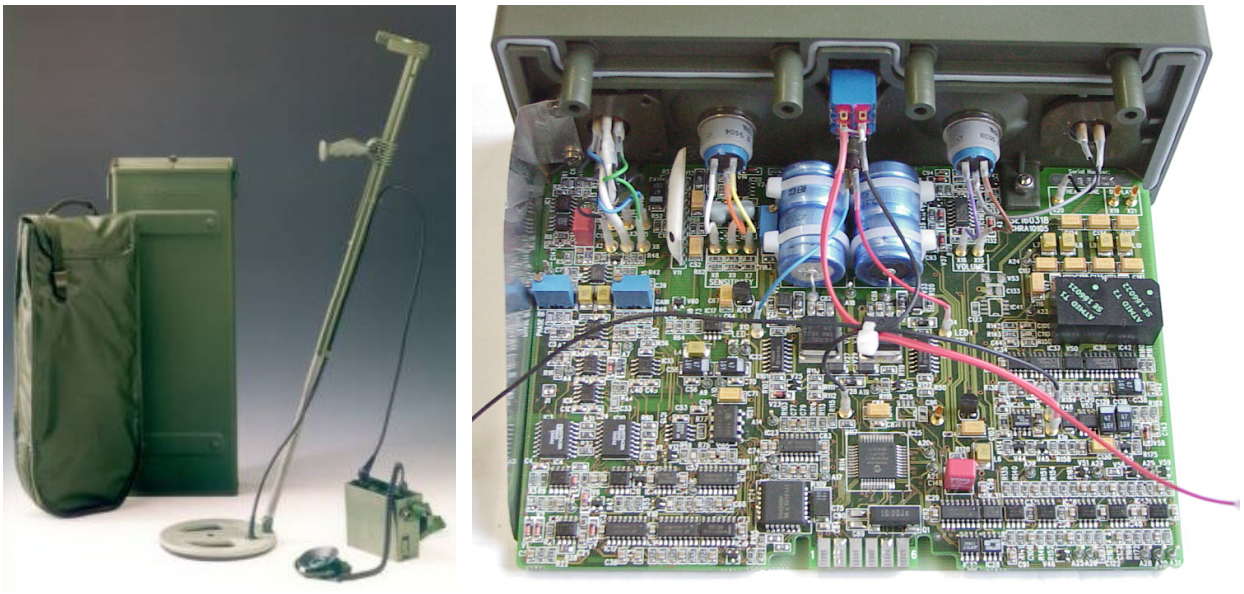


1 - 2 advancing cylinders;
2 - 2 scanning cylinders;
3 - 8 feet
4 - 8 lifting cylinders;
5 - landmine detection sensors;
6 - valves;
7 - body for electronics installation;
8 - 8 ground contact foot sensors;
9 - 8 ultrasound sensors;

(a)

(b)

Figure 2.3: A functional scheme of the robot (a) and its basic scanning step (b)



(a) (b)

Figure 2.4: ATMID metal detector

possible [22]. However, experiments show that the performed motion is unreliable due to the slippage and deformation of the robot body, so it cannot be used in practice to achieve a predictable rotation.

The basic scanning step is performed in several stages shown, for example, in Figure 2.3(b) for a left-to-right step:

1. The feet of *supporting axes* are landed, and the *moving axes* are shifted to the right;

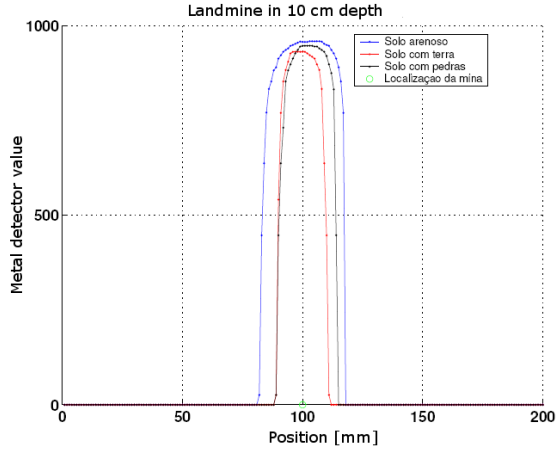
2. The feet of *moving axes* are landed and the feet of *supporting axes* are lifted;

3. The *moving axes* are moved left making the robot body to drive right;

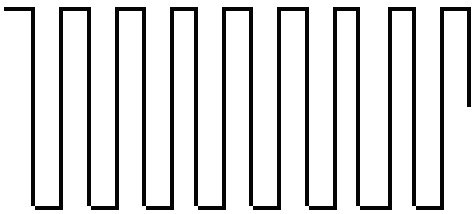
Depending on the step direction the supporting and the moving axes are changing. The *advancing step* is performed in a similar manner, but the moving axes are shifted by a small distance providing a small step.

2.1.2 Landmine Detection Sensors

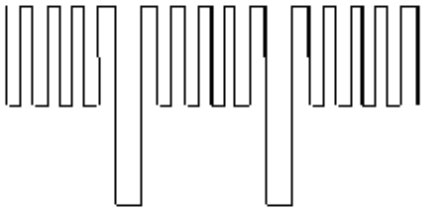
Landmine detection sensors are installed in front of the robot providing a possibility to detect landmines in advance in order to avoid them by the robot. The currently installed sensors are one metal detector and two infrared sensors.



(a)



(b)



(c)

Figure 2.5: Signature of a landmine using output audio frequency of metal detector [4] (a) and a typical form of audio signal as response to a distant (b) and a close (c) metal object

Metal detector

The Schiebel ATMID metal detector installed on the robot is a professional metal detector certified for manual demining Figure 2.4(a). Its control electronics is able to control one of the two available search heads: continuous (ATMID) and pulsed (AN 19/2). The continuous mode is known to be more sensitive. Thus, the ATMID search head is used in order to make possible the detection of small metal objects. The control board used to acquire the output signal of the metal detector processes the frequency of the audio signal provided for the human operator [4]. However, the informativity of this signal is relatively low since the obtained landmine signature has very rigid edges as in the case of an almost discrete sensor Figure 2.5(a).

In order to obtain a better output signal the control electronics of metal detector was investigated in more detail. It was found that the audio signal shape depends on distance and size of metallic object as shown in Figure 2.5(b) and 2.5(c). The processing of this signal is usually simplified (analyzing only the main frequency) which lowers the provided information. However, the audio signal is the most widely used, probably because it is the easiest to obtain from the control electronics. The control board of ATMID metal detector is based on a microcontroller which outputs the audio frequency signal.

Spectral band 6.5 - 14 μm

K-type output

(a) OS36-10-K-80F

Spectral band 8 - 14 μm

J-type output

(b) OS65-J-R2-4

Figure 2.6: Infrared sensors installed on the robot

Meanwhile, most of the signals received by the microcontroller are preprocessed before. Thus, there are several testing points on the circuit that provide intermediate data of the process. Based on this, at least two testing points can be considered as outputs of the metal detector (see Figure 2.4(b)). A26 TP provides an analog signal changing from -5V to 5V. It is located after the compensation of air and ground properties (stored during balancing procedure) and before filtering. The speed of the search head does not affect this signal and small metal objects cause a constant value. A29 TP is situated after filtering. When a small metal object is near the search head and motionless it does not change A29 signal. The sensitivity potentiometer located in the front panel does not influence A29 and A26 signals. Both signals change their values to negative if there is an extra metal object and to positive if there is less metal than it was detected during air and ground compensation. Thus, the negative part of the signal can be used to estimate the amount of the detected metal, and the positive part to determine if a new compensation is necessary.

The slave control board for the metal detector was modified to provide the negative part of the A26 signal as the output value of the metal detector.

Infrared sensors

Two infrared sensors shown in Figure 2.6 are also installed on the robot. They are based on the thermocouples with different characteristics: K and J. The sensors provide signals proportional to the temperature of the soil at the current location. To obtain an IR image of a larger area the sensors should be moved over the area by a scanning device. An example of the data obtained from IR sensor with K-type output (shown in Figure 2.6(a)) can be seen in Figure 2.7. It should be noticed that these signals are very noisy due to the natural fluctuation of the thermal radiation from the soil.

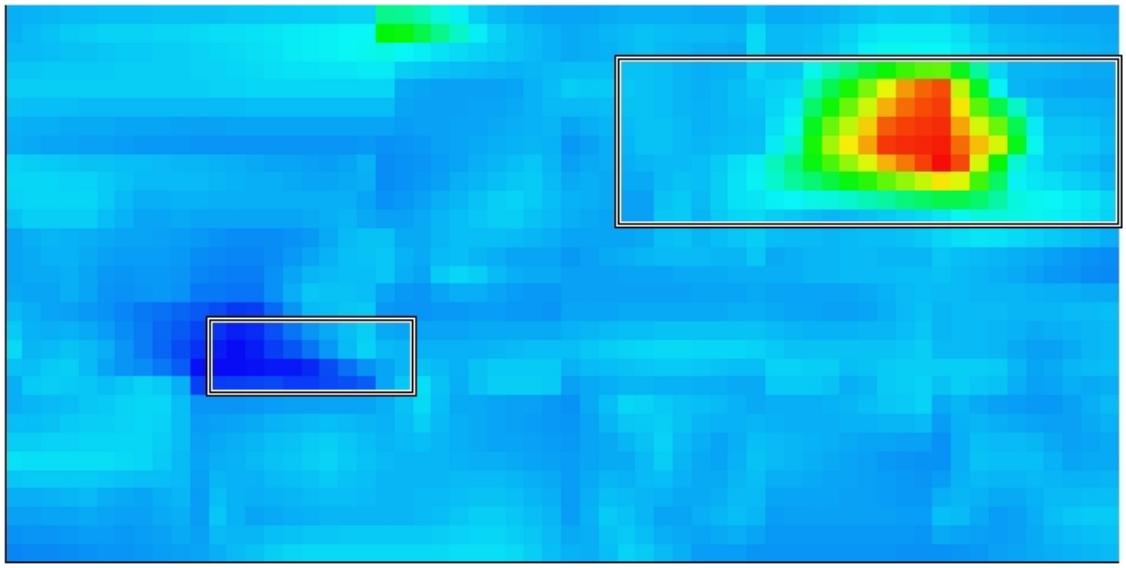


Figure 2.7: Experimental data obtained from an IR sensor in the area containing one hot and one cold object.

2.1.3 Control Hardware and Software

The structure of the robot control hardware and software is modular. The tasks which belong to different layers are distributed among the layers of hardware:

- *Onboard PC* is a VIA EPIA (VIA Embedded Platform Innovative Architecture) embedded PC dedicated to high-level control tasks: path planning, navigation, obstacle avoidance, landmine detection, archiving of the data, and communication with graphical interface. It is operated under Linux OS.
- *Main control board* is an Elektronikladen Card12 module based on Freescale MC912DG128A processor. It controls the actuators of the robot, acquires data from infrared sensors, and communicates with slave modules and with the onboard PC.
- Slave modules are based on Microchip PIC16 microcontrollers and were developed previously as reported in [4]. They are connected to the main control board via the SPI bus and control the peripheral functions: metal detector, end sensors of the cylinders, ultrasound sensors, and an electronic compass.

The structure of the control hardware of the robot is shown in Figure 2.8. The main control board and the onboard PC are connected via serial link using a protocol based on Modbus. Using this protocol the PC can access the memory of the robot where its current state is stored (middle table in Figure 2.9) using the standard Modbus *read registers* command (0x04). The control commands which the PC can execute on the robot are coded as user-defined Modbus commands: 0x41 - do step, 0x42 - rotate, 0x43

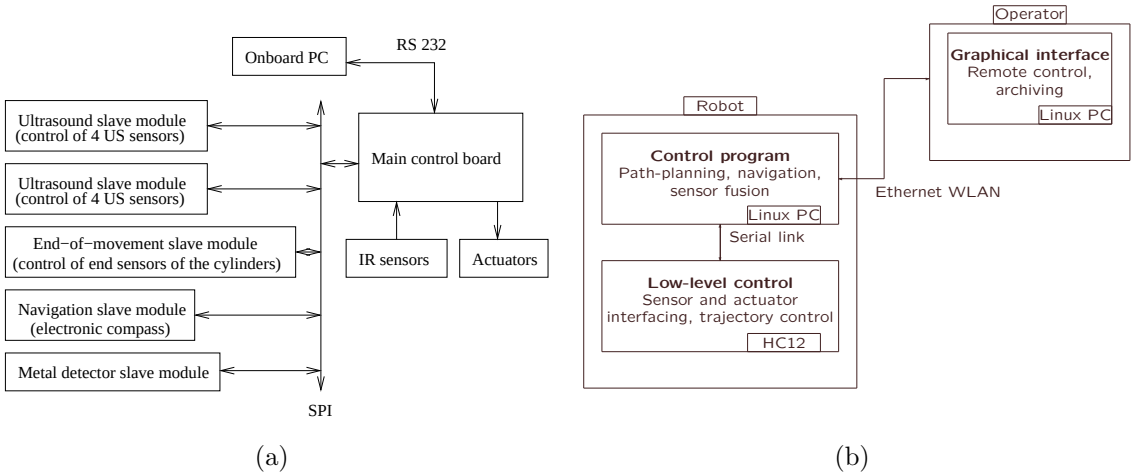


Figure 2.8: Software and hardware organization

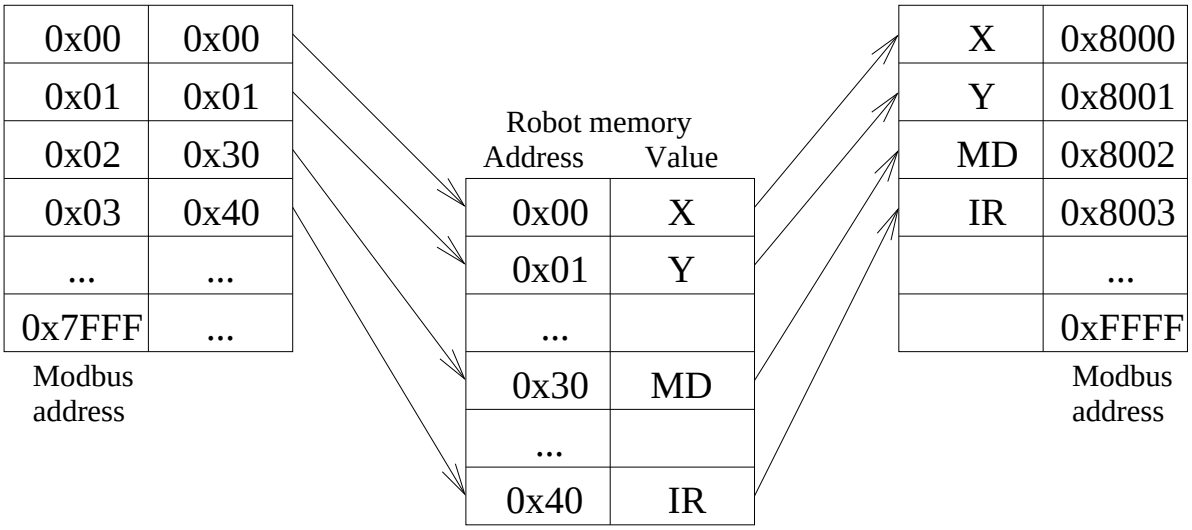


Figure 2.9: Organization of the robot memory for Modbus-based protocol

- stop, 0x44 - calibrate. Usually, the communication between the PC and the robot consists in repeating the *read registers*, command to update the current state of the robot and sending the control commands. The memory of the robot contains all its parameters, but in most cases only some of them are needed depending on the goal of the control program running on the PC. The necessary parameters may be located in different places of the memory table and require several *read registers* commands to be send. Thus, to simplify the communication an additional memory mapping is implemented on the robot (see Figure 2.9): the Modbus addresses from 0 to 0x7FFF contain the actual addresses of the robot parameters which will be read using Modbus addresses from 0x8000 to 0xFFFF. This way only one *read registers* command is needed to read all the parameters required by the current program. The program must first initialize the mapping (left table in Figure 2.9) using the standard Modbus *write registers*

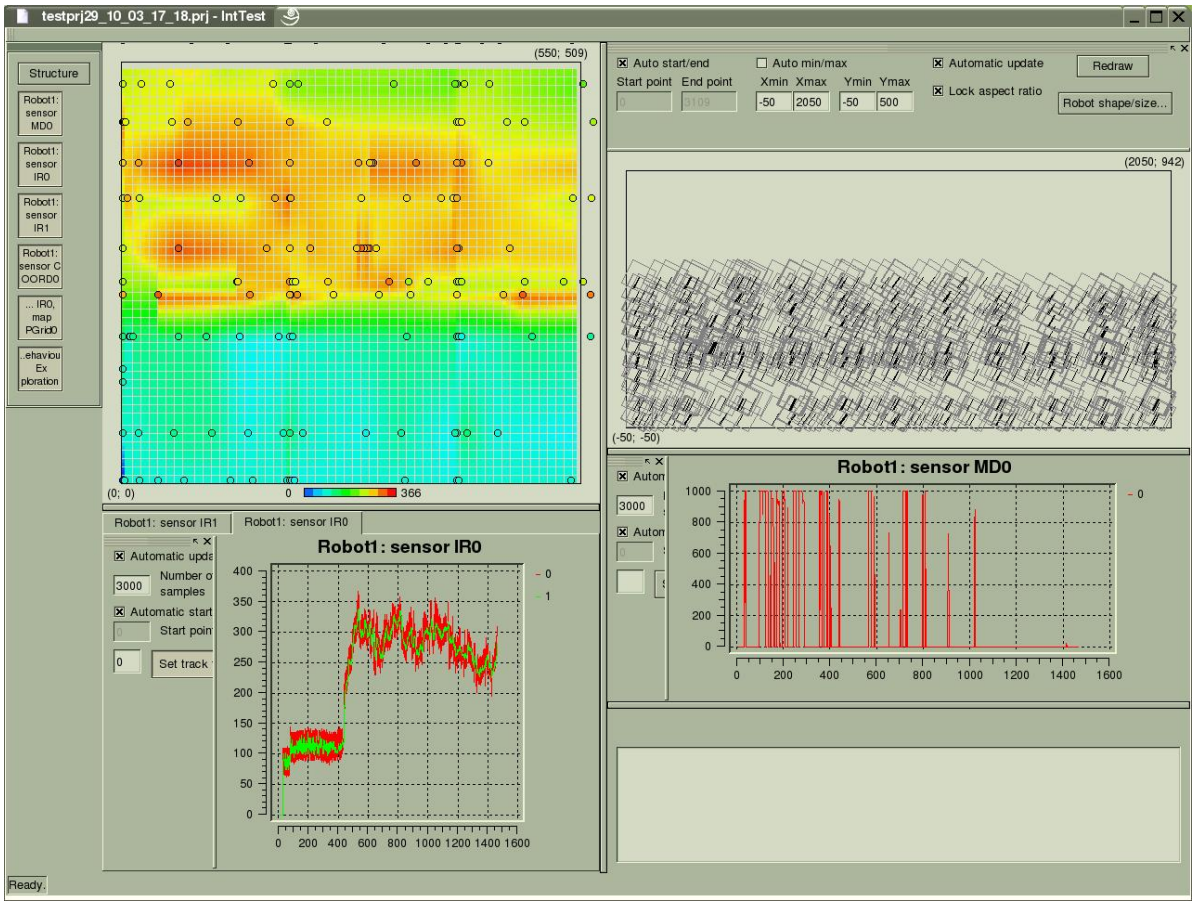


Figure 2.10: Graphical interface screenshot

command (0x10). The detailed format of the commands and the robot memory can be found in Appendix C.

The control software of the robot is also modular following the hardware where it is executed (see Figure 2.8). The *control program* contains all the algorithms developed in this work for landmine detection, path planning, positioning of the robot, etc. Its structure is described in Section 2.5. Part of the algorithms for robot positioning is implemented inside the *low-level control program* because they require low-level control of the robot movements (Chapter 5). Besides the robot software there is also a graphical interface connected to the robot via a wireless LAN. It provides visualization of the robot current state, results of landmine detection and allows to remotely control the robot (an example of the graphical interface screen is shown in Figure 2.10).

2.2 Approach for Landmine Detection

The most developed sensing technologies for landmine detection currently include metal detectors, infrared sensors and ground penetrating radar as discussed in Chapter 1. They are considered here as typical detectors whose data should be combined by means

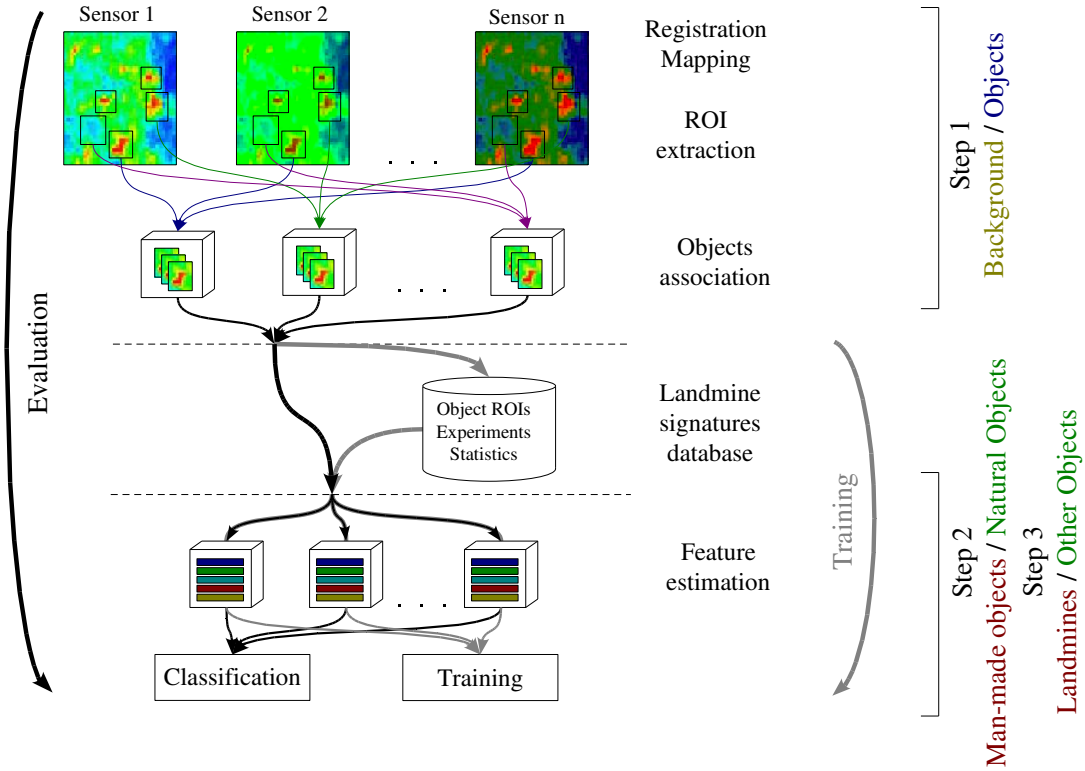


Figure 2.11: General structure of the proposed methodology for sensor fusion

of sensor fusion techniques. The goal of this process is to provide a single decision about presence or absence of a landmine. The approach developed in this work is based on feature-level sensor fusion and has several stages. The multi-stage nature of the approach is necessary to address the specific challenges associated with the landmine detection task.

First of all, it is important to mention that the used sensors do not detect the landmines directly. Instead, they can only distinguish heterogeneity of some physical parameter against the background. This physical property found in a landmine can also be present in other objects, called *clutter*. The presence of heterogeneity signifies only that there may be an object suspicious to be a landmine. The difference between the background and the objects (heterogeneities) gives the largest variation in the sensor signal, while the difference between landmines and clutter is much lower. An attempt to separate the three categories (background, landmines, and clutter) in one classification process is a complicated task because the signatures of landmines and clutter are nearly the same in relation to the background. Thus, the most effective first stage of the landmine detection process seems to be the classification with two classes: *Background* and *Suspicious Objects*. Here *Suspicious Objects* include the actual landmines, other man-made objects, natural debris, and can also represent just heterogeneities of the soil properties. The algorithm for detection of suspicious object developed in this work is described in detail in Chapter 3. It uses the difference of the sensor value and the spatial

information to detect a region, called *ROI*, where the signature of the object is located. In the following step, the ROIs which represent the signatures of the same object for different sensors are combined together forming a complete signature of the suspicious object.

Next stage of landmine detection should finally recognize the landmines from the previously detected *Suspicious Objects*. An important issue at this point is the classification features which can be used for this purpose. While the difference between the background and the objects is well seen on the spatially mapped sensor data (see, for example, Figure 1.13), the signatures of the landmines almost do not contain any visual tokens which would help to recognize them among other objects. As there is no single "selective" classification feature, a large number of unselective features can be used to perform the classification. This situation is similar to the usage of sensor fusion for several nonselective sensors each of which cannot be used alone. The classification features may reflect the difference between signatures based on their shape, statistical and information measures, etc, as described in Chapter 4. The common idea behind the classification features is an attempt to recognize some regularity in the object signature because a landmine has a regular structure with certain locations of the parts and some symmetry. This property is probably the only possibility that can be used to separate the landmines from clutter. However, regularity is also present in many other man-made objects. If the landmines were laid only on the ground surface their structural signatures could be easily recognized from other artificial objects by a vision system, but the subsurface detectors cannot usually provide so rich information.

The recognition stage is further divided in sub-stages which can be performed or not depending on the data quality: the *Suspicious Objects* are classified in classes $Man-made\ Objects$ and $Natural\ Clutter$, and then the *Landmines* are recognized among the $Man-made\ Objects$. The new algorithms for these two stages are developed in the present work as described in detail in Chapter 4. The quality of the sensor data obviously limits the possibilities of the recognition method. Having better sensor data this process could be even extended to the recognition of certain types of landmines. However, such data is not currently available.

Figure 2.11 shows the process of landmine detection described above. One more stage is added to it for the research purposes: the signatures of detected suspicious objects are collected in a database. Each object in the database is associated with the experiment where the data were obtained, the coordinate data, the ground truth, etc. providing an unified approach for representation of landmine signatures. The stored signatures are later used offline for the testing of classification algorithms. The developed database is described in detail in Chapter 3 (Section 3.6).

One of the advantages of the proposed strategy is its ability to account for the quality of the sensor data as shown in Figure 2.12. The landmine detection process can

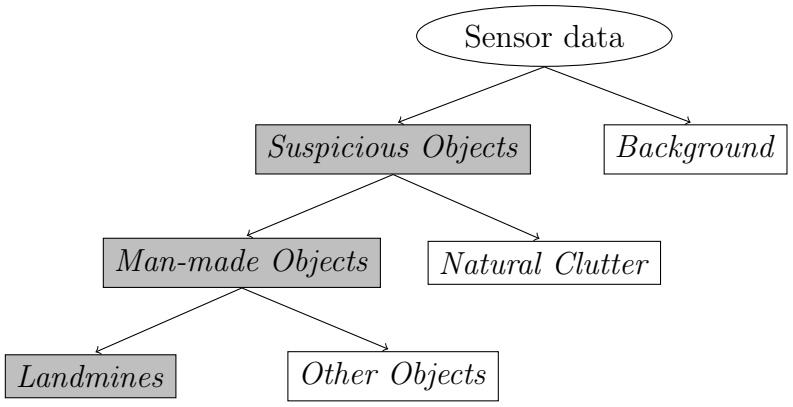


Figure 2.12: Three-step landmine detection (classes highlighted with gray background can be associated with landmines)

be terminated at any stage if the data quality is not enough to perform a more precise classification. Then, all objects of the appropriate class (shown with gray background in Figure 2.12) are considered to be *Landmines* allowing to perform the detection but with higher false alarm rate. Thus, the developed landmine detection strategy naturally appears as a step by step reduction of the false alarm rate.

2.3 Experimental Environment for Landmine Detection

To acquire experimental data for the development and testing of landmine detection algorithms significant resources are needed. This basically includes the creation of a test minefield with buried landmine dummies (usually, real landmines without explosives). Such effort supported by international programs was already made in order to provide model minefields for robots testing and to create databases of experimental data open for the public access. The data are usually acquired by a precise scanning device so they can be used for landmine recognition. For the purposes of this research (testing of the landmine detection algorithms) additionally to the testing results obtained by LADERO platform two databases which contain multisensor data were used.

2.3.1 Landmine signatures databases

Multi Sensor Mine Signature database

MSMS database is maintained by the EU Joint Research Center (JRC) in Ispra, Italy [5]. The database contains multi-sensor data recorded on 21 test fields of 7 different soil types (see Table 2.1), populated with mine surrogates and other objects. In this work the data from four sensors were used: Vallon ML 1620C pulsed metal detector, Foerster Minex

Soil type\Landmines	Small size	Medium size	Large size
Cluttered grassy terrain	A1	B1	C1
Loamy soil	A2	B2	C2
Sandy soil	A3	B3	C3
Pure sand	A4	B4	C4
Clay soil	A5	B5	C5
Soil with organic	A6	B6	C6
Ferromagnetic soil	A7	B7	C7

Table 2.1: MSMS test fields [5]

2FD 4.500 continuous metal detector, AGEMA 570 infrared camera and experimental ground penetrating radar C scans. Only data from 9 fields are available for all four sensors: A1, A2, A3, C1, C2, C3, C4, C5, C7 [5]. Thus, a combination of three sensors (without the GPR) was also used for experiments in order to maximally use all data sets obtained from 21 fields. Each field measures about 6m x 2m.

MACADAM

The MACADAM campaign was performed on the test fields of JRC by Thomson-CSF Missile Electronics [3]. The data are recorded for the following sensors: Mitsubishi IR 540 CD infrared camera, Ebinger EBEX 420 PB metal detector, Thomson-CSF TME SA X-band passive microwave radiometer, and ERA Technology SuperScan ground penetrating radar. The test fields are of three soil types: clean agricultural soil (12m x 6m), clean sandy soil (8m x 6m), and undisturbed local terrain (9m x 6m). Comparing to the MSMS database, the objects in these fields are located closer to each other providing a less realistic scenario (and a more challenging situation for the recognition algorithms).

2.3.2 Experimental test minefields

The experimental data obtained in well controlled conditions by a stationary scanning device as in the case of public databases are usually of a relatively good quality. This is important in the stage of development of the landmine detection algorithms to eliminate the influence of the scanning platform. However, to make a step towards the real situation it should be also confirmed that a mobile scanning platform is able to provide the data which can be used at least for some of the stages of the landmine detection. Such data will obviously have lower quality, but there is always a possibility for the improvement of the scanning mechanism with new technologies.

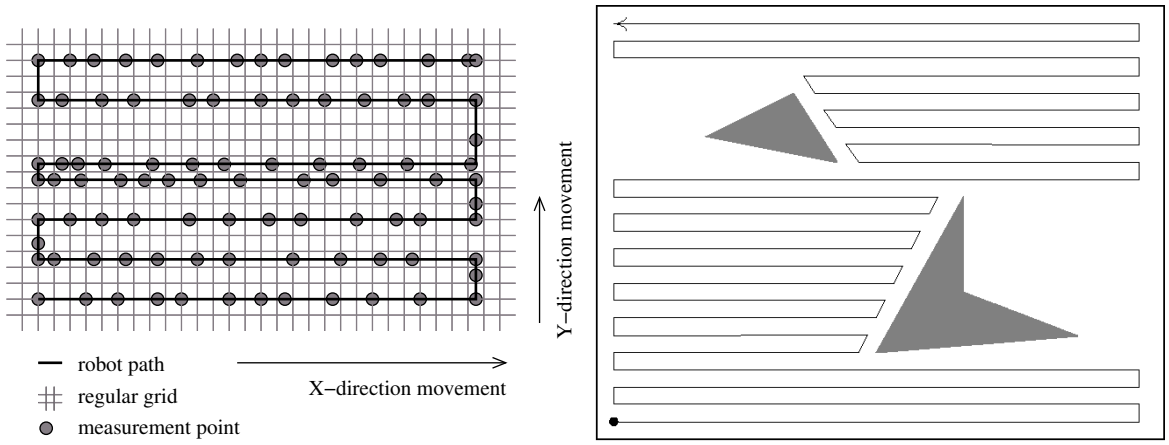
There are few test minefields available in Europe for the testing of the prototype

Figure 2.13: A part of the test minefield with mixed soil at Meerdaal bomb disposal unit

landmine detection platform according to its size and necessary equipment. A test field located at Meerdaal bomb disposal unit, Brussels, Belgium [87] was used for the experimental testing of LADERO platform . There are several fields with different soil types. However, only the one with mixed soil (see Figure 2.13 was used here because of easier accessability by the robot. The results of these experiments are presented in Chapter 7.

2.4 Approach for Path Planning

The landmine detection strategy described in the previous section defines the requirements for the path planning strategy of the robot. The main requirement is gathering of spatially mapped sensor data which can be used to acquire complex feature for landmine recognition. Assuming that the landmine detection sensors are fixed on the robot the robot itself has to execute a path which provides such data. The most convenient and widely used trajectory for this purpose is a back-and-forth path (also called, zigzag or boustrophedon pattern) when the robot moves along one coordinate forth and back



(a) A typical path of the robot

(b) An example of the area which cannot be covered by simple back-and-forth path

Figure 2.14: Basic back-and-forth motion of the robot

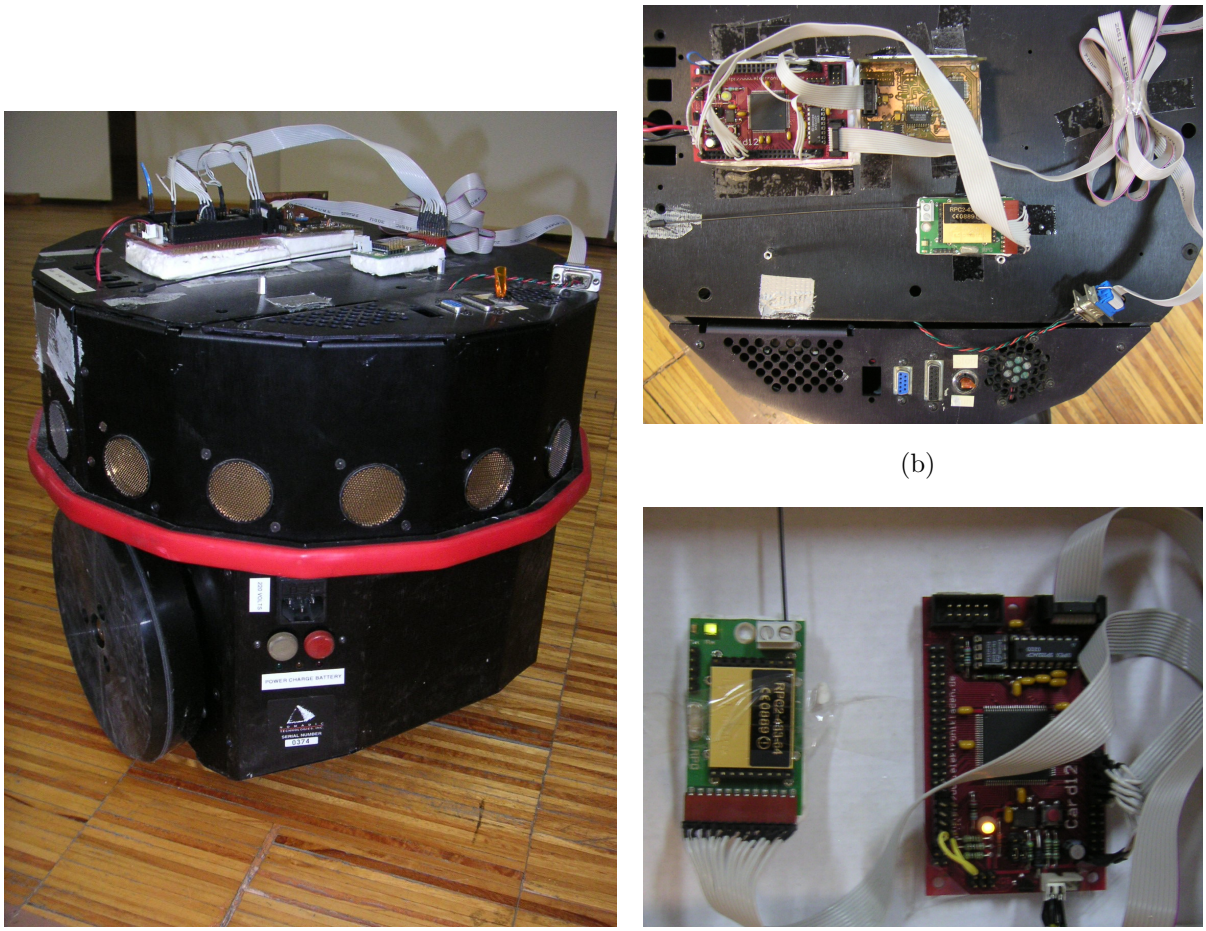
advancing for a small distance along the other coordinate. A scanning device is usually executing the same strategy. The prototype scanning platform is well suitable for such motion because it can move along straight lines in four directions. However, due to the low controllability of the pneumatic cylinders the coordinate grid which the robot can provide is not regular as shown in Figure 2.14(a). An algorithm for grid regularization required for further processing of the data is described in Chapter 3.

An accurate positioning of the platform is complicated both by internal and by external problems. Chapter 5 describes several of these problems and the solutions developed during this work including:

- Poor contact of the feet with the ground does not always allow a reliable motion. To improve this situation contact sensors were installed on the feet of the robot and a special algorithm for adjustment of the legs heights was developed.

- Two parallel cylinders of the robot are supposed to move with equal speeds to provide a reliable motion of the robot body. However, it is not practically possible to adjust their speeds with the required precision, which leads to the deformation of the robot and consequently to the distortion of its path. The developed algorithm provides an adjustment of the speed during the motion controlling the positions of the rods and switching the motion of the cylinders on and off.

Assuming these improvements the trajectory of the robot is still affected by unavoidable slippage and small rotations which provide localization errors if navigation is based only on the robot odometry. In order to solve this problem a controlling video system is installed as explained in Chapter 5. It uses a simple CCD camera and allows detection of relative displacement of the robot based on the virtual movements of natural landmarks found on the ground surface.



(a) (b) (c)

Figure 2.15: Nomad SuperScout Robot (a) and the control electronics for its connection to a PC: (b) - installed on the robot, (c) - installed on the PC.

The back-and-forth path executed by the robot is enough for scanning of an area if it does not contain any obstacles. Some minefields may be free of obstacles. However, it is considered in general that some obstacles could be present on the way of the robot. Then, the simple path cannot provide a complete coverage as shown, for example, in Figure 2.14(b). Moreover, the locations of the obstacles are usually not known, because of difficulties in the exact mapping of the minefield which cannot be entered. To assist the complete coverage of such area an online coverage algorithm was developed as described in Chapter 6.

2.4.1 Nomad SuperScout Robot

Testing of the path planning algorithms is usually complicated when using the prototype scanning platform due to its large size and low speed. Employing a smaller robot instead of the large one is possible in many cases if the algorithm does not rely on certain robot mechanics. In this work a Nomad SuperScout mobile robot was used for this purpose.

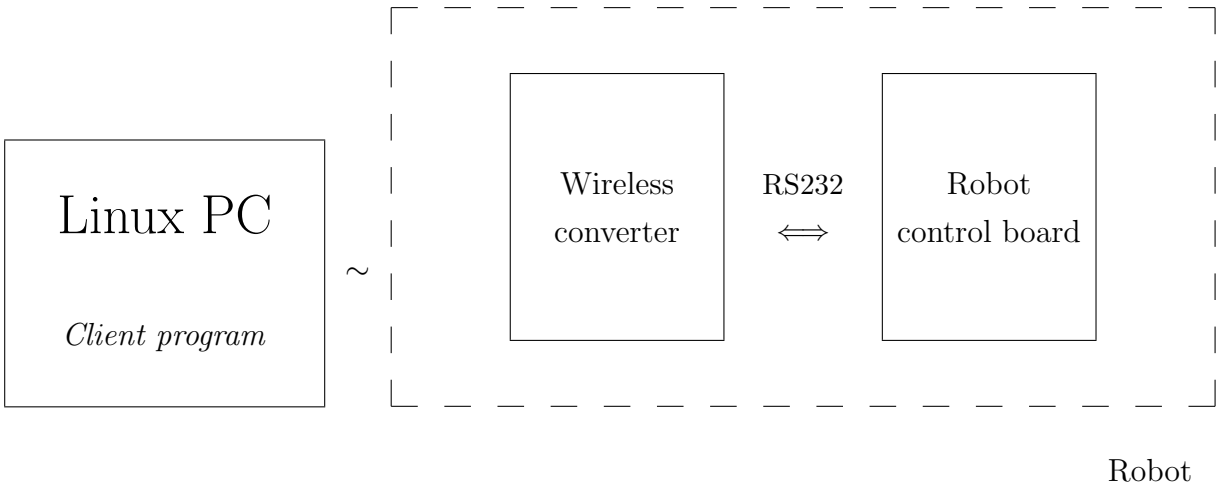


Figure 2.16: Structure of control hardware and software used for Nomad SuperScout mobile robot

This robot is based on a two-wheel differential drive and is equipped with 16 sonars for obstacle detection (see Figure 2.15(a)). The control of the robot is performed on a remote PC which is connected to the robot via wireless interface based on Radiometrix Radio Packet Controller (RPC) (Figure 2.16). The control board of the robot is connected by a serial link to the Card12 board which controls the RPC and simply transmits the data to/from the serial interface (Figure 2.15(b)). The same pair of Card12 and Radiometrix RPC is connected to the control computer (Figure 2.15(c)). Such interface allows a transparent wireless link, in which the control computer can behave in a way as if it was installed on the robot and directly connected to its control board. This design is especially convenient for the testing applications because it allows to save the battery power and eliminates the problems related to the installation of an onboard PC (hardware failures due to the movements of the robot and lack of power). A detailed description of the implemented hardware and software can be found in Appendix A. The results of the path planning experiments obtained with this robot are described in detail in Chapter 6.

2.5 High-level Control Software

As it was described in Section 2.1.3 the algorithms developed in this work are implemented in the *control program*. This software is written in C++ to run under Linux OS. The developed framework is not only dedicated to the implementation of the tasks of this research but also has some conceptual structures useful for robot programming in general.

There are several attempts to provide a common software framework for robotics

[88, 89, 90, 91, 92]. However, still there is no general framework which would support a diversity of robotic tasks. The existing projects in this area are usually focused on a certain type of mobile platform. This leads to dependence of the control algorithms on a particular mechanical structure of each robot. There are other frameworks which only provide an interface for communication between software objects and are convenient for large-scale projects which involve many developers.

The framework developed in this work does not claim to provide general software for robotics. However, the problems which were not addressed before are highlighted here and can be used in future developments of such a general mobile robot programming framework.

The structure of the *control program* developed here is object-oriented. A class represents an abstraction of a physical object or an algorithm. There is one *working loop* which contains all the fast processing required periodically, including reading of the robot status, taking decisions about the next actions and moving the robot. There is a separated thread for archiving and interface. Each software object may also have an additional thread for time-consuming processing.

This section describes the general principles of organization of the *control program*. The remarks about the implementation of certain algorithms are located together with their description in the next chapters.

2.5.1 Object Abstraction

The logical structure of the *control program* is determined by the goals of this work: it implements algorithms for path planning, navigation and sensor fusion. This variety of tasks can still be defined using a relatively simple tree-like structure of software objects:

- Robot
  - Sensor
    - Filter
    - Map
- Behavior
  - Decision maker

The *Robot* object defines the mechanical structure of the robot and its control hardware architecture. There is also an additional category *Robotconnection* which denotes the real interface between the program and the robot hardware, for example, it can be a real robot or a simulator. Object *Robot* contains several *Sensor* objects each of which provides an interface to the sensor data. *Sensor* obtains its current value automatically

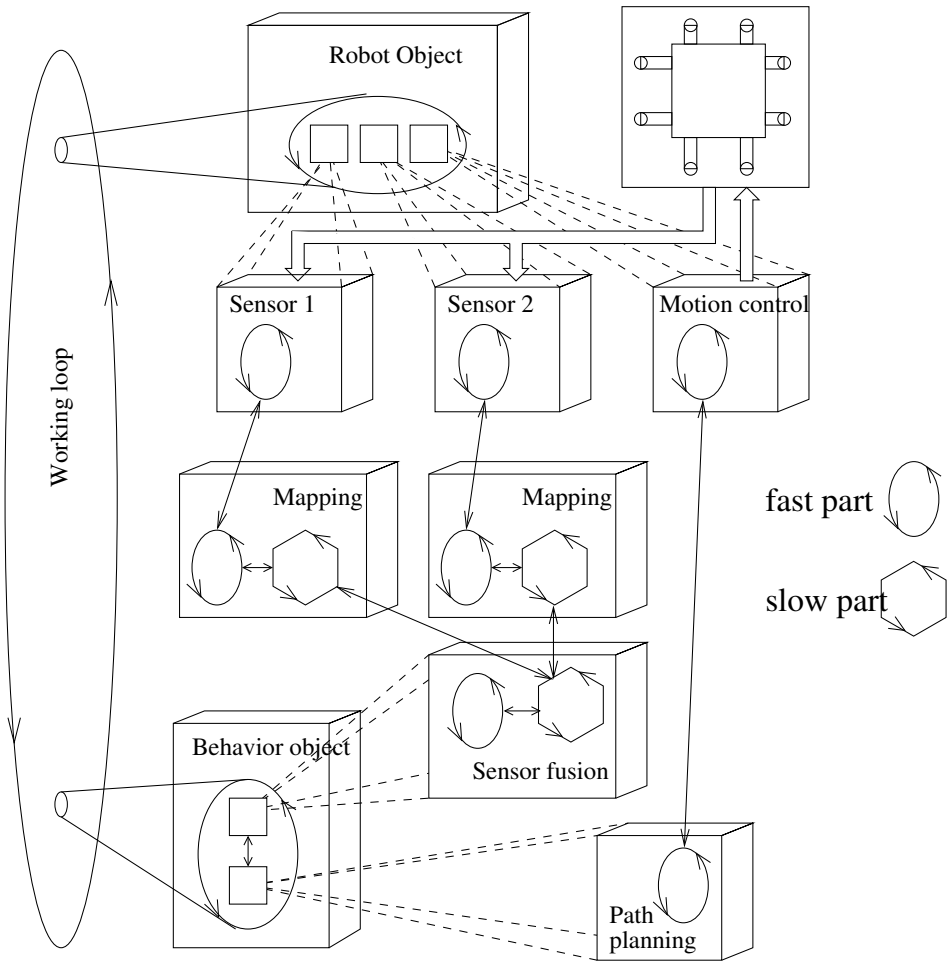


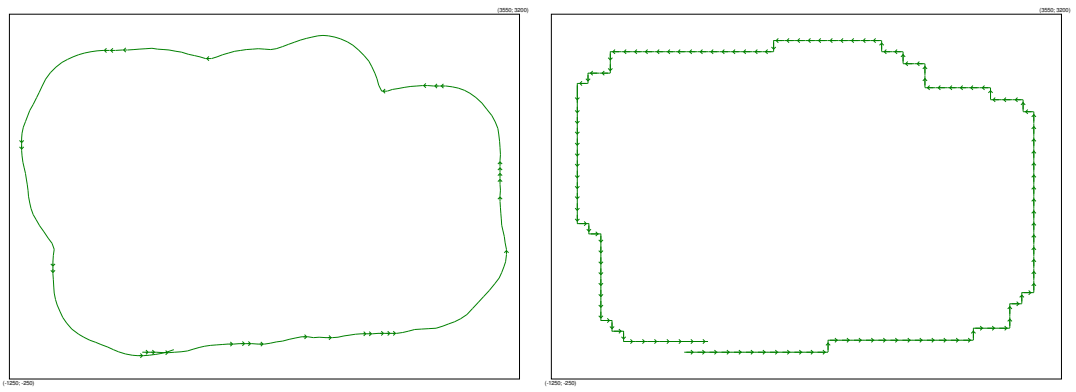
Figure 2.17: Structure of the *control program*

via the object *Robot* every cycle of the *working loop*. A *Sensor* may have two different types of conversions of its value:

- *Filter* performs a conversion without changing the type of the signal (for example, a Kalman filter provides an output value of the same type as the input value)
- *Map* converts the sensor value to a different form (for example, performing a spatial mapping)

The *Behavior* object provides an unified interface for any motion or behavior which the robot can execute. It defines a procedure for taking a decision about the next motion command to be sent to the robot. The algorithms for this processing are organized as objects *Decisionmaker*. A *Decisionmaker* is usually associated with several *Sensors* (and their *Filters* and *Maps*) to obtain the necessary data and provides a status which can be used by the *Behavior*. A recognition or sensor fusion algorithm is an example of the *Decisionmaker*.

Every software object has a possibility to perform fast processing in the *working loop*. However, this may be not enough for complex objects like pattern recognition algorithms



(a) Nomad Super Scout (b) Cartesian platfrom

Figure 2.18: Example of a wall-following motion controlled by the same algorithm for two different robots

which require long time for processing. The main control *working loop* cannot be frozen for such tasks because it controls the movements of the robot. For example, in the case of landmine detection, there is no specific *working loop* in which the landmine must be recognized, as long as it is far enough in front of the robot, and if the processing did not finish until the path-planning decision must be taken, the robot can be stopped. For this purpose each object can also implement its slow processing in a separated thread. To avoid multiply access of the same data from different threads an unified structure of such fast-slow object is implemented. It is based on standard possibilities of C++ including templates. A detailed explanation of this implementation can be found in Appendix B.

The structure of the *control program* described above is shown in Figure 2.17. This example shows a rough structure of the program for the demining robot: it contains a sensor fusion algorithm for landmine detection and an exploration behavior, and controls the robot.

2.5.2 Robot-Behavior Abstraction

The object abstraction allows to develop variety of sensor fusion, path planning and navigation algorithms without special attention on the robot mechanical structure. To support the robot abstraction the most challenging is to implement the behaviors of the robot because they have deal with the robot structure most closely.

The availability of two different robots for this research gave rise to an interesting problem: how to implement the abstraction between robot and behavior so that the path planning algorithm can be performed on both robots without changes and having a good efficiency. One solution can be to provide the simplest path which can be executed by all robots, but in this case more advanced robots are not used efficiently enough (for

example, the Nomad SuperScout can provide a smooth trajectory while the demining robot can only perform a path with rigid edges).

The approach implemented in this work is based on vector representation of the movement. The *Behavior* takes the decision about the desirable movement of the robot and forms the vector whose parameters define:

- starting point of the vector - desirable position of the robot;
- angle of slope - direction of the following trajectory;
- vector length - safe distance without obstacles in this direction;

Having this information each type of robot plans the actual motion. An example of how this strategy works is shown in Figure 2.18 where the same algorithm for wall-following is executed by both robots.

2.6 Summary

The approaches developed in the present work were outlined in this chapter in order to show a general picture of the system. The tools and the experimental environment which was needed for this research are also presented. Finally, it was necessary to show the structure of the software in which all the algorithms were implemented in order to be used onboard of the demining robot. The following chapters describe the newly developed algorithms in detail together with some details on their implementation inside this software framework.

Chapter 3

Suspicious Objects Detection

This chapter describes the algorithm developed in this work for the recognition of suspicious objects using landmine detection sensors. This process is a necessary first stage of landmine detection which provides preprocessing of the data to be used further by more advanced recognition techniques. The state of the at in this topic is described in Section 3.1 and the problem statement is formulated in Section 3.2. The novel approach developed for recognition of the suspicious object signatures for each sensor is presented in Section 3.3 and the approach for their association in order to form a complete object - appears in Section 3.3.6. This is followed by the experimental results in Section 3.5 and the description of the created landmine signature database in Section 3.6. Some details about the implementation of the algorithm are given in Section 3.4.

3.1 State of the art

The task of suspicious object detection was recently addressed in several works devoted to landmine detection. In most cases this processing is not concentrated solely on the detection of all possible objects against the background, but also attempts to include some recognition ability to reject the clutter, mixing several classification tasks in one. This process is also called *Region-Of-Interest* extraction by the name of a similar task in computer vision area. The *suspicious object* to be detected is a collection of Regions-Of-interest which represent signatures of the same real object obtained from the data of different sensors. The signature from each sensor is present in this collection only one time or can be missed in case if the sensor does not detect the object (as a metal detector will not detect a plastic object). The ROI is an area of spatially mapped sensor data which fully contain the signature of the object. It is characterized by its bounding rectangle. The task of this stage of landmine detection is the detection of ROIs and association of the ones representing the same object.

It is widely accepted that this stage is required for further landmine recognition

which includes feature extraction and classification. Similarly to other fields of pattern recognition in some works the ROIs are even selected manually [35] allowing to concentrate only on the recognition step. The common techniques for automatic ROI extraction are also adopted from computer vision by treating the spatially mapped sensor data as a gray image. These techniques include image segmentation and detection of certain shapes (circles, contours, etc.).

The simplest approach is a segmentation based on *thresholding*. It is used, for example, in [33] for segmentation of GPR data. A threshold value is applied to the image in order to create its binary representation where the objects (areas with abnormal intensity) are highlighted. Then a *region growing* algorithm is used to determine the connected pixels which finally form the ROI. To choose a good threshold value the whole image must be usually analyzed (for example, by analyzing its histogram). A too high threshold may lead to missing some objects, while a too low threshold may join several close objects together. Thus, the same threshold may not be used for data obtained in various environmental conditions when the background image intensity is typically different. In general, the threshold has to be constantly adjusted to the changing conditions. More advanced *texture segmentation* may also be used. However, without remarkable improvement in comparison to the usual thresholding [93]. It seems that the texture segmentation is not very applicable in this case because the data are poorly structured.

There is a group of approaches which use information about the shape of the object instead of its intensity. *Tophat filter* and *Hough transform* are used for the detection of circles [32]. In this case the radius of the circle has to be specified beforehand. Methods of *mathematical morphology* can be used as shown in [34]. For the application of these techniques the data have to be preprocessed with, for example, a Gaussian filter in order to reduce the noise.

A more general approach which does not rely on a certain shape of the objects is proposed in [58]. A watershed algorithm is applied there after a Laplace filter.

A common problem of these methods is the large number of parameters which have to be selected based on the data themselves (like, kernel size for a filter). This makes the tuning of the parameters complicated and restricts an online implementation. The existing algorithms also do not provide any means to determine if an object was fully scanned by the scanning device. In fact, the algorithm can be only started when the signature of the object is fully available, otherwise, it will detect only a part of the object. This fact provides one more limitation for an online implementation.

3.2 Problem Statement

There are no reliable algorithm of ROI extraction for landmine detection developed up to now. The following requirements should be accomplished by the ROIs detection algorithm:

- The algorithm should be able to use data of low-quality in terms of spatial resolution and noise. This requirement is important to allow the usage of the algorithms on the mobile scanning platform which is not able to provide good spatial resolution of the data.
- Signatures with different shapes should be detected. No particular shape, for example, circular, should be considered as a feature for the detection because the signatures are not limited to simple shapes. Even in the case of only circular signatures the data can be deformed due to the scanning errors, thus distorting the signature shape.
- The algorithms should provide all the processing online, which means that the signature of the object should be detected right after it is fully scanned by the scanning device. It should also not be detected only as a part until it is fully scanned.

An algorithm which satisfies the above requirement is needed and was developed in this work and is described in the next section.

3.3 Approach

The main idea behind the algorithm is a simple assumption that the ROIs can be found in places where heterogeneities in sensor readings occur. Such heterogeneities are analyzed using two iterative filters. This process is performed in 1D which helps to make the algorithm to perform online. The difference between the filters is mapped spatially and then the ROI is created using a technique based on region growing. All the process can be referred as a rule-based classifier because the region growing part of the algorithm is controlled by a set of rules. These rules, in fact, determine the model of the object to be detected and allow to obtain the required ability to detect the objects of different shapes.

All the stages of the developed approach are described in detail below.

3.3.1 Preprocessing of the data

It is assumed here that the sensor data are obtained by a scanning device. It can be a stationary scanning mechanism, as the one used, for example, in [5, 3], or a mobile

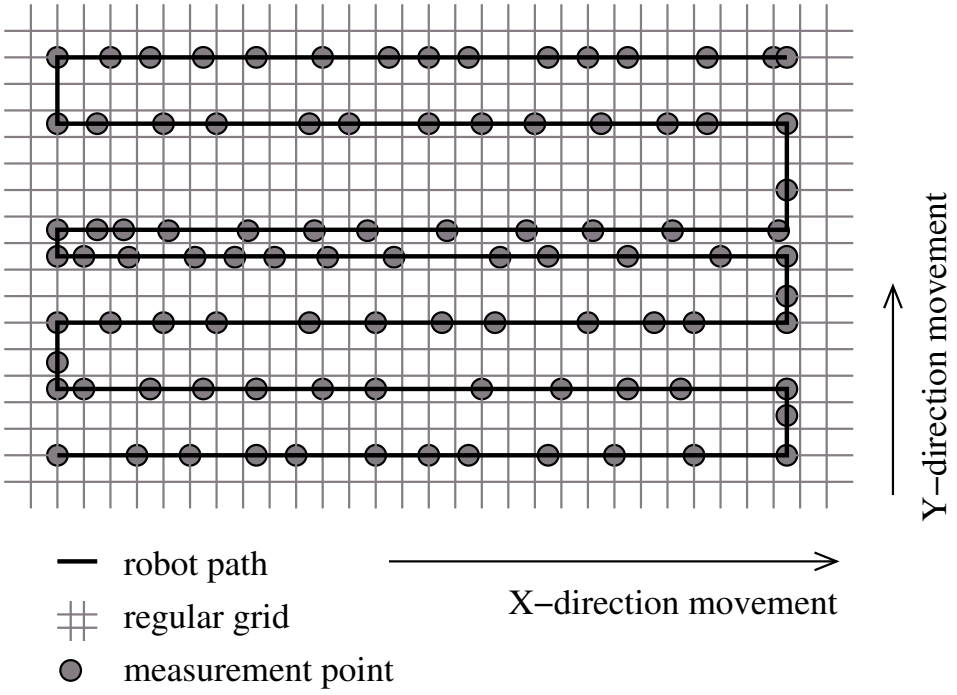


Figure 3.1: Example of a path followed by the robot

scanning platform as the prototype demining robot used in this work. The cartesian mechanical structure of the robot allows to scan the terrain while performing the movements of the robot body. To obtain the map (image) of the sensor data the robot performs back-and-forth motions covering the defined area. The robot performs the scanning of the area alternating between the two basic movements which were described in Section 2.1.1: *scanning step* along X axis and *advancing step* along Y axis.

In an ideal case this set of motions would form a regular grid of sensor readings, but in reality the situation is complicated by using discrete pneumatic cylinders which leads to a not completely regular data grid as shown in Figure 3.1. Therefore, regularization of the grid is needed in order to provide the data suitable for further processing.

Grid regularization is performed using a simple algorithm based on 1D linear interpolation. The mapping process is performed as follow:

1. During X-direction movement each two real points form a set of temporary points with the same Y coordinates. X coordinates of the temporary points fit to the desired grid knots whose values are calculated using linear interpolation (Figure 3.2a and b).

2. Two temporary points with the same X coordinate obtained in step 1 form the values of the grid knots situated between them by linear interpolation as well(Figure 3.2c).

In case of a stationary scanning device the same algorithm is used for the grid reshaping if necessary. This simple algorithm does not require significant processing

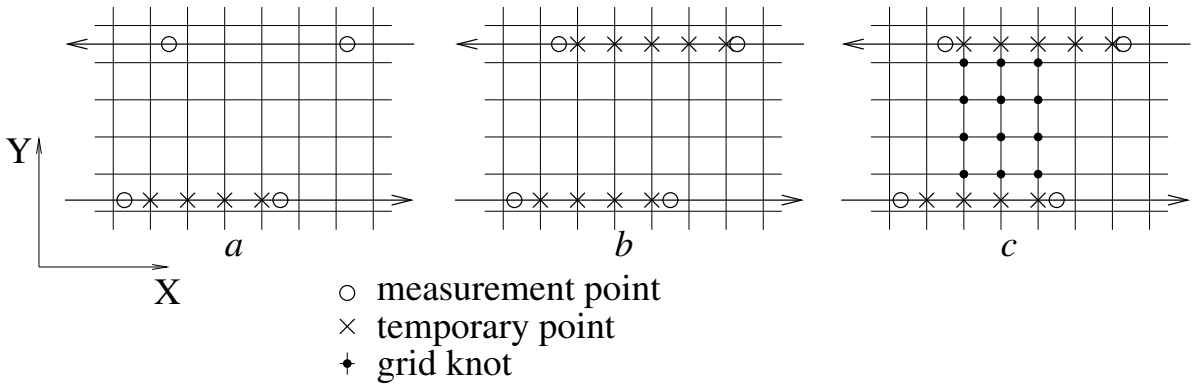


Figure 3.2: Mapping the sensor values into a grid-map

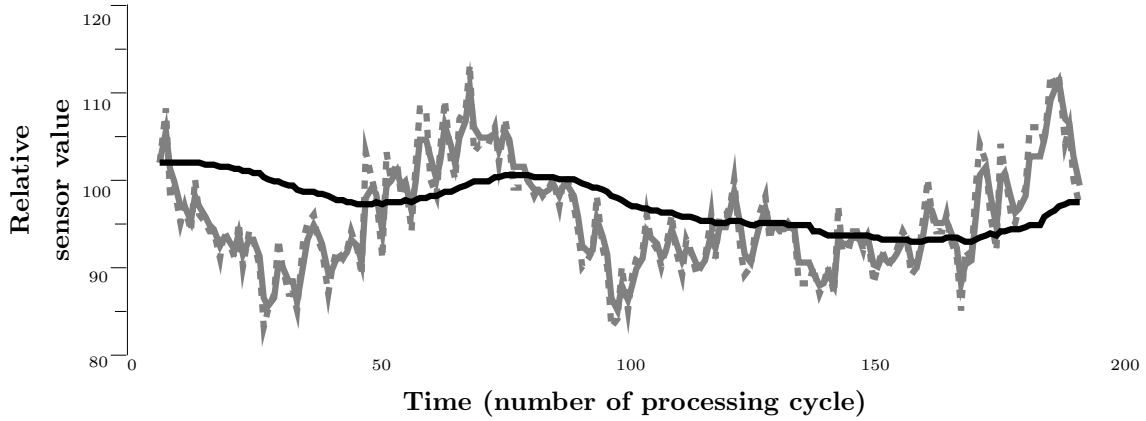


Figure 3.3: Detecting signal heterogeneities: the fast filter (gray line) follows the signal (gray dotted line), and the slow filter (black line) follows its average.

time, thus a grid-map for each sensor can always be updated online.

3.3.2 Detecting signal heterogeneities

While the sensor signal is acquired it is processed to detect heterogeneities which may signify the presence of a suspicious object. This is a 1D analysis without considering the spatial location of the current sensor value. An even when the heterogeneity occurs is called an *interesting point*.

A heterogeneity in sensor readings can be detected by analyzing changes of its value. It is proposed to use for this purpose two iterative filters based on the principles and equations of the Kalman filter with different parameters.

The process model is assumed to be: $x_k = ax_{k-1} + w_{k-1}$ where $a = 1$. This means that the initial signal is considered to be constant with the process noise w_k. The process noise covariance Q represents a possible signal deviation from a constant value. Q is

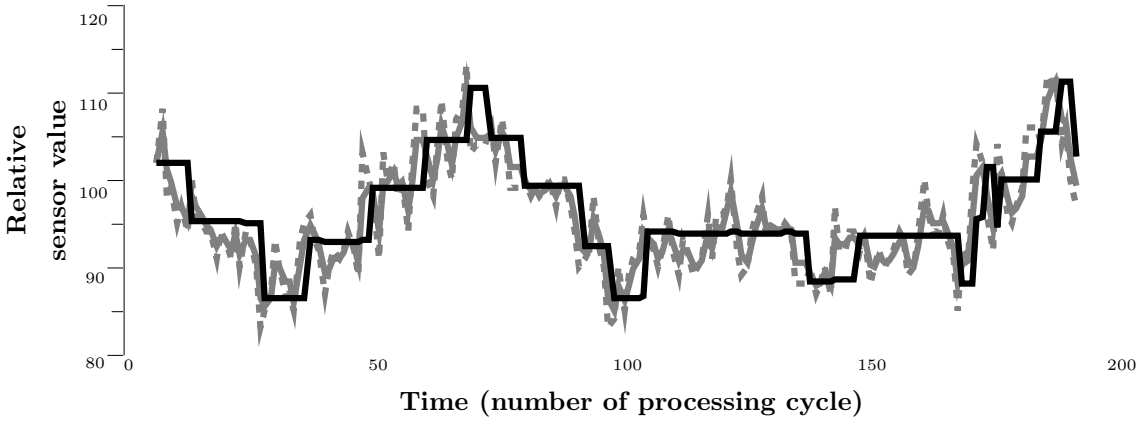


Figure 3.4: Detecting signal heterogeneities: the slow filter is pulled up (down) to the fast one at the *interesting points*

considered to be constant in time. Selecting different values for Q it is possible to obtain filters with different behaviors: the larger Q, the more exact and faster a filter follows the signal. Figure 3.3 shows a sample signal and two filters with $Q = 40$ and $Q = 0.004$. The measurement noise covariance R for both filters has the same value. It can be noticed that the *slow filter* provides an average level of the signal while the fast filter tracks it more closely. An *interesting point* is detected if the difference between the *slow filter* and the *fast filter* exceeds a threshold. This threshold takes the same value as Q used in the *fast filter*: this value is considered to be significant to form an *interesting point*.

Every time an *interesting point* is discovered the *slow filter* is "pulled up" to the *fast filter* by changing its system parameter a. This procedure prevents detecting the same *interesting point* several times and allows to track the dynamics of the filters differences which will be used for the next steps. This process is shown on Figure 3.4.

3.3.3 Segmented Map

Segmented Map (SM) is a grid-map which represents *interesting points* spatially. It is formed as follows:

1. An initial integer *current value* is chosen (it can be any number, for example 100)
2. If $Value_{fastfilter} - Value_{slowfilter} > threshold$ then *current value* is incremented

   If $Value_{slowfilter} - Value_{fastfilter} > threshold$ then *current value* is decremented
3. *Current value* is mapped to the *segmented map* according to the *interesting point* coordinates

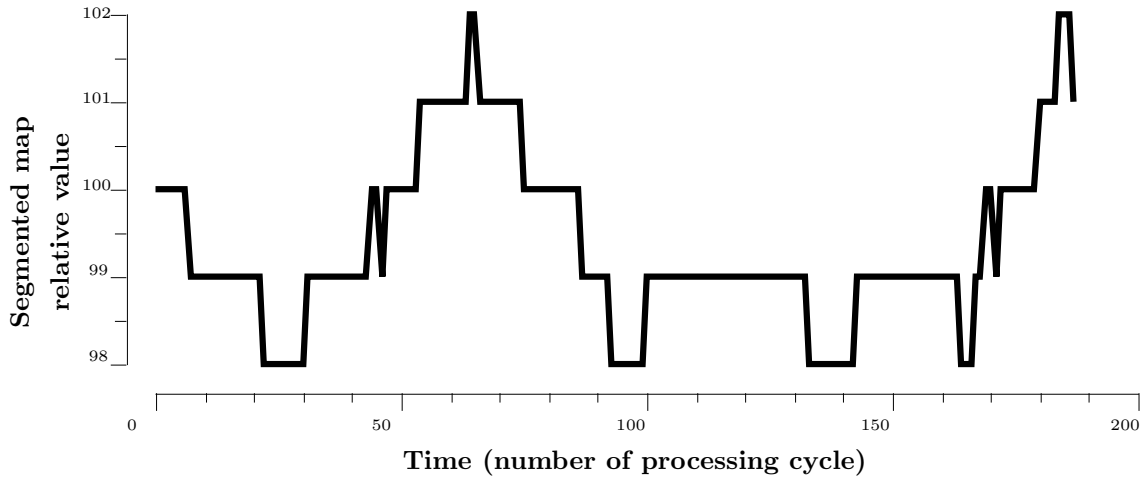
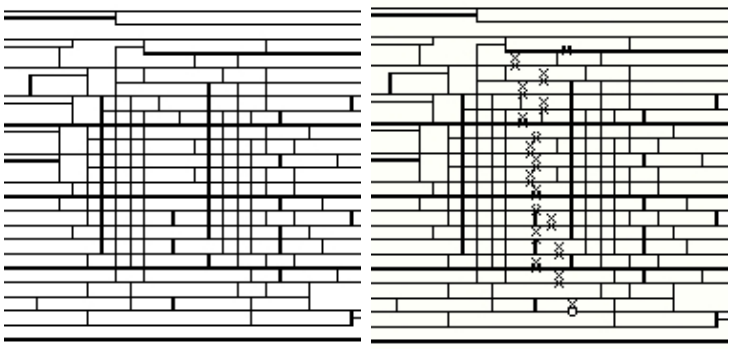


Figure 3.5: Value of the Segmented Map is increased and decreased by an integer value depending on the difference between the filters



(a) (b)

Figure 3.6: An example of *Segmented map* (a) and extrema searching in it (b)

This process forms a map with homogeneous segments that differ from each other by an integer value (the values inside segments are not important for the rest of the algorithm, whereas the difference between them allows to distinguish the segments from each other). Figure 3.6*a* shows a *segmented map* obtained from the sample data.

An important property of the *segmented map* is: the center of the ROI along X coordinate corresponds to the local extremum on the *segmented map* along X coordinate. A maximum/minimum searching is performed to detect all local extrema for each value of Y coordinate. The nature of the sensor determines which type of extremum should be used: for a metal detector only maxima are appropriate, while for an IR sensor both maxima and minima are considered. The result of a maximum searching is presented in Figure 3.6*b*.

3.3.4 Region growing

Each extremum encountered on the previous step starts a region growing process as follows:

1. Selecting a *segment* to which the extremum point belongs - *start segment*. The *start segment* forms an initial *Object Area* - a grid map containing all the segments which belong to the detected object

2. Detecting all *segments* adjacent to the *Object Area*

3. Any detected *segment* is tested according to the rules presented below. If the *segment* passes the test it is joined to the *Object Area*

4. If there are no new *segments* joined the region growing is finished, otherwise repeat from step 2.

In step 3 each segment is tested to satisfy the following requirements:

- 95% of the path between the current *segment* and the *start segment* must be located inside the *Object Area*

- The aspect ratio of the *Object Area* should not be increased by the addition of the *segment*

- The extremum value inside the *segment* must not exceed the one in the *start segment* (this ensures that the region growing is started from the middle of the object)

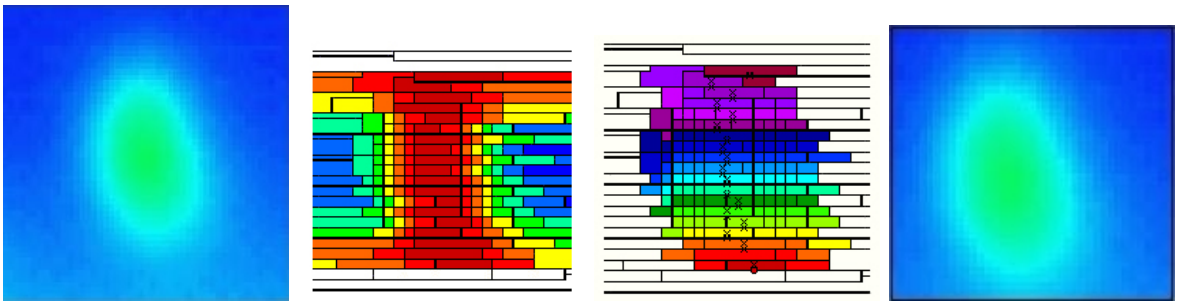
The obtained *Object Area (OA)* is tested again in order to reject the object whose aspect ration exceeds 3. If the object is rejected the extremum which started it is marked in order not to initiate a new region growing. This algorithm is applied to all encountered extrema.

3.3.5 Final ROI representation

The final ROI is a rectangle which contains the obtained *Object Area*. This rectangle is brought in correspondence with the initial grid-map forming a *Data Map (DM)*.

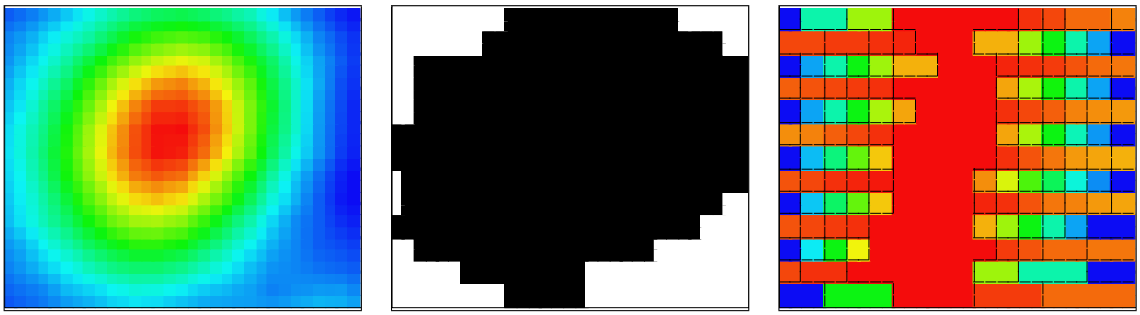
The final ROI consists of three grid-based maps shown in Figure 3.8: *Data Map (DM)*, *Object Area (OA)* and *Segmented Map (SM)*. The two additional maps increase the amount of information about the object:

1. The *Object Area* contains a rough geometrical shape of the object. Thus, by multiplying the *Object Area* with other maps, the background around the object can be fully eliminated.



(a) (b) (c) (d)

Figure 3.7: Stages of ROIs extraction process: (a) - object to be detected (a small part of the sensor data map), (b) - *Segmented Map*, (c) - result of region growing in the *Segmented map*, (d) - *Data Map* of the obtained ROI



(a) Raw Map (b) Object Area (c) Segmented Map

Figure 3.8: Example of ROI maps (high-metal landmine seen by pulsed metal detector): (*a*) *Data Map*, (*b*) *Object Area* and (*c*) *Segmented map).*

2. The *Segmented Map* presents a more unified fingerprint of the object. Thus, some features show better performance when calculated using the *Segmented Map* than with the *Data Map* (see Chapter 4).

3.3.6 Object Association

After completing the ROIs detection step, the ROIs from different sensors that represent the same object are associated together in order to form a complete *suspicious object* which will be further processed by the landmine recognition algorithms. This association is performed if the distance between them is less than a certain threshold (e.g. 100 mm) and one ROI can be associated with different objects.

3.4 Implementation

The described algorithms are implemented as a number of C++ classes embedded in the software framework developed in this work. For each sensor the objects of the following classes are used:

- Detecting signal heterogeneities is implemented by a *decision maker* `CTimeROIDetector` which uses two Kalman filters as the sources and runs in the common working loop
- Mapping the sensor data (regularization of the grid) is implemented in classes `CPlaneGridMap` (for *Data Map*) and `CSegmGridMap` (for *Segmented Map*). They include fast parts which only acquire the data from the sensors and store them in the buffer. The slow parts perform the actual mapping using the stored data.
- ROI detection is executed by a *decision maker* `CGridROIExtractor` which also performs the processing in a separated thread of its slow part.
- Object association is carried out by a *decision maker* `CObjectAssociation`

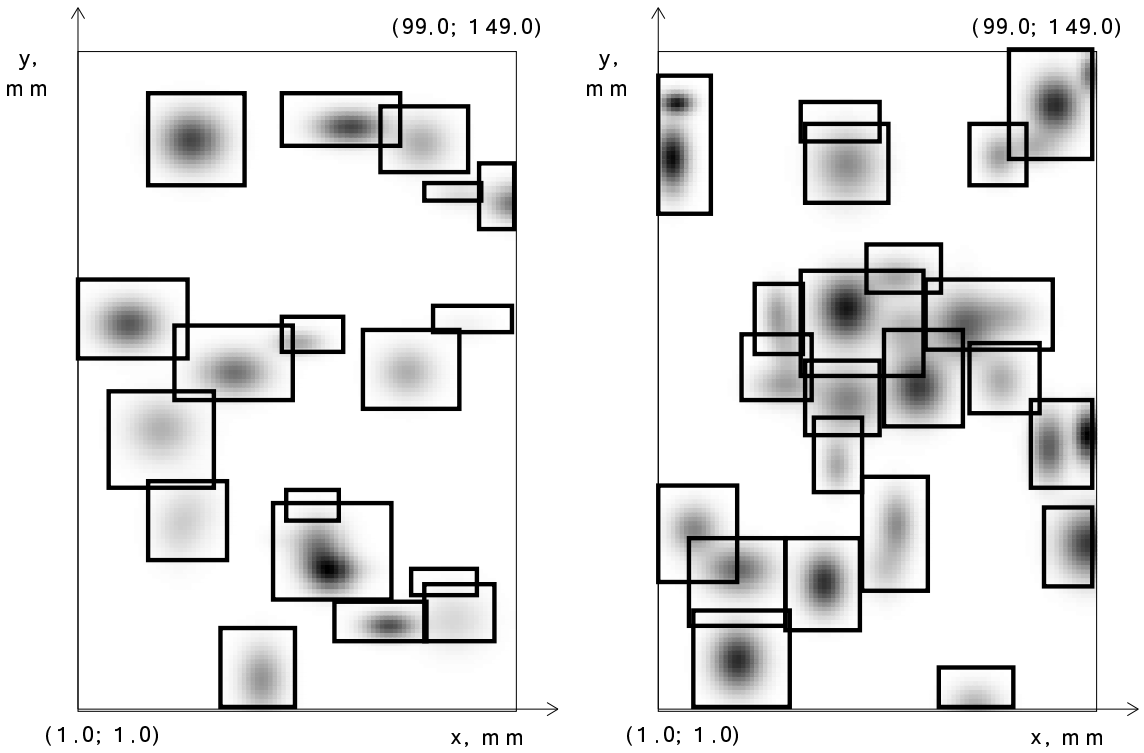
Please refer to Appendix D for more details.

3.5 Experimental Results

The developed ROI detection algorithm was extensively tested on simulated and real experimental data. The performance requirements for it should be very strict: if an object is not detected by the ROIs extraction, it will not be considered for the classification, and thus an eventual landmine can be missed. On the other hand, if a part of the background is selected as an *Object*, the performance of the classifier will be degraded.

Sensor data which has to be processed by the ROI detection algorithm allows to test it on several cases of object shapes and experimental conditions. Before providing these results the general properties of the algorithm can be tested on simulated sensor data which allows more detailed analysis. The simulated data were generated by placing a number of objects simulated as 2D Gaussian distributions inside a test area. The locations and parameters of the objects (mean and deviation) are generated randomly from predefined ranges. This technique results in test data with different density, shape and contrast of the objects as it can be seen in Figure 3.9.

Figure 3.9 shows the results of ROI detection on simulated data in cases of different density of objects. The case shown in Figure 3.9(a) depicts one of the simplest sensor data because the objects are well separated from each other. However, it can be seen that the performance of the algorithm is not degraded in the case of higher density of objects, as shown in Figure 3.9(b). The intersections of the ROIs detected on this data



(a) (b)

Figure 3.9: Example of ROI detection on simulated data with different density of objects

do not mean that the algorithm specifies the same part of the area to belong to different ROIs. In most cases only the bounding rectangles of ROIs are intersecting, and the *Object Areas* of the detected objects do not intersect and allow discrimination of the objects.

ROI detection should provide good results on objects with different shapes and in different experimental conditions. It should be noticed that a number of failures in the data acquisition system may occur but the object still has to be detected. In a case of failure the signature of the object will be deformed which will probably not allow to classify it with high accuracy. However, the object should be detected by the ROI detection algorithm to be considered at all. A number of cases with different parameters of data acquisition system were simulated in order to test the performance of the algorithm. As a reference of the algorithm performance, the result of ROI detection in case of nondeformed object is considered as shown in Figure 3.10(a). Figure 3.10(b) presents a case when background conditions change (due to a sensor failure or a real change in environmental conditions like changing of temperature in case of IR sensor). The failure occurs in the middle of the object, but, the result of ROI detection is unchanged. This is possible due to the fact that the algorithm relies on the relative changes of sensor values (recorded in the *Segmented Map*) and not on the absolute sensor values. Another source of possible failure is the positioning of the scanning

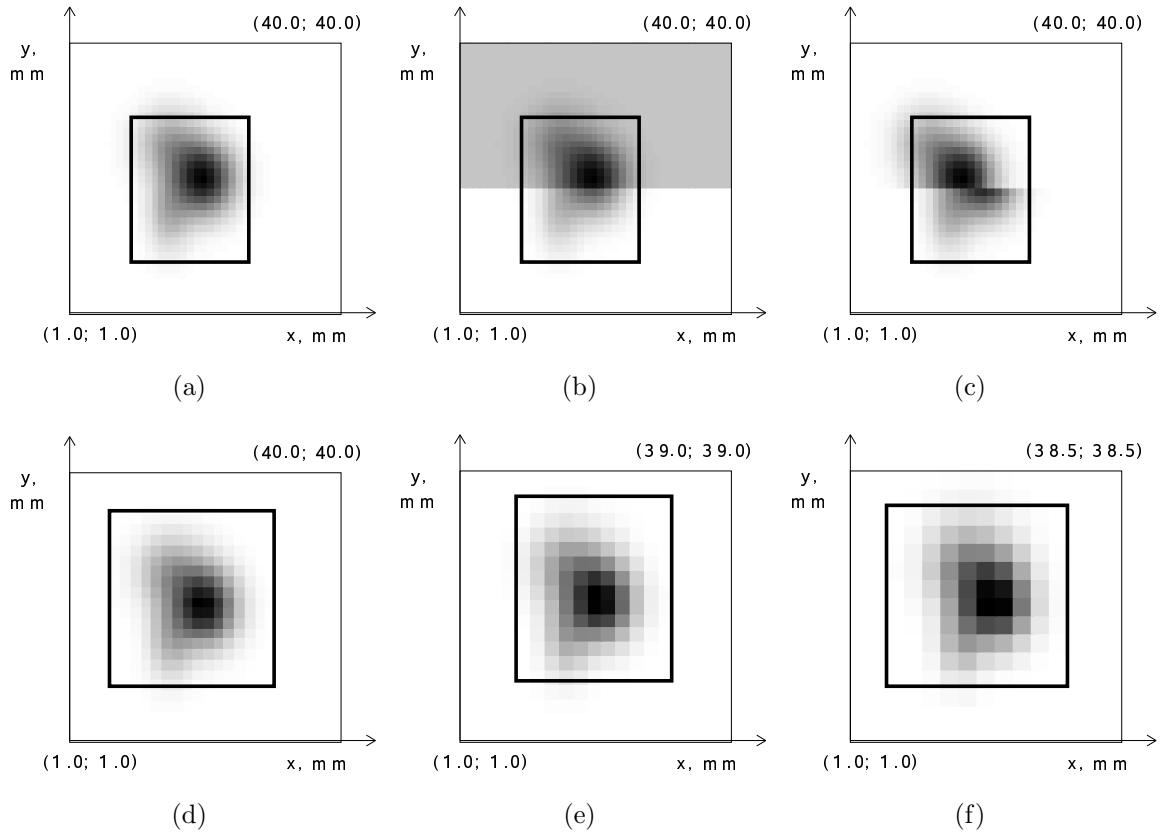


Figure 3.10: Example of ROI detection for different parameters and failures of the data acquisition system: (a) - original object for reference (using data with grid size $1 \times 1mm$), (b) - changing of background conditions, (c) - the shape of the signature is deformed due to the positioning system failure, (d) - using data with grid size $1.5 \times 1.5mm$, (e) - using data with grid size $2 \times 2mm$, (f) - using data with grid size $2.5 \times 2.5mm$

device which can result in a deformation of the object shape as shown in Figure 3.10(c). In this case, the middle part of the object is still detected by the algorithm. If a larger deformation occurs, the result will most probably consist in two separated objects because the algorithm will not be able to recognize the connection between the two parts.

Different scanning systems provide sensor data with different quality, which first of all consists in the size of data grid, the larger the size the worse the quality of data. To examine the influence of this parameter on the performance of ROI detection three cases with different grid size are considered in Figures 3.10(d) - 3.10(f). It can be seen that the object is detected in all case. However, comparing with the original object in Figure 3.10(a) (with the smallest grid size) a larger part of the area is detected as ROI. The small dependency of the performance on the data quality is essential because the algorithm is intended to be used on the mobile scanning platform.

The algorithm allows the operation in several modes depending on the contrast of suspected objects: positive or negative contrast, and the manner how to handle the objects with different contrasts. The available modes include:

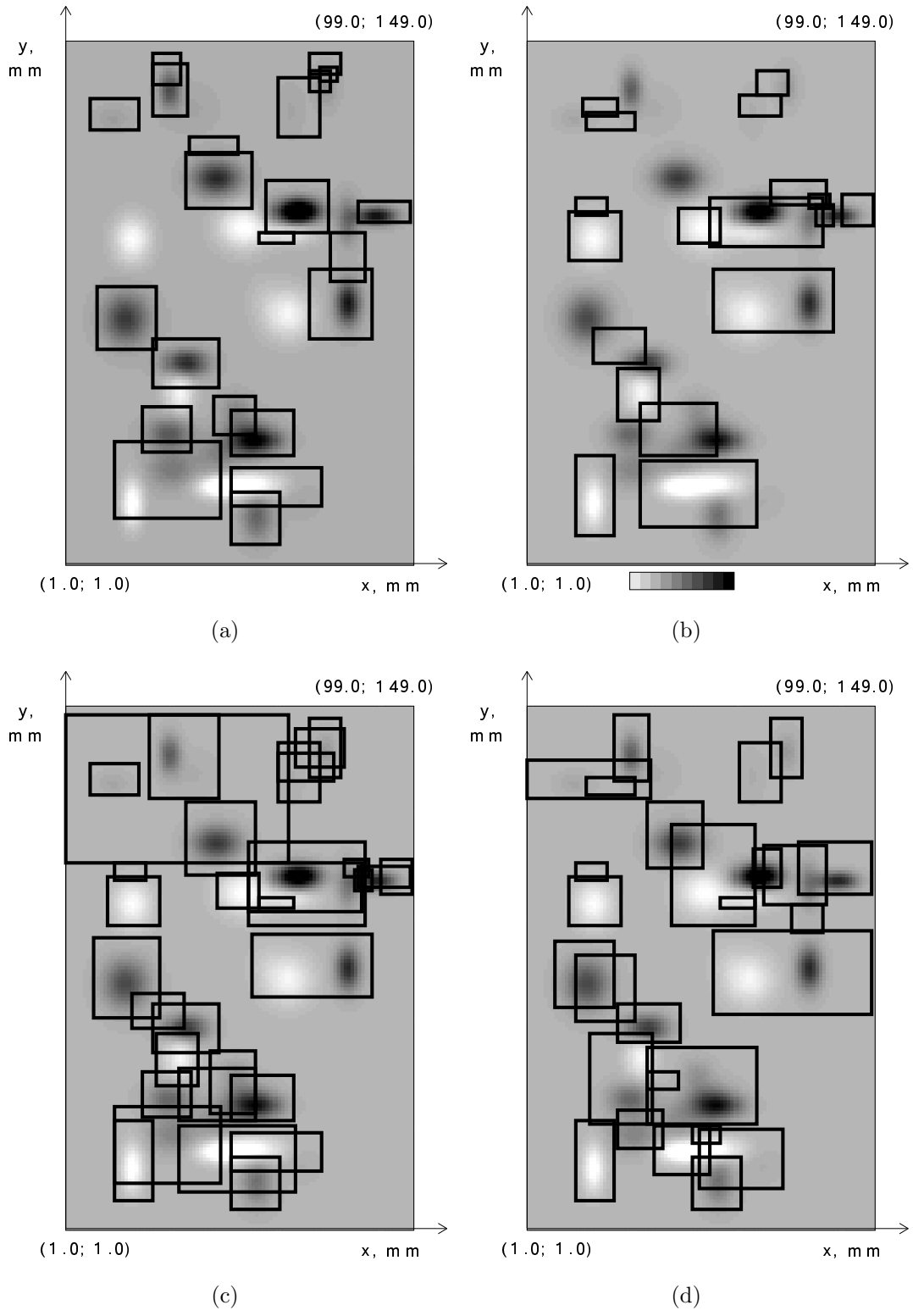


Figure 3.11: ROI detection results for different modes of the algorithm (simulated data, positive contrast is shown with darker areas, negative contrast is shown with lighter areas): (a) - detecting only objects with positive contrast (mode 1), (b) - detecting only objects with negative contrast (mode 1), (c) - detecting objects with positive and negative contrast separately (mode 2), (d) - detecting objects with any contrast ignoring the difference between positive and negative contrast (mode 3)

1. detecting objects with only one type of contrast (positive or negative)
2. detecting objects with any contrast separately
3. detecting objects with any contrast considering them as one type of contrast

The mode of operation is controlled by the type of extrema which is analyzed during extrema searching inside the *Segmented Map* (see Section 3.3.3. Figure 3.11 shows the results of ROI detection performed on the same simulated data with different modes of operation. When only one type of contrast is considered (Figure 3.11(a) and 3.11(b)) the objects of the other contrast are mostly ignored (however, there are some cases of joining a low-contrast object to a high-contrast object of different contrast). It should be noticed that the considered data are confusing when the algorithm is operated in this mode because it assumes that the objects are surrounded by the background and not by the other objects. Figure 3.11(c) shows the result of operation in the second mode. Now all objects are detected separately with one failure when two objects with different contrast are joined. This mode of operation is the most challenging but important for several sensors, like IR sensors and GPR, where the type of contrast the objects have is usually not determined. For example, for IR sensors the contrast of objects depends on the time of the day and on the environmental conditions). The last mode of operation allows to ignore the difference between the types of contrast and to detect the objects consisting of the parts with different contrast. It can be seen in Figure 3.11(d) that closely located objects with different contrast are joined together and detected as one object. This mode is used in case of continuous metal detector as shown below.

The developed algorithms were tested using experimental data from MsMs database [5]. An example of ROI detection performed on the data from four sensors can be seen in Figure 3.12. The sensor data were processed in different modes: pulsed metal detector and ground penetrating radar - in mode 1 considering only objects with positive contrast, continuous metal detector - mode 3, infrared camera - in mode 2. It is well seen that the data provided by the infrared camera and ground penetrating radar have a lot of clutter objects which provide the main source of false alarms.

To estimate the performance of the ROI detection algorithm, the experimental data were labeled to create a sensor-specific ground-truth map, which contains all the objects seen by a human on the sensor data with adjustable scales. Only data from pulsed MD and continuous MD were analyzed, because the other sensors are difficult to process manually. The detection rate (DR) was estimated as the number of ground-truth objects detected by at least one ROI divided by their total amount. The false alarm rate (FAR) was calculated as the area occupied by false detections divided by the total area of background. A performance of DR = 90% and FAR = 4% was achieved.

From the 90% of the detected objects, 47% were detected more than once. This large number of repeated detections increases the processing time but, on the other

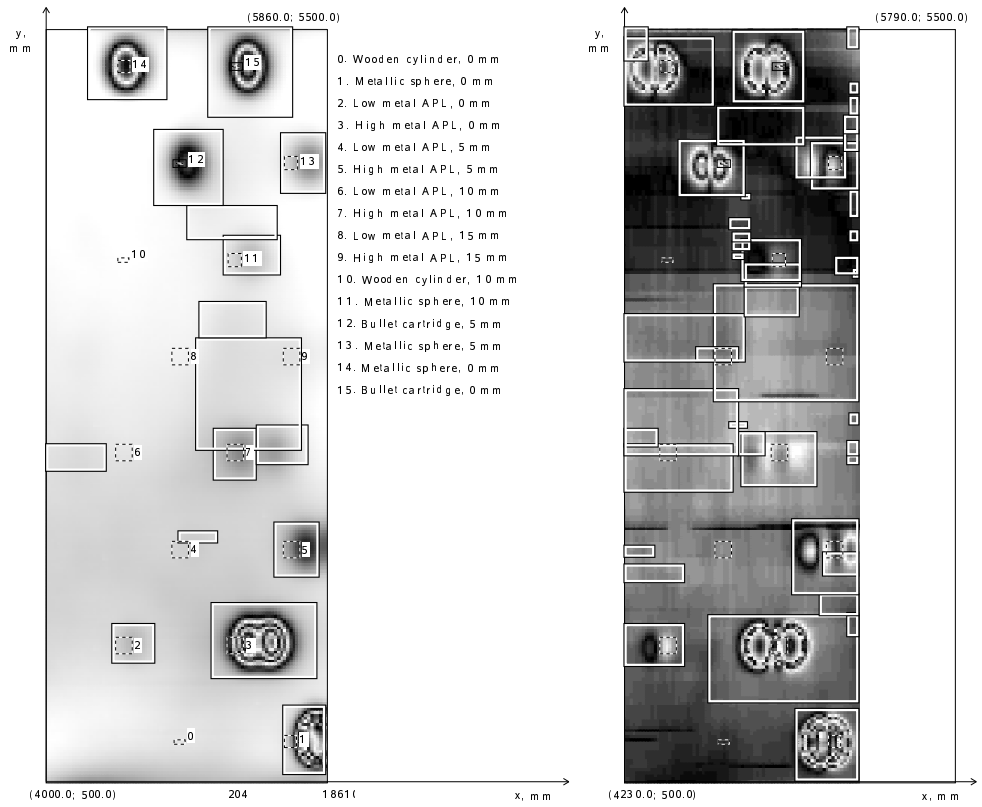


Figure 3.12: Example of ROIs extraction for the data from C7 MSMS field [5] (a) - pulsed metal detector, (b) - continuous metal detector, (c) - infrared camera, (d) - ground penetrating radar; the objects are marked with dashed rectangles and numbers according to the legend shown on (a)

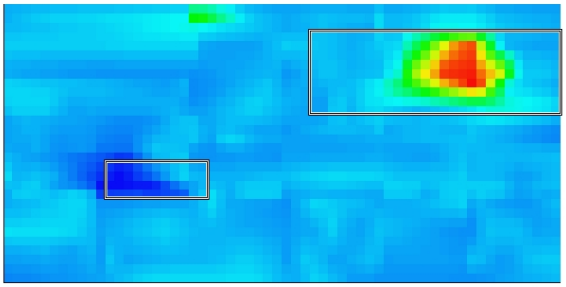


Figure 3.13: Results of ROI detection for data obtained by the mobile scanning platform LADERO from an IR sensor

hand, most of such objects have weak (or confusing) fingerprints. Therefore, in the case of a landmine several detections may improve the probability to detect it by at least one of the ROIs.

Sensor data obtained by the mobile scanning platform LADERO has much lower spatial quality than the experimental data obtained by a precise scanning device in [5]. Next examples show the performance of the ROI detection algorithm on such data. Figure 3.13 depicts the data obtained from one of the IR sensors in a test area with two objects of different contrast (hot and cold). The algorithm is operated in mode 2 so both objects are detected. Examples of data obtained by the metal detector from real landmines located on the test field are shown in Figure 3.14. In both cases the ROI detection algorithm allows to detect all the objects even having data with the low spatial quality. However, some of the objects are deformed due to the positioning failures of the platform and detected as several objects.

The detection of suspicious objects may appear to be the first and the last step of landmine detection if the quality of sensor data does not allow further processing. Then, according to the strategy descried in Chapter 2 all the suspicious objects should be considered to be landmines as shown in Figure 3.15. In order to estimate the performance of landmine detection in this case the data from [5] for four sensors obtained from 9 fields was analyzed (the considered sensors were: Vallon ML 1620C pulsed metal detector, Foerster Minex 2FD 4.500 continuous metal detector, AGEMA 570 infrared camera and experimental ground penetrating radar C scans). By treating the entire detected object as landmines for a 95% detection rate a value of 12.1 FA/m^2 was obtained.

3.6 Object Signatures Database

The presented algorithms allow transformation of the raw sensor data into a unified form when only the objects are considered. This can be used not only for the online landmine detection but also as a convenient preprocessing for further research on more

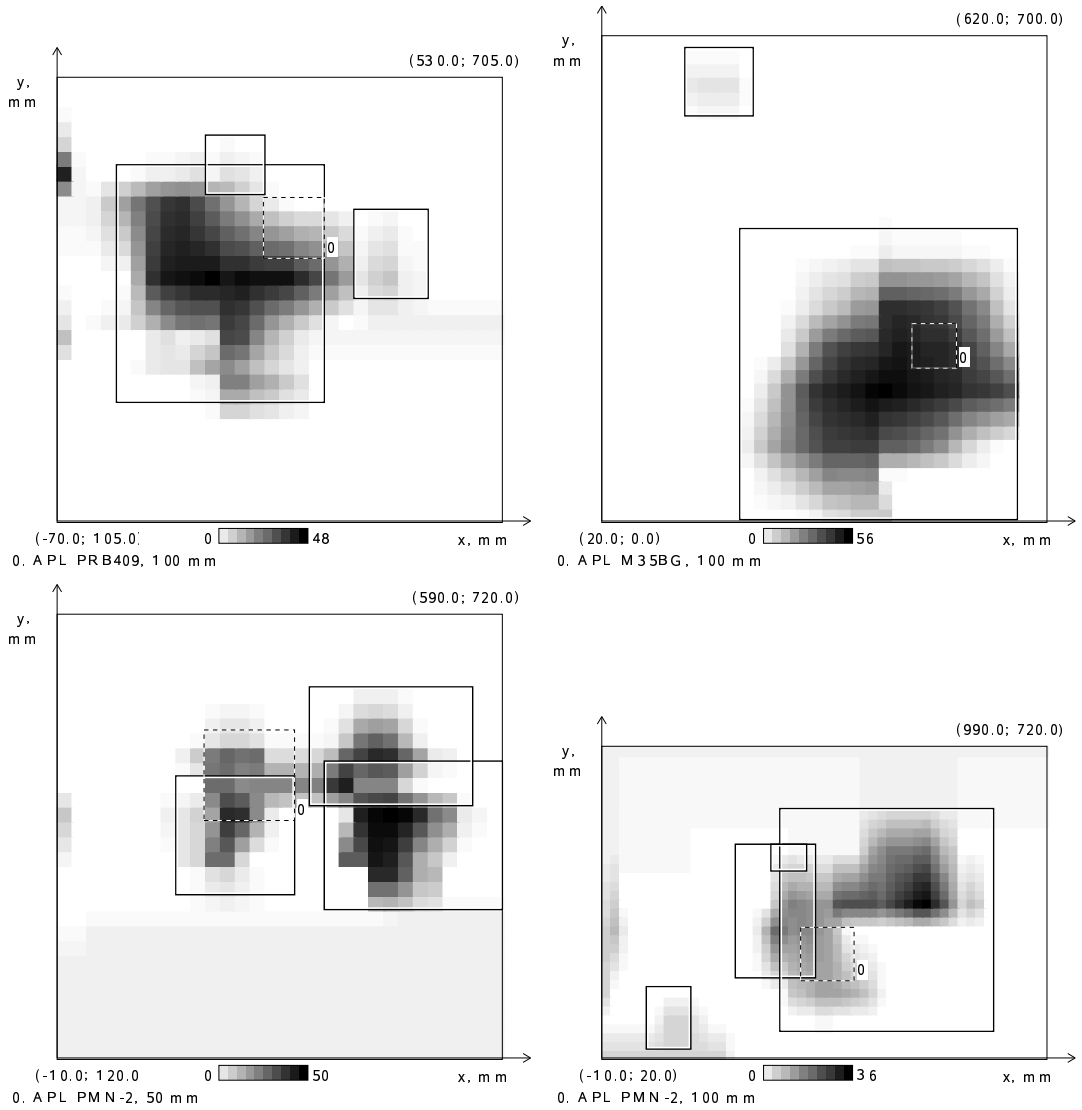


Figure 3.14: Results of ROI detection for data obtained by the mobile scanning platform LADERO on the test minefield; the objects are marked with dashed rectangles and numbers according to the legend shown on each image

sophisticated recognition technologies. For this purposes a database of suspicious object signatures was created in this work. It stores the objects as collections of signatures obtained from different sensors associated together. Each signature is associated with the experiment where the data was taken, includes coordinate information, and is composed by three maps: *Data Map*, *Segmented Map*, and *Object Area*. Thanks to this database most of the analysis performed for the development of the landmine recognition techniques described in the next chapter was performed offline.

The database is implemented in MySQL and has interfaces in C++ and PHP for data management and visualization (see Appendix E for details).

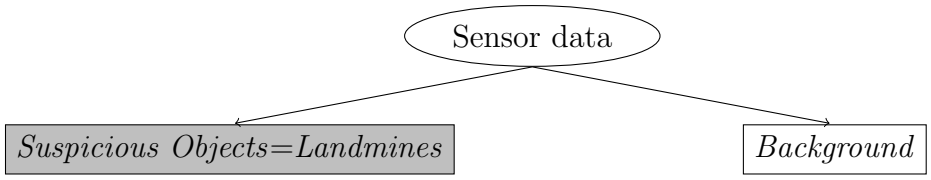


Figure 3.15: Landmine detection considering only first step

3.7 Summary

This chapter presented the novel algorithm for online detection of the suspicious objects based on their signatures obtained from several landmine detection sensors. The algorithm is able to detect the object right after it was fully scanned by the scanning device thus enabling an online processing of the data while the mobile scanning platform is exploring the terrain. Depending on the quality of sensor data it can provide the final result of landmine detection or the preprocessing result for the further recognition. Assuming the second possibility next chapter describes the algorithms developed in order to decrease the false alarm rate achieved by this first step.

Chapter 4

Landmine Recognition

This chapter describes the algorithms developed for recognition of landmines among the suspicious objects detected during the first step of landmine detection strategy (Chapter 3). The landmine recognition approach is developed in the framework of feature-level sensor fusion accounting for the specifics of this task by proposing new concepts. After a short overview of the prior art in Section 4.1 and the problem statement in Section 4.2, the classification features are described in Section 4.3. The overall classification strategy based on Bayesian classifier is presented in Section 4.4. After the discussion of the problems arising from the high nonselectivity of the classification features the novel possible solutions are presented. The concept of *selective training* aiming in improvement of classification in presence of bimodal feature distributions is described in Section 4.5. The concept of *dominant class* which allows to improve the classification results further in the conditions of low data quality is presented in Section 4.6

Section 4.5.1 describes a combined strategy for using of the developed approaches together. Finally, the developed concepts are verified experimentally and the results are presented in Section 4.8 starting from the justification of the multi-stage approach and following by the results related to each concept. The implementation details are given in Section 4.7.

4.1 State of the art

Having a suspicious object obtained during the first step of landmine detection the task of the consequent steps is to assign a class label to it. This is a classification task which can be accomplished by different classifiers as mentioned in Chapter 1. However, using different classifiers in previous reports did not lead to a clear conclusion on a preferable one. Probably, this is due to the fact that the performance of the classification depends on many other factors, including classification features, training strategy, and the availability of data for training. It must be noticed that landmine recognition is a

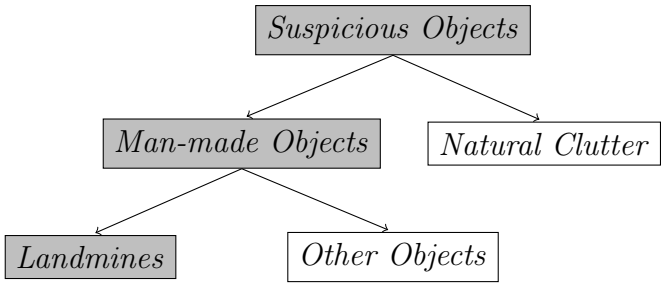


Figure 4.1: Two classification tasks

very challenging classification task due to the high ambiguity of the sensor data. As the classifier represents the top structure in this process, the best strategy seems to be the implementation of a simple classifier and using it for analyzing of the other factors.

Particularly, the analysis of classification features is one of the important tasks to be done before a conclusion about the best configuration can be made. The problem of selection and estimation of classification features is not considered with the required attention as it can be concluded from the analysis of previous works. The most used features are statistical measures ([35, 64, 23]), such as mean value inside the ROI, and measures related with simple shapes [35, 59, 34, 58]. More complex features like entropy, contrast and correlation are considered in [23] for processing of IR data, but the results are difficult to analyze due to the small number of samples in the available data sets. In [94] the possibility to use shape symmetry features for landmine detection is analyzed. However, the presented results were obtained only for well controlled conditions. Texture features are used in [65] for surface landmine detection.

In summary, there is definitely a need for a detailed research which would clarify the choice of the classifiers and classification features. Moreover, the existing solutions do not provide the performance required by humanitarian demining standards yet, so new ideas are needed.

4.2 Problem Statement

This step of landmine detection is represented, in fact, by two classification tasks as shown in Figure 4.1. These tasks are naturally the same in terms of classification strategy but should be accomplished separately in order to improve the separability of the classes and to account for the quality of sensor data.

The basic classifier considered in this work is a Bayesian classifier which models the joint probability as a multi-variate Gaussian distribution. Based on this model effective classification features should be developed. The classifier should perform as a fusing algorithm which accepts the feature vector created from the features calculated for each signature of the object. Thus, the fusion is performed on the feature-level by using this

combined feature vector.

Taking into account that landmine recognition is characterized by high overlapping of the classes, suitable solutions based on the simple classifier should be developed.

4.3 Classification Features

The first stage of any classification process is the estimation of the classification features. In general, the raw sensor data can play the role of the feature themselves. However, such strategy is prone to overtraining. It is considered here that a classification feature is a single number calculated for a ROI (its *Data Map*, *Segmented Map* or *Object Area*) or a collection of ROIs. Such features are suitable to be used in the framework of feature-level sensor fusion forming the combined feature vector.

Some of the features described below were adapted from different fields of pattern recognition mostly related to computer vision, and others (object skewness, fractal dimension and golden ratio measure) were specially developed here. The main property which a feature should reflect is the regularity of the signature, which may signify the regularity of the object itself and provide useful information for landmine recognition. As long as, there is no any perfect feature which provides this information, several less specific features can be used instead.

4.3.1 Sensor-Based Features

This group of features reflects the properties of a signature obtained from data of a single sensor.

Features based on absolute value

These are the most intuitive features. For example, when using a metal detector the landmines usually have low metal contents, while artificial metal objects have much higher metal contents. However, this criterion cannot provide good results alone because natural and artificial metal debris may also be low-metal objects.

Statistical measures: mean, standard deviation, skewness, kurtosis

Contrast

$$C = \max(C_{i,j}),$$

where $C_{i,j}$ is the local contrast

$$C_{i,j} = |x_{i,j} - (x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1})/4|$$

Features related to object shape

Sensors like IR cameras and GPR may provide a good estimation of the object shape.

Size, ***aspect ratio***

Vertical skewness of object

$$VS = \frac{\mid \max(D_i) - \min(D_i) \mid}{width},$$

where D_i is the distance between the center of the object along X axis and the local maximum along X axis

Horizontal skewness of object

$$HS = \frac{\mid \max(D_i) - \min(D_i) \mid}{height},$$

where D_i is the distance between the center of the object along Y axis and the local maximum along Y axis

Occupied part

$$OP = \frac{Area\ of\ object}{Size\ of\ ROI}$$

Compactness

$$M_{cmp} = \frac{\mu_{00}}{\mu_{20} + \mu_{02}},$$

where μ_{00}, μ_{20}, μ_{02} are central shape moments

Eccentricity

$$M_{ect} = \frac{\sqrt{(\mu_{20} + \mu_{02})^2 + 4\mu_{11}^2}}{\mu_{20} + \mu_{02}},$$

where μ_{20}, μ_{02}, μ_{11} are central shape moments

Circularity

$$F = \frac{4\pi S}{P^2},$$

where S is area and P is perimeter of the object

Features related to object nature

These features are the most challenging ones because they intend to analyze if the object is natural or artificial. One of such features is the Fractal dimension: in most cases its value for natural objects must be higher than for artificial ones. A novel feature introduced here is based on the idea of the Golden Ratio.

Fractal dimension (measure of self-similarity)

$$FD = \frac{\log(N_r)}{\log(1/r)},$$

where N_r is a number of copies of the object scaled down by ratio r. FD is estimated using a differential box-counting approach [95]

Entropy (measure of disorder)

$$H(X) = -\sum_{x} P(x) \log P(x),$$

where $P(x)$ is the probability that X is in the state x

GR measure (measure of golden ratio) The *Segmented Map* is analyzed as follows in order to determine how different the averaged ratio between two neighboring segments from the Golden Ratio is:

$$GRM = \sum_{i=1}^{N-1} grm_i/N,$$

$$grm_i = \left| \frac{\min(S_i, S_{i-1})}{\max(S_i, S_{i-1})} - \frac{1}{\phi} \right|,$$

where S_i is the size of segment i, N is a number of segments and $\phi = \frac{1}{2}\left(1 + \sqrt{5}\right)$ is the golden ratio

4.3.2 Multi-Sensor Features

Besides using of the sensor-based features there is a possibility to analyze the relation between the signatures obtained from different sensors. This feature can be calculated directly using several ROIs, as *correlation features*, or a combination of several sensor-based feature - *combined features*.

Correlation features

Distance between ROIs

Aspect ratio correlation

$$HWC = \frac{HW1}{HW2}$$

if

$$HW1 < HW2$$

and

$$HWC = \frac{HW2}{HW1}$$

otherwise. Where HW is Aspect ratio.

Correlation (image correlation)

$$Corr = \max_{s1,s2} \left[Corr(s1, s2) \right],$$

$$Corr(s1, s2) = \frac{\sum_{p1=s1,p2=s2}^{N1,N2} Vn1_{p1} \cdot {Vn2_{p2}}^2}{\sum_{p1=s1}^{N1} Vn1_{p1}^2 \cdot \sum_{p2=s2}^{N2} Vn2_{p2}^2},$$

where $Vn1$ and $Vn2$ - normalized values, $s1$ and $s2$ - coordinates of the starting points for the maps. Changing of $s1$ and $s2$ provides the shift of one map relatively to the other allowing to search for the best correlation.

Shape correlation

$$SC = \max_{s1,s2} [SC(s1, s2)]$$

$$SC(s1, s2) = \sum_{p1=s1,p2=s2}^{N1,N2} OA1_{p1} \cdot OA2_{p2}/N(s1, s2),$$

where $s1$ and $s2$ - see Correlation, $N(s1, s2)$ - number of common points for both maps.

Contour correlation

The *Object Area* is processed to obtain its contour chain code (represented by numbers from 0 to 7). Then the correlation between the chains is calculated as follows:

$$ContourCorr = \sum_{i1=0,i2=0}^{N1,N2} \frac{7 - |C1_{i1} - C2_{i2}|}{7},$$

where $C1$ and $C2$ - values of the chain codes, $|C1_{i1} - C2_{i2}|$ are adjusted to be always ≤ 4.

Combined features

The combination of several sensor-based features can improve the feature performance if they represent a general property of the object reflected in signatures of different sensors in a similar way. By other words, the feature should not highly depend on the sensor from which the signature was obtained. The process of feature combination is identical to the signal-level sensor fusion when the signals are fused, for example, by averaging.

In this work the combined features are calculated by averaging the values of the same feature calculated for signatures of the object obtained from different sensors. In this case the confidence in the feature value should increase with the number of sensors detecting the object. Moreover, even if there is only one sensor detecting the object, the feature is still present. Of course, this operation cannot be considered for very simple features like mean or size, but it can show a good performance for the more complex ones, like entropy.

4.3.3 Feature Analysis

The large number of features analyzed in this work cannot be directly used by a classifier because the amount of training data is usually not enough to estimate the joint distribution with so many parameters. Therefore, using of too many features decreases the performance of a classifier. Moreover, some features can even confuse the classifier because they do not represent useful information for the separation of classes. Each feature should be evaluated in terms of relevance, and only the most relevant ones should be considered for the classification. This process is usually called *feature selection* and is commonly used in pattern recognition tasks to reduce the dimension of the feature vector in order to reduce the processing time.

Feature evaluation

Classification features are evaluated in this work using two evaluation criteria.

Mutual information.

Mutual information gives a measure of how much information a random variable contains about another [96]. Two random variables are considered: continuous value of the feature f and discrete class c. All possible feature values are divided into K intervals with the width Δf to create a histogram. Then the mutual information is estimated as:

$$I = \sum_{i=1}^{M}\sum_{j=1}^{K} \tilde{p}(c_i, \Delta f_j) \log \frac{\tilde{p}(c_i, \Delta f_j)}{\tilde{p}(c_i)\tilde{p}(\Delta f_j)},$$

where M is the number of classes, and $\tilde{p}$ is an estimation of the appropriate probability, calculated as the number of samples appeared in the histogram interval divided by the whole number of samples. In this work 20 histogram intervals were used for feature evaluation.

Hausdorff distance.

Hausdorff distance characterizes the position of each point of one set, relatively to the points of another set [97]. It can only be applied for feature selection in problems with two classes. Considering a set F_{c1} containing feature values corresponding to the class $c1$, and a set F_{c2} containing feature values corresponding to the class $c2$,

$$H = \max[h(F_{c1}, F_{c2}), h(F_{c2}, F_{c1})]$$

where

$$h(F_{c1}, F_{c2}) = \max_{F_{c1}}[D(f_{c1}, F_{c2})] and$$
$$D(f_{c1}, F_{c2}) = \min_{F_{c2}}[d(f_{c1}, f_{c2})],$$

d is a measure of distance.

The larger the value of the evaluation criterion, the more relevant the corresponding feature should be. However, it was noticed that the nature of the evaluation criterion must be taken into account for the correct interpretation of the results:

- Large values of Mutual information do not provide good separability of classes.
- Large values of Hausdorff distance might appear due to scattered samples inside the distribution, and together with a large enough value of the Mutual information it can mean a good separability of classes.

Approach for feature selection

In order to benefit from both evaluation measures, the product of them is used as an evaluation criterion for the selection of the most relevant features. The feature selection was performed manually by choosing the features with the largest value of the evaluation criteria.

4.4 Classification

4.4.1 Multi-stage classification

As it is defined by the strategy of landmine detection the recognition of landmines should be performed in several stages according to the quality of sensor data. After each stage, only the objects of a landmine-suspicious class are taken for the further stages. In this work this process is limited to a two-stage classification, as it does not seem reasonable to extend it more with the available sensor data.

Each stage is implemented as a standard Bayesian classifier [53]. The Bayesian classifier is based on the optimal decision rule, which is in case of two classes:

$$decide\ class_1\ if\ P(class_1|\mathbf{x}) > P(class_2|\mathbf{x}),$$

$$overwise\ decide\ class_2,$$

where $\mathbf{x}$ is a feature vector representing the state of the object to be classified, and $P(class_i|\mathbf{x})$ is the probability of the object represented by $\mathbf{x}$ to belong to the class $class_i$, also called *posterior probability*. The posterior probability can be obtained by the Bayes' formula:

$$P(class_i|\mathbf{x}) = \frac{p(\mathbf{x}|class_i)P(class_i)}{p(\mathbf{x})},$$

where $p(\mathbf{x}|class_i)$ is a probability of observing feature vector $\mathbf{x}$ from an object of class $class_i$. This probability can be learned from the training data set using one of the techniques for parameters estimation [53]. Then, the decision rule can be modified as follow:

$$decide\ class_1\ if\ \frac{p(\mathbf{x}|class_1)}{p(\mathbf{x}|class_2)} > \lambda,$$

$$overwise\ decide\ class_2,$$

where λ is a constant which includes the unknown $P(class_i)$ and risk factors. Varying λ the decision can be shift in favor to one class, which can be used in case of landmine detection in order to achieve high detection rate. To obtain an optimal value of λ the training data set can be used.

The analysis made in this work showed that the probability distribution of most classification features can be modeled by a Gaussian function. Thus, $p(\mathbf{x}|class_i)$ is modeled as multivariate Gaussian:

$$p(x) = \frac{1}{(2\pi)^{n/2}\sqrt{\hat{\Sigma}}} \exp[-\frac{1}{2}(\mathbf{x}-\hat{\mu})^t\hat{\Sigma}^{-1}(\mathbf{x}-\hat{\mu})], \tag{4.1}$$

where n is the dimension of the feature vector, and Σ is an $n \times n$ covariance matrix. A maximum likelihood estimation is used to compute the parameters of (4.1) from the

Table 4.1: Number of samples of training and evaluation sets used in experiments (LM - landmines, Others - not landmine objects)

Experiment description		Training set			Evaluation set		
		All	LM	Others	All	LM	Others
3 sensors	1. Random	1287	183	1104	1287	183	1104
	2. Soil 2	666	82	584	1908	284	1624
	3. Manual	404	168	236	2170	197	1972
4 sensors	4. Random	965	129	836	965	129	836
	5. Soil 2	472	52	420	1458	205	1253
	6. Manual	351	98	252	1579	159	1420

training data set [53].

$$\hat{\mu} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x_k} \tag{4.2}$$

$$\hat{\mathbf{\Sigma}} = \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{x_k} - \hat{\mu})(\mathbf{x_k} - \hat{\mu})^t, \tag{4.3}$$

where N is the size of the training set.

4.4.2 Classifier Training

One important challenge in landmine detection is the absence of the ability to train the classifier on the data obtained directly from the minefield. In most cases it is possible to acquire some data from the background and nonhazardous objects which can be useful for adjusting parameters of suspicious objects detection. However, the availability of the data from real buried landmines should not be considered. In this situation the algorithm has to be pretrained beforehand with data obtained from other experiments made on test fields. To address this issue, different training scenarios are considered in this work:

- Randomly chosen training set. The whole data set is randomly divided in two equal sets, one of which is used for training, and another one for evaluation.

- The training set only contains data from one field with some soil type. Each soil type can be considered as a separated training experiment, and then other

fields are used for evaluation. This scenario models the case when the algorithm is trained in one type of conditions and then is used in others.

- The training set is chosen manually to contain only "good" signatures. Such scenario models a case when only well controlled experiments are available for training. If it is possible to train the algorithms in such a way, then it would simplify the experiments, and thus increase the amount of the training data.

Table 4.1 shows the number of samples of the considered training and evaluation sets. It is logic to assume that among the mentioned possibilities the use of randomly chosen training sets may give better classification results. However, in practice it is not possible to implement this scenario because the full data set (from which the random samples would be chosen) is usually not available.

4.4.3 Missed Features

A practical problem of any object recognition system is the fact that in real conditions some features can be missed. A Bayesian classifier can solve this problem by integrating the posterior probabilities over the missed features [53]. However, in the case of landmine detection, the features are missed only when a sensor does not detect the object at all (for example, a metal detector does not sense a plastic object). In this situation, all features related to that sensor are missed. Thus, there are a limited number of feature combinations determined by the number of sensors (for example, there are 15 combinations for 4 sensors: 1, 2, 3, 4, 1-2, 1-3, 1-4, 2-3, 2-4, 3-4, 1-2-3, 1-2-4, 1-3-4, 2-3-4, 1-2-3-4). It is practically possible to consider all the combinations and train several classifiers. Then, during the process of object classification, a classifier with the largest possible number of features is chosen (Figure 4.2(a)). Another solution comes automatically if only combined features are used as shown in Figure 4.2(b) because all features are always present if at least one sensor detects the object. However, the last approach is probably limited because there are not enough features which show good performance in the combined version. Thus, the strategy implemented in this work is a combination of both techniques.

4.5 Concept of Selective Training

The classification features implemented in this work basically show high overlapping of the classes, which is represented by two types of behaviors:

1. Distributions for different classes have shapes close to the Gaussian distribution and maxima are poorly separated (see, for example, Figure 4.3).

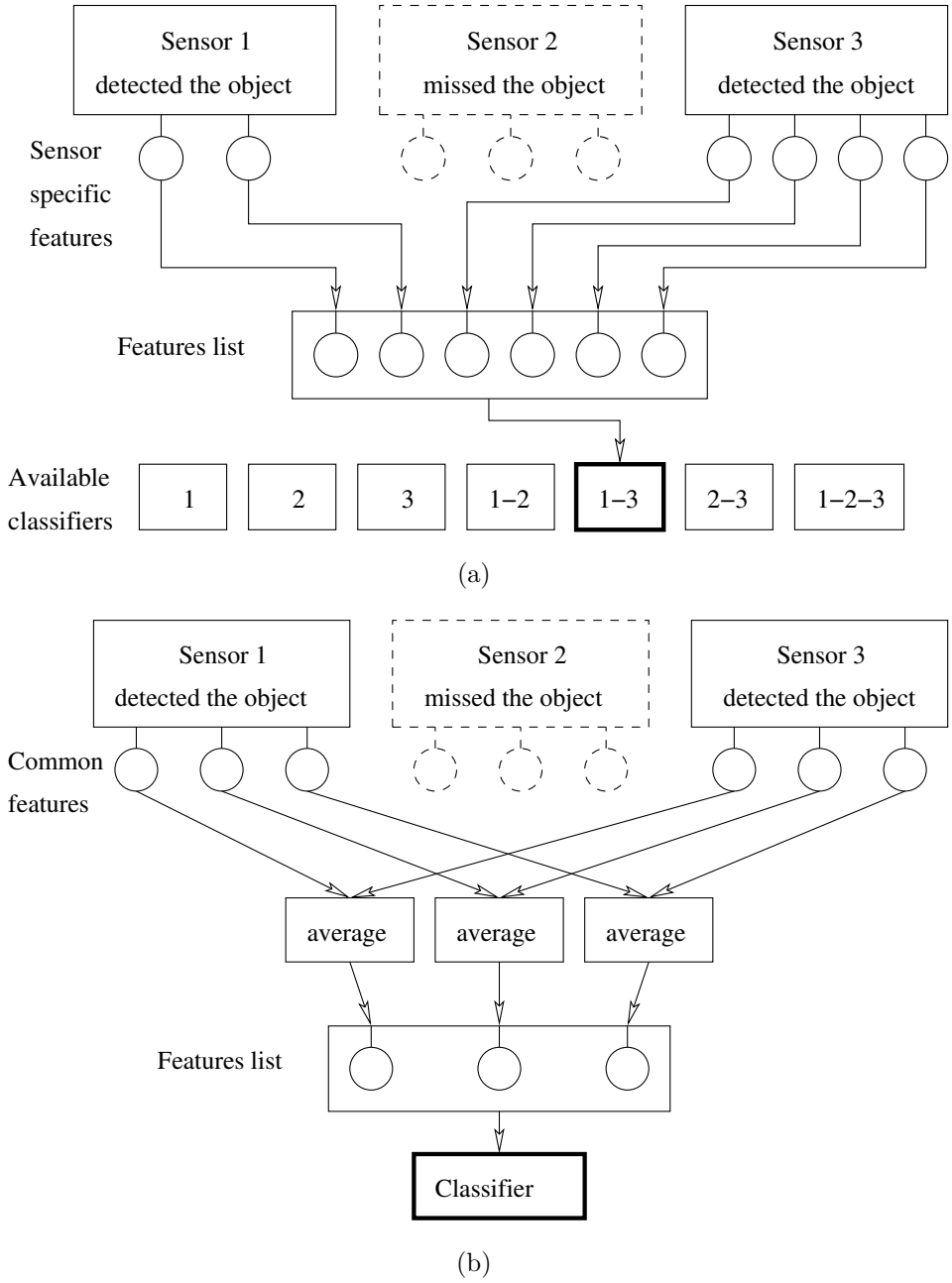


Figure 4.2: Possible solutions for missing features problem: (a) - training several classifiers, (b) - using only combined features

2. One of the distributions is bimodal: the main maxima are overlapped, and another weaker maximum is separated from the main one (see Figure 4.4(a)).

The usual strategy followed in the second case is to consider only the main maximum. However, if the feature has a weak (or equally sized) but better separated maximum, it may signify that it reflects a specific property of the landmine, but this property is not present in the majority of the samples. In these conditions, the main maximum can be ignored, and then only information from the least overlapped maximum is used for the classifier training. This process is called in this work *selective training*. It is performed

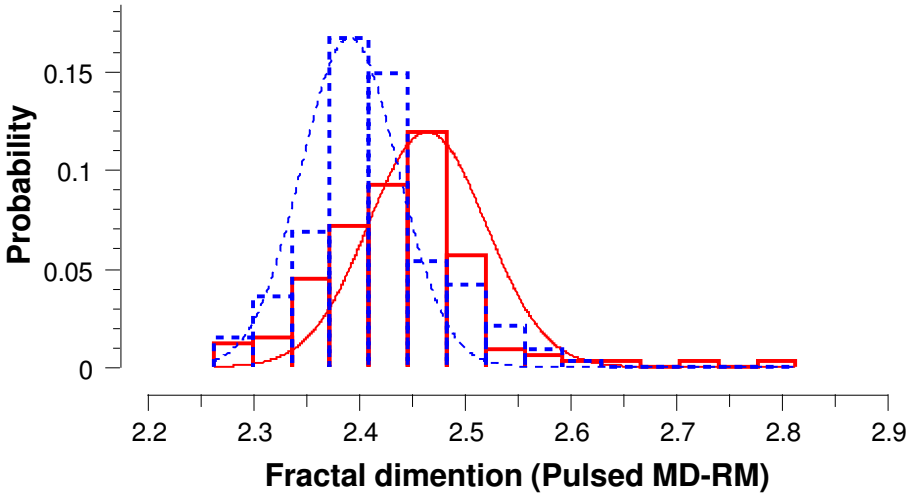


Figure 4.3: Example of a feature whose distributions for different classes are poorly separated (solid line - *Landmines*, dotted line - *Other objects*)

by analyzing and modifying the histograms of the distributions. For example, assuming that the *Landmines* class has a bimodal distribution, the algorithm is performed as follows:

1. *Landmines* distribution histogram is searched for the maximum which has the lowest overlapping with the *Other objects* distribution histogram (the maximum is considered together with the neighboring descending intervals).

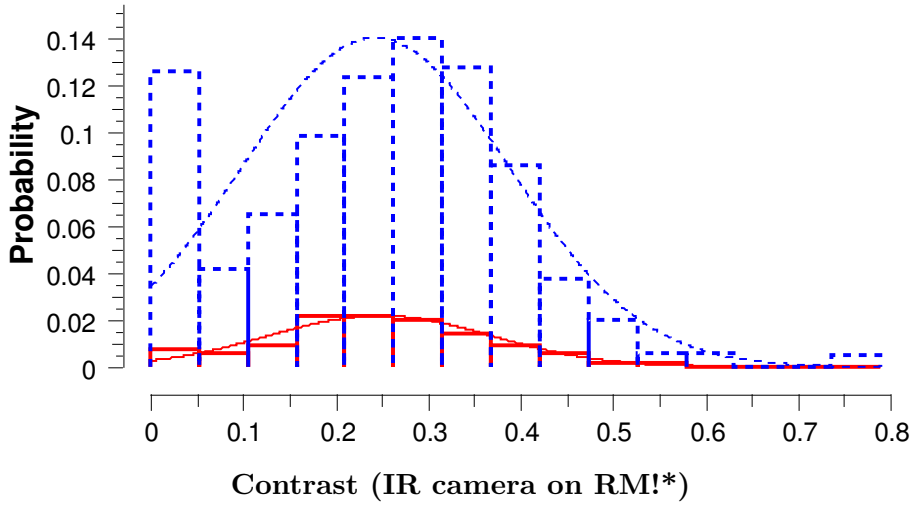
2. All histogram segments besides the found maximum of the *Landmines* distribution are reduced to zero.

3. Histogram segments of the *Other objects* distribution which overlap with the new *Landmines* histogram are also reduced to zero.

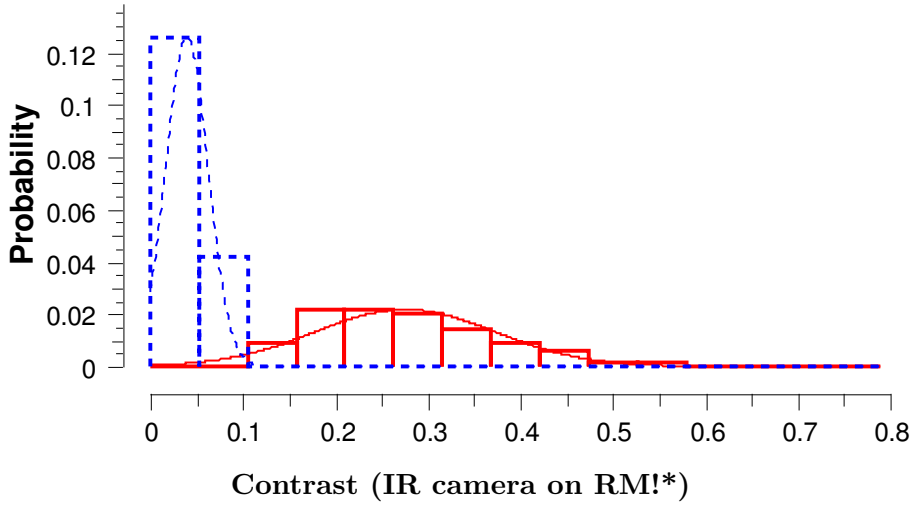
The obtained distributions (see, for example, Figure 4.4(b)) are then used for training.

If there are enough features with such property able to separate different samples, this process may improve the separability between the classes. Figure 4.5 shows an example of two features that satisfy this requirement. There are samples which are separated by both features, but there are a lot of samples which are separated only by one feature.

Assuming that only a well-separated part of the distribution of each feature is used, the relationships between the features change. It becomes inappropriate to represent $p(\mathbf{x}|class_i)$ as a multivariate Gaussian because it reflects an AND relation between the features. However, the features processed by the selective training algorithm have an OR relation to each other: the effect of one feature should be added to the others to



(a) raw distributions



(b) distributions selective for *Other objects* class

Figure 4.4: Example of a feature which has bimodal distribution for *Other objects* class: (a) initial distributions, (b) distributions processed by the selective training algorithm (solid line - *Landmines*, dotted line - *Other objects*)

assure that the object is detected if values of some of its properties are in the selected ranges (but not values of all properties, as would be the case for AND relation). In this case a more suitable model for $p(\mathbf{x}|class_i)$ is an *additive Gaussian model* which can be represented as follows:

$$p(x) = \sum \frac{1}{\sqrt{\pi}\hat{\sigma}} \exp[-\frac{1}{2}(\frac{(x-\hat{\mu})^t}{\hat{\sigma}})^2], \tag{4.4}$$

The parameters of each univariate distribution are estimated using (4.2) and (4.3).

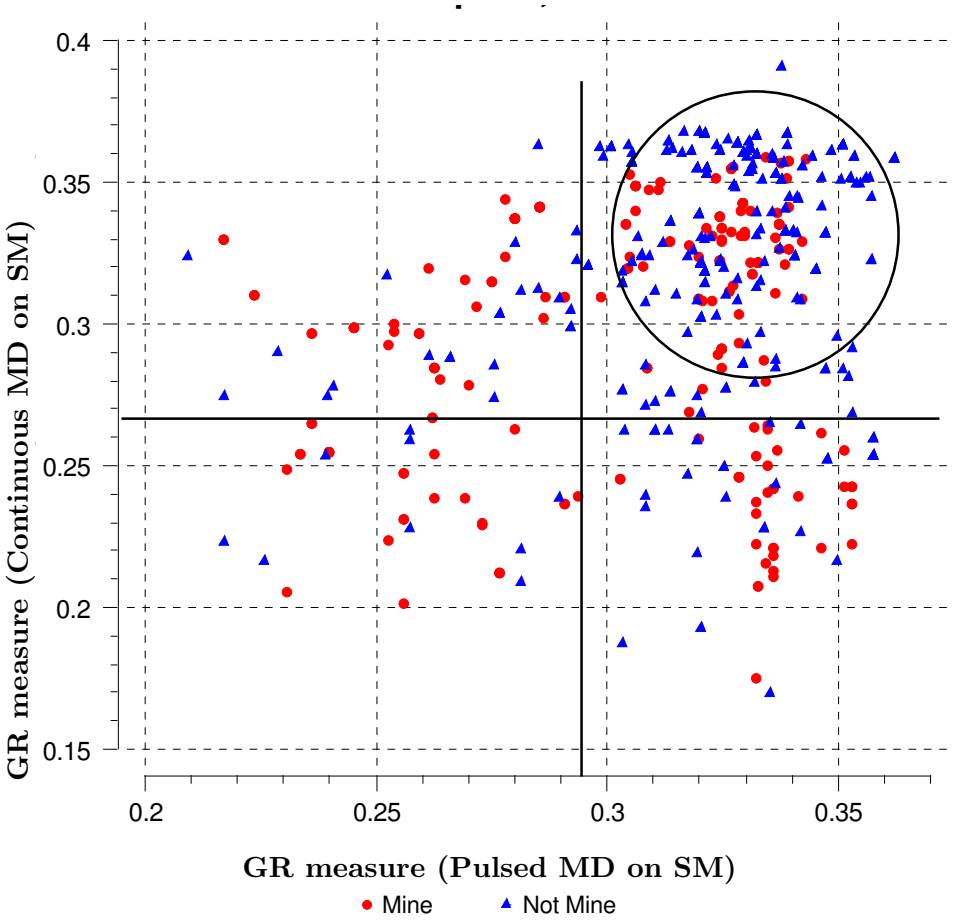


Figure 4.5: Example of two features which have bimodal distributions for *Landmines*. The highly overlapped area is indicated by the circle, the vertical and horizontal lines indicate approximate borders for the *Landmines* distribution in case of applying selective training

4.5.1 Combined Strategy

Analysis of the classification features suggested that the ideas presented above should be combined in order to provide an optimal strategy. Thus, the combined strategy merges the structures shown in Figure 4.2 assuming only some of the features to be averaged, while the others to be used directly as sensor-based and correlation features. The features which reveal the bimodal behaviors are processed by the selective training algorithm while the others are used in their initial form. The combined $p(\mathbf{x}|class_i)$ is then represented by the following equation:

$$p(x) = \frac{1}{(2\pi)^{n_1/2}\sqrt{\hat{\Sigma}}} \exp[-\frac{1}{2}(\mathbf{x}-\hat{\mu})^t\hat{\Sigma}^{-1}(\mathbf{x}-\hat{\mu})] + \sum_{j=1}^{n_2} \frac{1}{\sqrt{\pi}\hat{\sigma}_j} \exp[-\frac{1}{2}(\frac{(x_j-\hat{\mu_j})^t}{\hat{\sigma}_j})^2], \quad (4.5)$$

where n_2 is the number of the features for which the selective training is applicable, and n_1 is the number of other features.

4.6 Concept of Dominant Class

The concept of selective training allows to perform the classification in case of highly overlapped classes as in the case of landmine detection task. Experimental results show that this strategy helps to improve the quality of the recognition (see Section 4.8. However, there is one specific feature of the landmine detection task which is not fully accounted from this concept: the recognition process should provide the highest possible detection rate (which is specified to be 100% by the landmine detection standards). A usual solution for achieving a high detection rate is to choose the classification constant λ appropriately. This allows to shift the working point on the ROC curve to the area of higher DR (and consequently higher FAR). However, the problem still remains because for high detection rates a small increase of DR causes much higher increase of FAR and in most cases the DR of 100% cannot be reached at all. This follows consequently from the fact that the classification process intends to obtain the best solution in respect to the classification error which includes the quality of detection for the both considered classes. On the other side, in case of landmine detection one class (*Landmines*) should dominate allowing to achieve the 100% DR (which cannot be achieved by simply shifting the λ).

It is proposed in this section to develop the concept of selective training further to provide the domination of the *Landmines* class. Following the same principle the feature distributions are modified before they are used for the classifier training. The goal of the modification is to consider only the dominant class where the distributions overlap and let the distribution of the other class to be present only outside of the main class. Here, any type of distributions are considered, both unimodal and bimodal. The preprocessing of the distributions is performed in the following steps (see Figure 4.6):

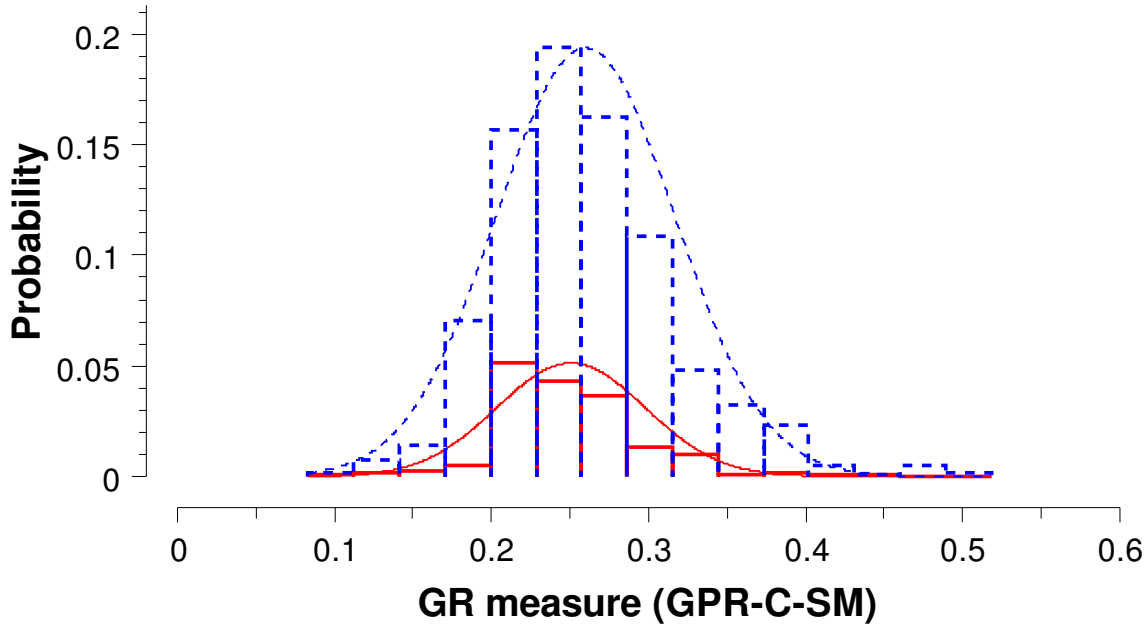
1. *Landmines* distribution histogram is searched for the main maximum (the maximum is considered together with the neighboring descending intervals).

2. Histogram segments of the *Other objects* distribution which overlap with the found maximum are reduced to zero.

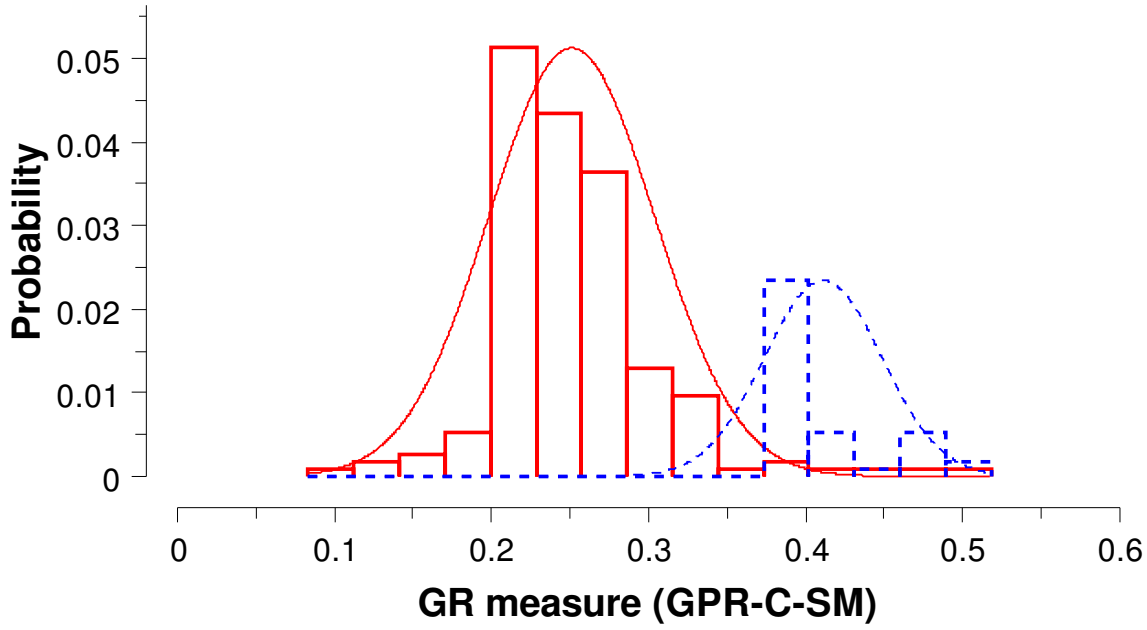
The obtained distributions are then used for the training of the classifier which later processes the data using the model (4.1).

This process obviously leads to loss of information about the *Other Objects* class. However, it might be not possible in general to perform a classification which achieves good quality in relation to both classes in case of highly overlapped distributions. By using this concept, in turn, it can be still possible to obtain the highest possible quality of classification keeping the DR on a high level.

The concept of dominant class logically supports the idea of the multistage landmine detection as a process of FAR reduction because it allows to be sure that the DR remains



(a)



(b)

Figure 4.6: Illustration of the training process to account for a dominant class: (a) - original distributions, (b) - distributions after processing assuming *Landmines* class to be dominant

on the same lever after each classification step. Considering the worst case when the distributions of the classes completely overlap, this approach will output a 100% FAR which is a poor result from the point of view of the classification error but, an acceptable result from a point of view of safety of landmine detection.

4.7 Implementation

The developed classification algorithms for landmine recognition were implemented as a set of classes representing *decision makers*. Each classifier is implemented as a separate class which provides two main methods: training and classification. The concepts of *selective training* and *dominant class* are implemented inside the training procedures.

The classification features are implemented in separate classes derived from `CGridMapFeature` class. Please refer to Appendix F for more details.

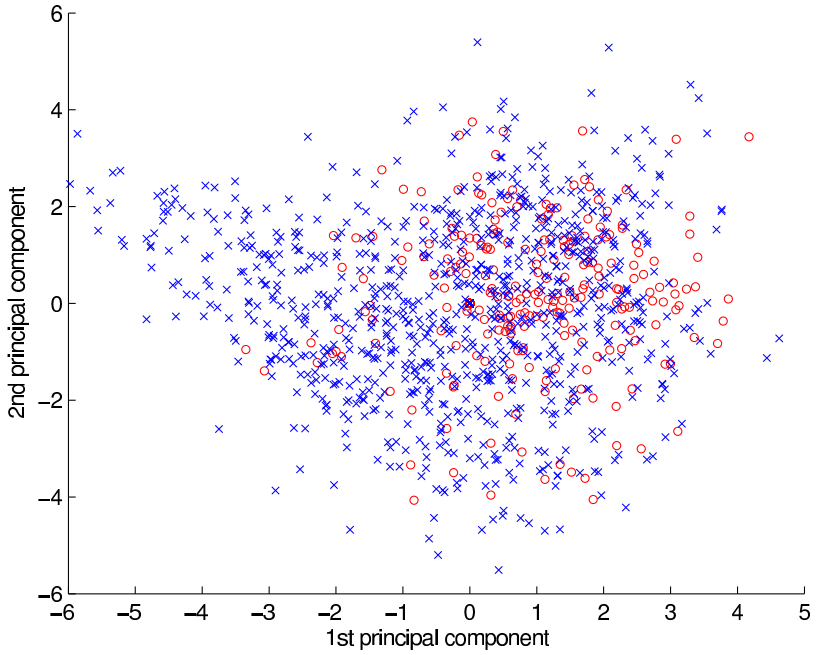
4.8 Experimental Results

The concepts developed in the present work and described above can be used together in some cases, for example, forming a *combined classifier*, while in other cases only some of them are applicable, as can be in the case of the *dominant class* concept. In this section the developed approaches are evaluated using real sensor data available in MsMs database [5]. The feature selection process was performed first, according to the strategy described in Section 4.3.3, to provide a relevant feature vector to be used for classification. To illustrate the performance of the different ideas developed for landmine detection several questions were analyzed as described below.

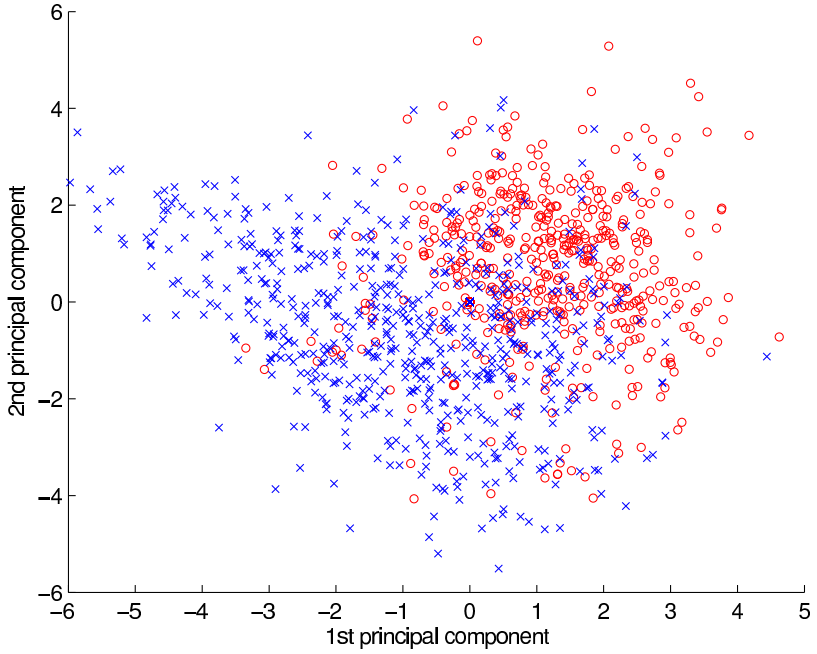
4.8.1 One-stage vs multi-stage classification

It is not obvious that for landmine detection a multi-stage classification is required. To analyze this issue, the feature space was processed by *principal component analysis (PCA)* [53]. Figure 4.7(a) shows a feature space produced by the first two principal components where the classes *Landmines* and *Other Objects* are highlighted. It can be seen that the variance in the data revealed by PCA does not reflect the difference between classes confirming the idea of highly overlapped classes in the case of landmine detection. Moreover, during such classification most false detections consist in other man-made objects: the classifier is not able to distinguish landmines from other man-made objects.

The same result of PCA is shown in Figure 4.7(b), but here the classes *Man-made Objects* and *Natural Clutter* are highlighted (these principal components provide only 31%



(a)



(b)

Figure 4.7: Principal component analysis of the feature space: (a) *Landmines* - o, *Other Objects* - x, (b) *Man-made Objects* - o, *Natural Objects* - x

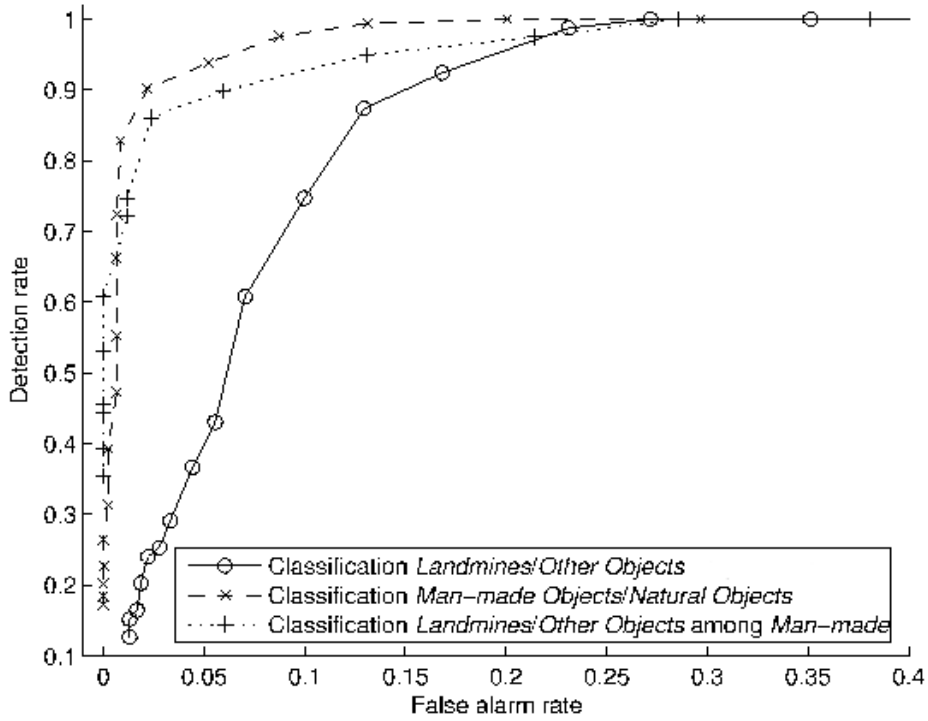


Figure 4.8: Comparison of different classifications (on the training set) in terms of detection rate and false alarm rate

of the whole variance in the data). It can be seen that these classes are better separated than the previous pair. Thus, the presented results demonstrate once again that breaking the classification in (at least) two stages as shown in Figure 4.1 is reasonable and can improve the final result.

Another possibility to support the two-stages approach is to compare the classifications involved in the one-stage and two-stage classifications in terms of detection and false alarm rates. A comparison of ROCs for different classifications shown in Figure 4.8 also confirms this idea. It can be seen that, as expected, the classification with classes *Man-made Objects/Natural Clutter* has better performance than the classification with classes *Landmines/Other Objects*. Moreover, the classification *Landmines/Other Objects* performs better when used to distinguish among the *Man-made Objects*.

4.8.2 Influence of the training set

The problem of classifier training is important due to the difficulties in obtaining of the sensor data for landmine detection. To analyze this issue a classifier which contains only sensor-based features was constructed eliminating the influence of the *combined features* and *selective training* on the results. Table 4.2 shows the results of classification using different combinations of training/evaluation data sets and the mentioned classifier with

Table 4.2: Sensor fusion results for different training/evaluation sets, N - number of sets from Table 4.1

N	Performance on training set			Performance on evaluation set		
	DR, %	FAR, %	FA/m^2	DR, %	FAR, %	FA/m^2
1	98	36	3.3	96	37	3.3
2	90	13	2.7	70	23	2
3	90	35	3	74	37	4.9
4	96	35	4.7	95	35	4.9
5	100	13	2.8	77	20	2.8
6	93	23	2.6	74	19	3

Table 4.3: Comparison of sensor fusion results for different feature sets and different types of training, N - number of sets from Table 4.1

N	Training	Performance on evaluation set		
		DR, %	FAR, %	FA/m^2
1	normal	90	30	2.7
	selective	95	32	2.8
4	normal	90	32	4.1
	selective	95	32	3.9

fixed λ. Three possibilities for the forming of the training set were considered: random, manually chosen "good" signatures and data from a selected type of soil (Table 4.1). As expected, the using of the randomly created training set leads to the best classification results (Table 4.2 with N=1 and N=4). From the two other possibilities, the training based on the selected experimental field is the most practical because it shows better results (Table 4.2, N=2,5) than in the case of manual selection (Table 4.2, N=3,6). This signifies that the manually chosen training set does not contain enough information due to, for example, the absence of clutter. The ability to train the classifier on one type of soil is important for the practical implementation of the automated landmine detection system in which neither random nor leave-one-out training is possible.

4.8.3 Selective training

In order to show the benefits of the concept of selective training, the performance of the classification with and without it was compared. A combined classifier containing sensor-based and combined classification features was created for this purpose. In one case the classification features with bimodal distributions were trained normally (without

Table 4.4: Changing of false alarms per m^2 for detection rate 95% over the steps of landmine detection

	ROIs extraction	Two-step classification	
		Man-made Objects/ Natural Clutter	*Landmines/ Other Objects*
FA/m^2	12.1	5.4	4.5
		One-step classification	
		7	

ignoring the parts of the distribution), in the second - according to the *selective training* concept. Table 4.3 shows the results of applying this classifier with fixed λ for both cases. It can be seen that introduction of the *selective training* improves the results.

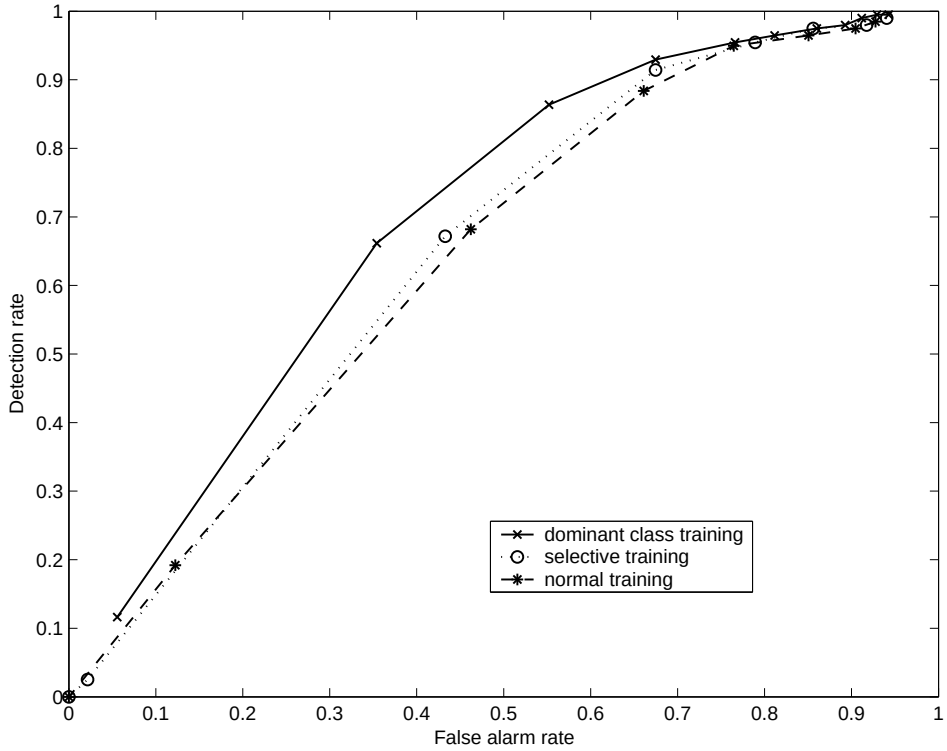
4.8.4 Dominant Class

The *dominant class* concept is tested here using the experimental data from [5]. These data are extensive allowing the *selective training* to be performed as it was confirmed in the previous section. While the *dominant class* concept is expected to be specially effective in the case of poor sensor data, the results of applying both concepts are very similar in the case of rich sensor data. To perform a clear comparison of the training concepts the performance of the classifier was intentionally degraded by using only few *combined features* simulating a situation when the sensor data are not enough to perform the training. Training and evaluation of the classifier were carried out with the sets under number 5 in Tab. 4.1. The results of comparison of three training techniques in this case are shown in Figure 4.9. The *dominant class* training outperforms the *selective training* and normal training (where the distributions of the classification features are not changed). The *selective training*, in turn, outperforms the normal training confirming again its usefulness.

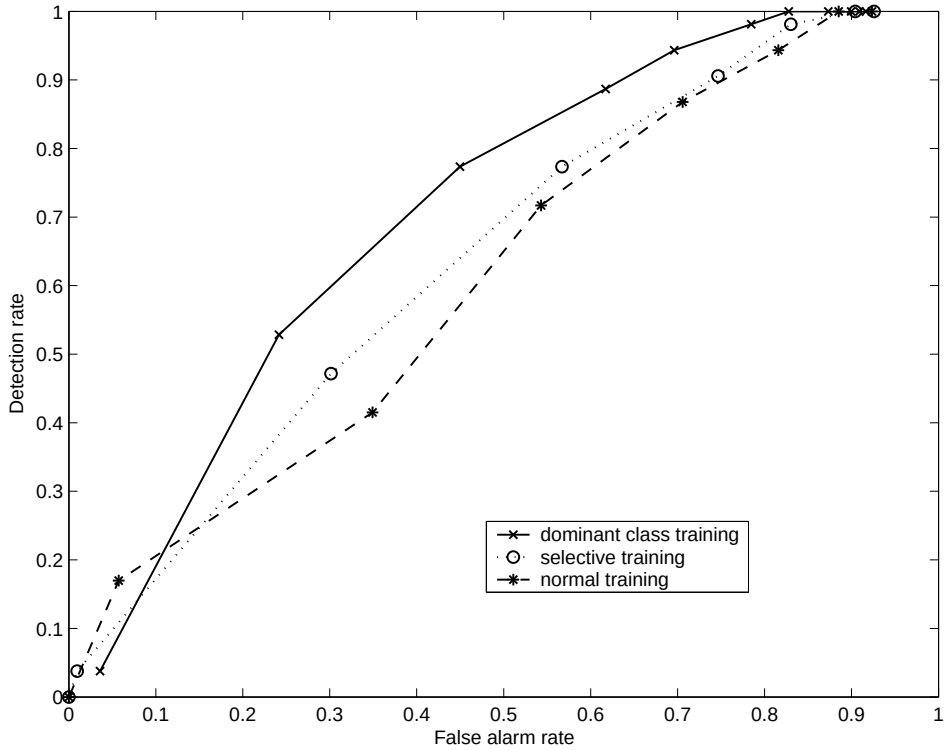
A more appropriate case when the *dominant class* training is vital is considered in Chapter 7 where the landmine recognition is performed on the sensor data obtained by the mobile platform LADERO.

4.8.5 Step by step reduction of false alarm rate

In the previous subsection only the one-stage classification was performed in order to evaluate other approaches. Here the multi-stage strategy can be finally evaluated by providing specific landmine detection results. A combined classifier incorporating *combined features* and *selective training* where applicable is used for this purpose. The classifier is trained on the data set obtained from soil type 2 in [5] and evaluated on the rest of the data. The developed multi-step landmine detection strategy allows to track



(a)



(b)

Figure 4.9: A comparison of ROCs for different training strategies: *dominant class* training, *selective training* and normal training (without the change of distributions), (a) - performance on the training set, (b) - performance on the evaluation set

the reduction of the false alarm rate at each step therefore monitoring the improvement of the recognition (assuming that all objects of the landmine-suspicious class are considered to be landmines). Table 4.4 shows how the FAR is changing over the stages. It starts from a very high FAR after the detection of suspicious objects, and is constantly reduced. Comparing the final FAR with the one for one-step classification illustrates a significant improvement.

4.9 Summary

This chapter completes the development of landmine detection strategy proposed in this work. The strategy itself has the necessary ability to detect the objects online, while the mobile platform is scanning the area, thanks to the suspicious object detection algorithm. The detected objects can be further processed to improve the recognition using the algorithms described in this chapter, if the quality of the sensor data is sufficient enough. The recognition algorithms are especially suitable for landmine detection accounting for the high overlap of the classes by implementing the *selective training* and by introducing the *dominant class*. Both concepts deserve attention as they shall be used in different situations depending on the quality of the sensor data. Having this processing available the next challenge consists in the ability of the robot to acquire appropriate sensor data. This problem will be addressed in the next chapter.

Chapter 5

Platform Positioning during Scanning

The landmine detection approach proposed in this work analyzes the spatially mapped sensor data in order to distinguish the signatures of the landmines among the other objects and the background. Such technique allows to benefit from the usage of an automated landmine detection platform, in contrast to manual techniques when only one-point measurements are usually possible. In order to provide the spatial mapping of data, the platform should be able to perform scanning of the area with appropriate precision. This includes appropriate control of the robot in order to perform the movements required for scanning, and localization of the current position of the landmine detection sensors. The specificity of this task is the need of accurate local positioning of the robot body while walking in rough terrain affected by slippage problems. At the same time the global positioning of the robot is less important assuming, for example, that the locations of the detected landmines can be marked directly on the ground. In this chapter the problems related to the task of platform positioning are analyzed in order to find possible solutions and to conclude if the used scanning platform is sufficient for the job.

An overview of the prior art in the fields related to robot positioning is given in Section 5.1. The odometry system of the robot is described in Section 5.2. The factors which can affect its reliability are analyzed in Section 5.3 together with the solutions developed in this work for the most critical problems of the platform positioning. To improve the positioning performance, a possibility to use an additional relative positioning system based on vision is analyzed in Section 5.4. The developed vision system is combined together with the odometry system of the robot by means of data fusion based on Kalman filter. This process is presented in Section 5.5 followed by implementation details in Section 5.7 and the experimental results in Section 5.6.

5.1 State of the art

The problem of robot localization is considered in many previous works. This problem is vital in most of the applications of mobile robots, from a package delivery task to exploration missions. The developed approaches often incorporate both localization and mapping, and thus are called Simultaneous Localization and Mapping (SLAM). However, there are no investigations about positioning of a scanning platform which requires high accuracy on the local level. Among localization methods there are mainly two groups: *relative localization* and *absolute localization*.

The *relative localization*[1] approach is based on incremental updating of the robot position using only internal robot information. The calculation of the position is performed relatively to some known initial position. The input information for a relative localization system is usually obtained from dead-reckoning sensors, e.g. encoders for a wheeled robot. In case of a legged robot the calculation of the position can be performed in a similar manner based on relative positions of the legs and the body of the robot, although there are no standard dead-reckoning sensors available (see for example [98]). Moreover, there are additional sensors which can be used for continuous control of the robot position: accelerometers, gyroscopes and compasses. Relative positioning system is usually cheap and simple to implement. However, due to its incremental nature it has a drawback consisting in the accumulation of small position errors. The accumulated error can be reset from time to time using an absolute localization system.

As a mathematical basis for relative localization, the Kalman filtering technique is widely used. It allows the fusion of data from several sensors and improve the overall performance of the system [99, 98]

Absolute localization techniques provide information about the robot location in each given position without taking into account the previous position and motions of the robot. The main idea of these approaches is to use information about range and angle between the robot and several known markers present in the environment. The markers can have different nature, and mainly there are two types: active beacons and landmarks (natural and artificial). Localization is performed in the following main steps:

- In the beginning of the operation the robot is provided with a map which contains the locations of the markers relatively to the global coordinate system
- When the position of the robot needs to be determined, it identifies surrounding markers and finds their positions on the map. Then, each marker is localized in the robot coordinate system
- If there are at least three markers with know positions then the robot position relatively to the global coordinate system can uniquely be found using a triangulation

[1] It is also called *position tracking* or *trajectory control*.

or a trilateration approach

In the triangulation approach it is assumed that the distances between the markers are known. After the visual angles between markers relatively to the robot are determined, it can be said that the robot is situated on an arc of the circle spanned by each pair of markers. Trilateration approach determines robot position using values of distances between the robot and markers: the robot is situated on the circle with the center in the marker location and radius equal to distance between the robot and the marker. As well as in case of the triangulation at least three markers are needed to find robot position as intersection of the circles generated but each marker.

In spite of the large history of the described technologies there are still unsolved problems related to the precise localization of a mobile platform like the one used in the present work. The goal of the methods presented below consists in aiming to improve the positioning of the platform by using both internal and external resources.

5.2 Robot Odometry

Analysis of previous reports shows, that an absolute localization system is not applicable for the mobile scanning platform due to the high cost in case of good precision systems (e.g. DGPRS or difficulties in the deployment of active beacons). Thus, it is assumed in this work that only relative techniques, which do not require any external systems, can be used. So, the localization of the robot is performed using its odometry system. In this case the safety of the robot does not depend on the functionality of the external systems, which can be disturbed by unpredictable outdoor conditions, e.g. occlusion of line-of-sight to the active beacon or of a satellite.

Current position of the robot body is determined incrementally using its position relatively to the legs of the *moving axes*. The location of the robot at time t is determined as follows:

$$X_t = X_0 + \sum_{k=1}^{N_t} l_k + \Delta l_{N_t+1}(\Delta t), \tag{5.1}$$

where l_k is the length of step k, N_t is the number of full steps performed by the time t, $\Delta l_{N_t+1}(\Delta t)$ is the position of the robot body relatively to the back legs of the *moving axes* (see Figure 5.1), and Δt is the time passed since the beginning of the current step.

The simple pneumatic cylinders used for the robot construction do not contain the in-built possibility for the continuous measuring of the piston rod position. To provide this information a contactless measuring system based on ultrasound sensors for the measuring of the leg positions relatively to the robot body was previously implemented [4]. However, experiments showed that this system is not reliable, being dependant on the environmental conditions. Thus, a simpler approach which utilizes discrete magnetic

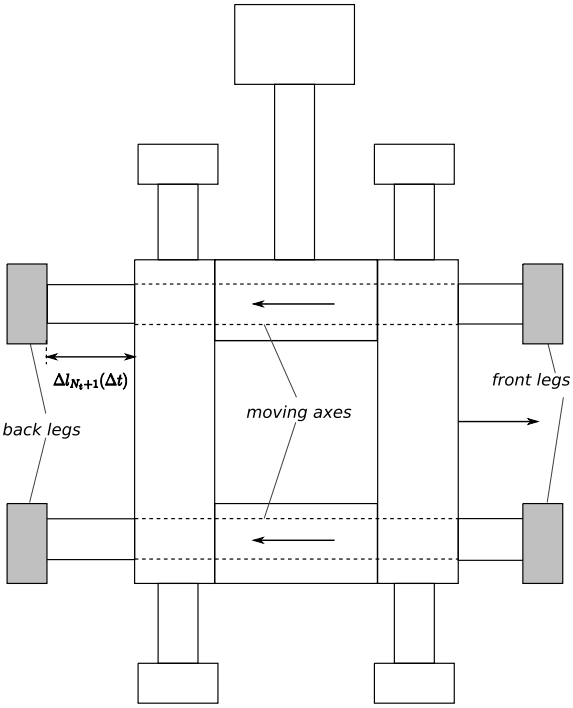


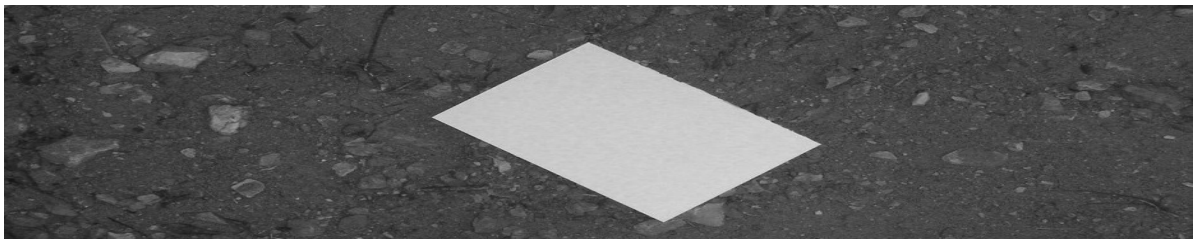
Figure 5.1: Determining the location of the robot body for odometry calculation (5.2)

Figure 5.2: Sensors for the position control

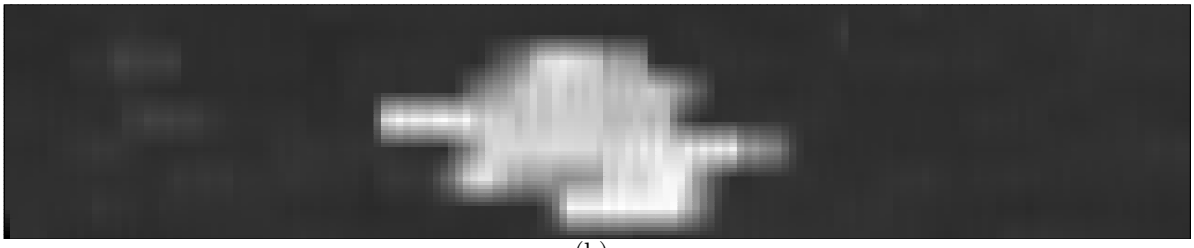
sensors was proposed in the present work. The sensors are installed directly on the cylinders as shown in Figure 5.2. The position of each sensor can be adjusted with high accuracy and reliably fixed. It is assumed that the speed of the body is constant during each interval between sensors. Then, the position of the robot body for the current step is determined as follows:

$$\Delta l_k(\Delta t) = \Delta s_m + (\Delta t - t_m) \cdot V_{m \to m+1}, \tag{5.2}$$

where m is the number of the last sensor passed, the time Δt is the interval from the beginning of the step k, Δs_m - position of sensor m, t_m - time when the sensor m was



(a)



(b)

Figure 5.3: Experimental results of the mapping obtained by the robot from the *point video sensor*: (a) the real image of the covered area, (b) obtained map

passed (relatively to the beginning of the step), and $V_{m \to m+1}$ is the speed of the body between sensors m and $m+1$. In the beginning of the the operation a calibrating *scanning step* is performed to determine the speeds $V_{m->m+1}$ on each segment between the sensors. During the sequent steps the speeds are constantly recalculated in order to account for the changing conditions, so $V_{m->m+1}$ in (5.2) is the value of the speed calculated during the step $k-1$.

There are three magnetic sensors installed on each of the *scanning cylinders* (as shown in Figure 5.2) and one sensor installed on each *advancing cylinder*. For the *Advancing step* the sensors are used as stopping points, so their locations determine the length of the step (currently adjusted to 50 mm). The sensors installed on the *scanning cylinders* are used for the calculation of the position in (5.2) and as stopping points for performing smaller steps in order to provide a more precise obstacle avoidance.

The presented positioning system allows to acquire sensor data approximately every 5 mm during the *scanning step*. This rate is determined by the relation between the speed of the *scanning cylinders* and the speed of the data acquisition. For the experiments carried out in this work the speed of the *scanning cylinders* was adjusted to be approximately $40mm/s$. Faster speeds were found to reduce the reliability of the movement.

To evaluate the mapping abilities of the robot, the following simple experiment was performed:

- A point sensor was emulated using a Web-camera horizontally mounted on the robot pointing to the ground. An averaged grayscale value of the central region of the image was used as the sensor value, providing a *point video sensor*

- The experiment was performed indoor using a box with real soil and a high-contrast object for the sensor input

During the experiment the robot followed a scanning path covering the area of 800x280 mm. The obtained image is shown in the Figure 5.3 together with the real image of the covered area. Due to the robot slippage, the final position of the robot as recorded by odometry was shifted by 10 mm along the X axis and 20 mm along the Y axis from the real one, and the robot rotated by 2°. On a longer path and in more complex outdoor conditions, the increase of odometry errors can be expected. The most relevant problems are caused by the small rotations of the robot body since such rotations significantly deform the trajectory and cannot be corrected due to the mechanical limitations.

5.3 Platform Positioning

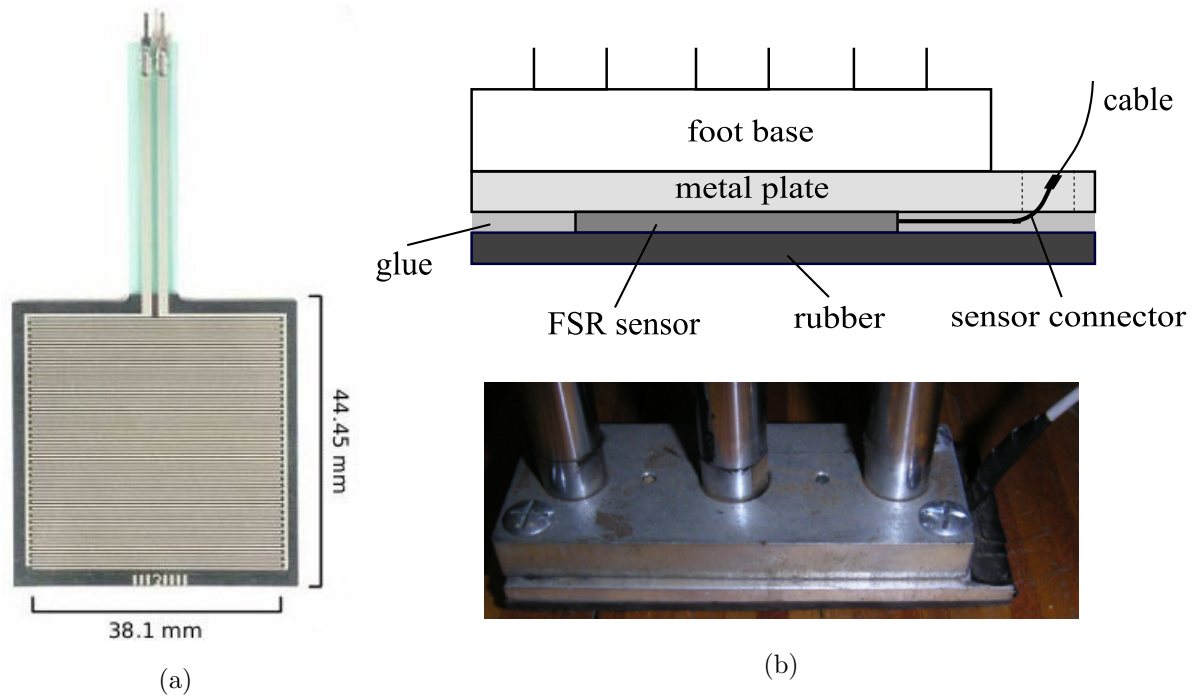
An important property of the utilized mobile scanning platform is its simplicity, which in turn causes its relatively low cost. Additionally, the positioning of the platform can be also realized by simple control algorithms. In ideal conditions the positioning can be performed by the basic steps (*scanning step* and *advancing step*) while the position is calculated using the techniques described in the previous section. However, due to its simple structure, the reliable positioning of the robot implies several assumptions, which are not always held in a nonideal situation:

- During the *scanning step*, the body of the robot is moved by two parallel cylinders. In order to provide a reliable movement, the cylinders should move with equal speeds. Experiments showed that this assumption does not hold in most cases. At the same time, the difference in speed of parallel cylinders provides bending of the robot and can even make the movement to stuck completely. Such deformations also provoke slippage and small rotations.

- Due to the cartesian structure of the platform, a situation when at least one foot does not have ground contact is easy to occur. This happens even while walking on a flat surface because the legs do not have exactly the same height. Lack of the ground contact provides in turn slippage and rotation of the robot.

During experiments it was found that the two situations described above significantly affect the quality of the robot positioning. To improve the situation, additional simple algorithms described below were developed.

In order to control the foot ground contact, force-sensitive sensors were mounted on the feet as shown in Figure 5.4. The utilized sensor is a square-shaped Force Sensing Resistor (FSR) whose resistance changes depending on the applied force (from several $M\Omega$ in unloaded state to several hundreds Ω). The FSR is included into a voltage divider



(a) (b)

Figure 5.4: Foot ground contact sensor (a) and its mounting on the foot (b)

and the voltage measured on the FSR V_{FSR} is considered to be the output signal of the sensor. To determine if the ground contact is present the value of V_{FSR} is compared to a threshold which corresponds to an appropriate value of applied force (weight of the robot). It is practically very challenging to determine the threshold by measuring the values V_{FSR} corresponding to the ground contact because in most cases it is not possible to provide a reliable ground contact of several legs simultaneously (due to the different height of the legs as it was described above). Thus, it was decided to use the value of V_{FSR} measured in unloaded state (when the foot is lifted) as reference V_{FSR}^{air}. Experiments showed that the value $0.8V_{FSR}^{air}$ can be used as the threshold. Thus, the foot contact is determined as follow

$$Foot\ contact = \begin{cases} 1 & if\ V_{FSR} < 0.8V_{FSR}^{air} \\ 0 & if\ V_{FSR} \geq 0.8V_{FSR}^{air} \end{cases} \tag{5.3}$$

The FSR sensors need to be periodically calibrated because their reference value may change due to a mechanical disturbance (small shift of the mounting rubber, dust, etc.). Thus, each time the leg is lifted (the topmost position of the leg cylinder is measured by a magnetic switch) the value V_{FSR}^{air} is recorded.

Having the information about the ground contact of the feet, an algorithm for adaptive landing of the legs providing reliable ground contact of all legs was developed. An additional objective was to decrease the mechanical stress applied to the robot at the moment when the legs are landed, which can provide extensive shaking of the robot and provoke slippage.

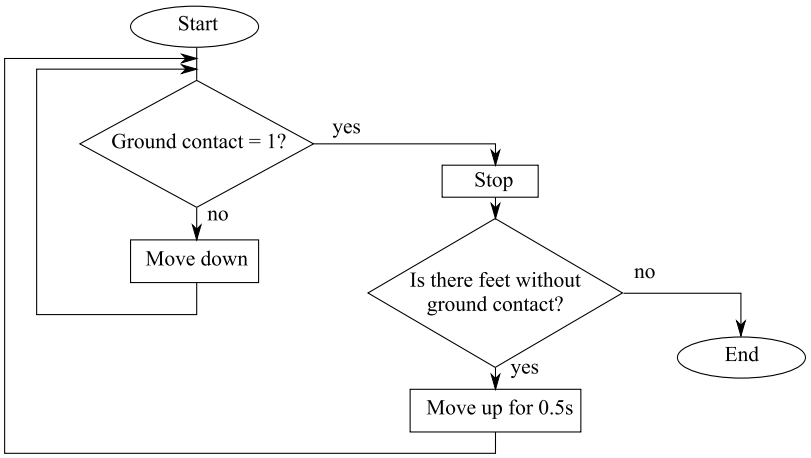


Figure 5.5: Leg landing algorithm (the loop is controlled by a timeout not shown on the diagram)

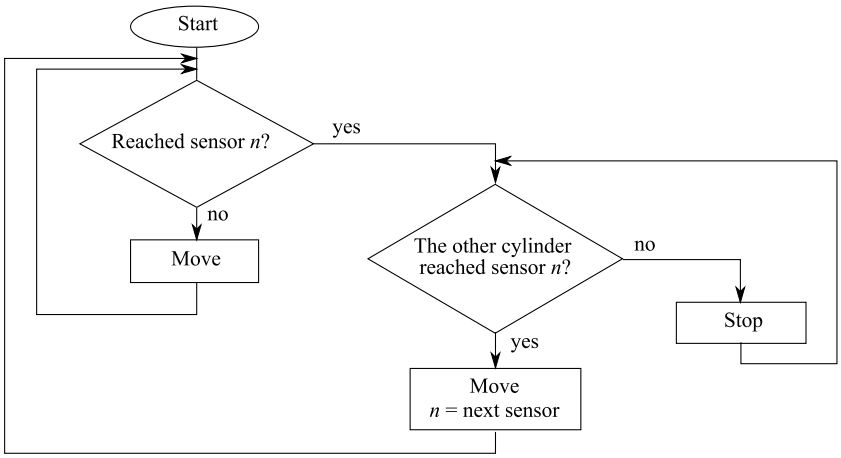


Figure 5.6: Algorithm for moving of the cylinder during the *scanning step*

The algorithm consists of simple rules applied to each landing leg as shown in Figure 5.5. It allows the legs, which already have a ground contact, to support the remaining legs by moving up by a small distance. This algorithm executed for each leg leads to a global behavior of the robot which consists in adapting to the ground surface allowing also less shock at the moment of landing.

Assuming that the required ground contact of the robot feet can be provided, the problem with the inequality of the cylinders speeds still remains. In practice, even if the speeds are adjusted properly, they depend on the current conditions (e.g. ground surface) and may change during the operation of the robot. Thus, it is assumed that the speeds of the parallel cylinders are in general different. To account for this difference the

algorithm shown in Figure 5.6 is executed for each cylinder. The algorithm implies that the faster cylinder waits for the slower one at reference points specified by the installed magnet switches. This way if the speeds are different, one of the cylinders will move in steps and prevent a large deformation of the robot.

Although having these improvements, the robot trajectory is still affected by small slippage and rotations which are intended to be detected with the help of the vision system described below.

5.4 Vision based positioning

Vision based navigation for mobile robots has been extensively explored by many researchers (see, for example, [100]). Many developed algorithms provide good results in indoor environment, but still there is a number of difficulties for utilizing this concept in outdoor and specially in unstructured environments, which is the typical environment of a demining robot. The largest challenge for outdoor (and unstructured) vision-based navigation seems to be the detection of natural landmarks, because a well-defined landmark model is not available. An appropriate correlation function is also required for the association of the detected landmarks. Moreover, it is assumed here that the landmark detection algorithm should take into account the data association problem: the landmark should be good enough so it can be associated with itself at the next moment.

A number of approaches for landmark detection in natural environment have been investigated in the previous works. For example, [101] uses color and orientation features utilizing the concept of saliency map, [102] uses the Scale Invariant Feature Transform (SIFT) descriptors and a Gaussian filter for identifying natural landmarks in forest environment, [103] proposed to use fractal dimension for landmark extraction when the landmarks have some texture property.

5.4.1 Detection of Natural Landmarks

The objective for the development of a new algorithm in this work is to provide a simple solution for the problem addressing the specificity of mapping for a demining robot. The nearest surroundings of a demining robot usually contain few obstacles. Distant objects, like mountain peaks or the sun, do not provide enough information for relatively small movements of the robot. However, suitable landmarks can be found on the ground under the robot, because the natural ground usually contains a lot of objects like stones and leaves. A simple Web-camera pointed to the ground is utilized for this purpose as shown in Figure 5.7. It should be pointed out that this approach is not appropriate for a completely empty terrain like, for example, in a desert. However, the proposed vision system is considered as supplementary to the odometry of the robot. Thus, its

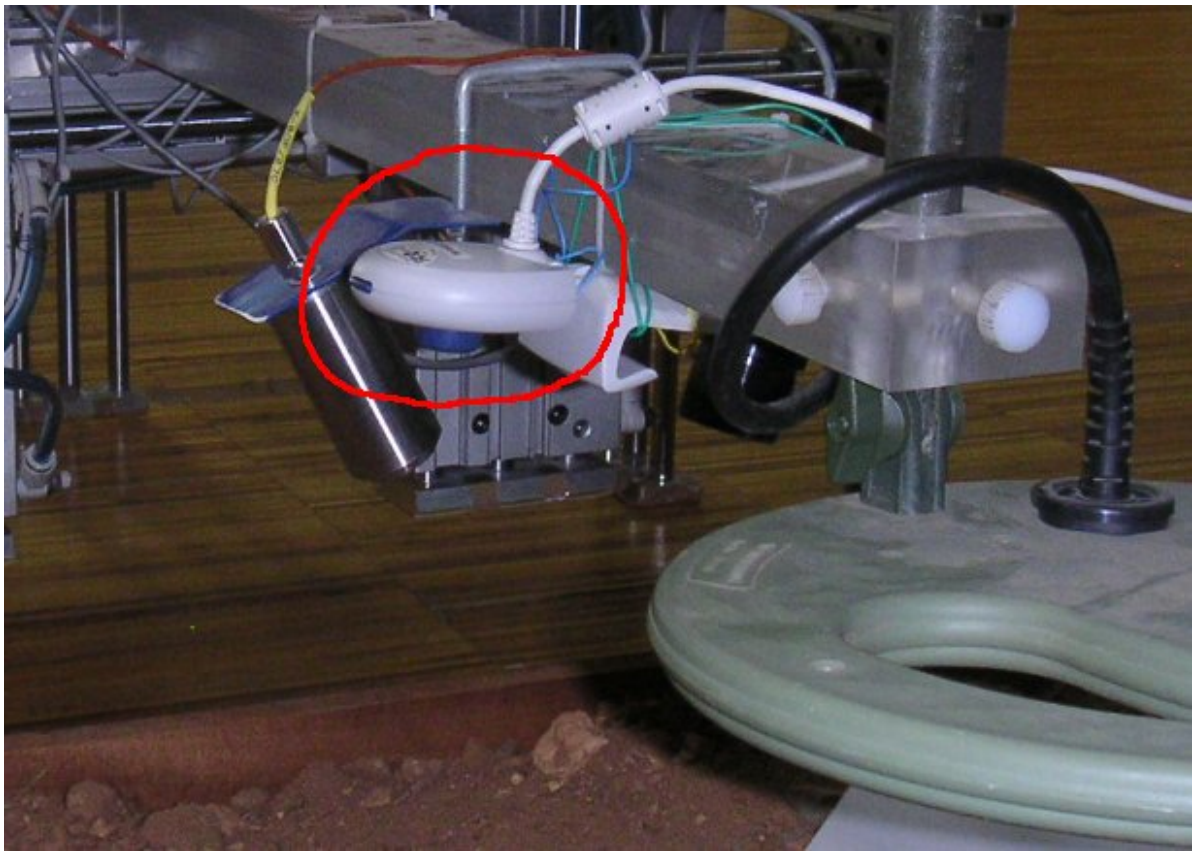


Figure 5.7: A camera mounted on the robot for landmark detection.

temporary inaccessibility in case of too few or no landmarks does not disable the whole localization of the robot.

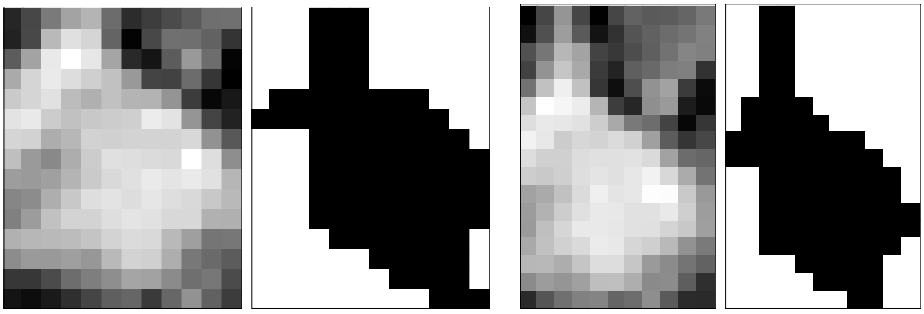
The landmark detection algorithm developed in this work utilizes a similar principle as the ROI extraction algorithm for detection of suspicious objects. But for the algorithm used here only segments which contain extrema can be joined to form the *Object Area*. The criterion for object detection is general, so no special landmark model is required. The algorithm detects only complete objects implying that they will have a good chance to be detected in the next frame and can be successfully associated. An example of a detected and associated landmark can be seen in Figure 5.8.

5.4.2 Landmark Association

The landmark detection algorithm provides a number of landmarks each time a new image from the camera is obtained. Then the algorithm has to associate each new landmark with some landmark from the previous image (if the corresponding landmark exists). The information provided by the landmark detection algorithm allows to use for this purpose the following correlation functions:

- *Image correlation.*

  This measure is widely used in computer vision, for example, in stereo-vision. In



(a) (b)

Figure 5.8: An example of the detected landmark (raw image and shape): (a) and (b) show the same landmark detected on different frames and successfully associated with itself

this work the image correlation is calculated between the images of two landmarks as follows:

$$Corr = \max_{s1,s2} [Corr(s1, s2)]$$

$$Corr(s1, s2) = \frac{\sum_{p1=s1,p2=s2}^{N1,N2} Vn1_{p1} \cdot Vn2_{p2}{}^2}{\sum_{p1=s1}^{N1} Vn1_{p1}^2 \cdot \sum_{p2=s2}^{N2} Vn2_{p2}^2}$$

Where $Vn1$ and $Vn2$ are normalized values of the images, $s1$ and $s2$ are the coordinates of the starting points for the images. Changing of $s1$ and $s2$ provides the shift of one image relatively to the other allowing to search for the best correlation.

- *Shape correlation.*
  An estimation of the shape provided by the landmark detection algorithm is used to calculate the measure of shape correlation as follows:

$$ShapeCorr = \max_{s1,s2} [ShapeCorr(s1, s2)]$$

$$ShapeCorr(s1, s2) = \sum_{p1=s1,p2=s2}^{N1,N2} \frac{OA1_{p1} \cdot OA2_{p2}}{N(s1, s2)}$$

  Where $s1$ and $s2$ - see *image correlation*, OA is the *Object Area* of the object (taking values 0 or 1) and $N(s1, s2)$ is the number of common points for both landmarks for the corresponding $s1$ and $s2$.

- *Contour correlation.*
  The shape of the object can be also analyzed using its contour. For this purpose the *Object Area* of the landmark is processed to obtain its contour chain code: a representation of the contour by numbers from 0 to 7, where each number signifies a direction of the current pixel relatively to the previous one.

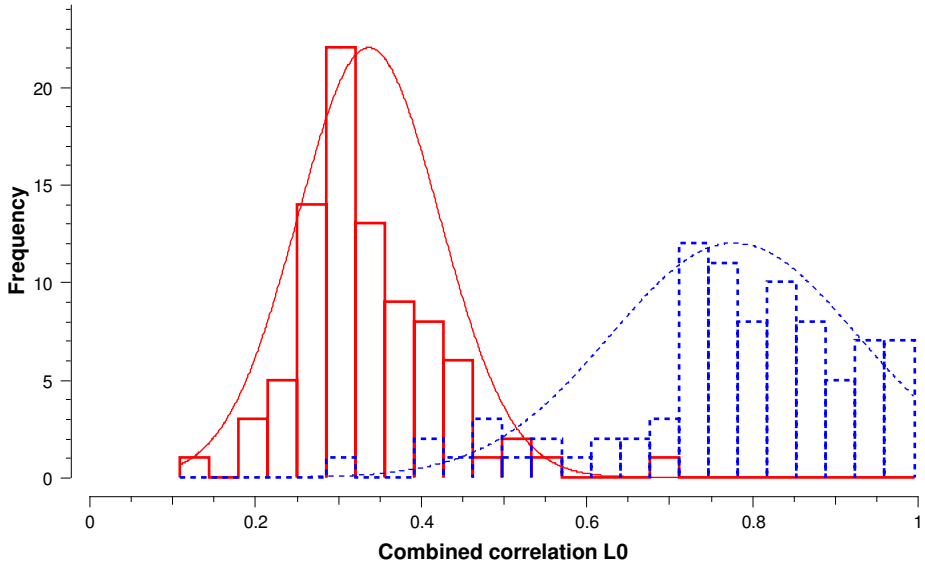


Figure 5.9: Distribution of the *combined correlation* measure for a set of 86 correct (blue dotted line) and 86 wrong (red solid line) associations

$$Chain\ codes = \begin{matrix} 3 & 2 & 1 \\ 4 & & 0 \\ 5 & 6 & 7 \end{matrix}$$

Then the correlation between the chains is calculated as follows:

$$ContourCorr = \sum_{i1=0,i2=0}^{N1,N2} \frac{7 - |C1_{i1} - C2_{i2}|}{7}$$

Where $C1$ and $C2$ are the values of the chain codes and $|C1_{i1} - C2_{i2}|$ is adjusted to be always ≤ 4.

Experiments showed that the *image correlation* itself provides good results for landmarks association. However, combination of it with the other two measures confers further improvement of the results. The *combined correlation* is calculated as a multiplication of all three measures. If the *image correlation* between the landmarks is less than 0.85, it is assumed that the other measures may be not reliable, so the value of $Corr$ is reduced by 0.85/0.65 before the multiplication (the value 0.65 was chosen because it requires both the $ShapeCorr$ and the $ContourCorr$ to be greater than 0.96 to pass the test for a correct association). When a new camera image is obtained, for each landmark one object is selected, which provides the maximum value of the *combined correlation*: $CombinedCorr_{max}$. If
$CombinedCorr_{max} > 0.6$ the landmark is associated with this object.

The aforesaid algorithm for calculating the *combined correlation* and the threshold of the correct association were determined based on heuristics obtained from sample images. Then a manually created set of 86 correctly associated pairs of landmarks and

86 wrong associations was considered to show its performance for the separation of the correct and incorrect associations (see Figure. 5.9). The *combined correlation* increases the value of Hausdorff distance [97] for the selected set of samples from 0.07 (for *image correlation* only) to 0.12.

5.4.3 Experimental Results

Figure 5.10 shows the performance of the algorithm during robot movement: linear movement (Figure 5.10(a)) and rotation (Figure 5.10(b) and 5.10(c)). The relative displacement of the robot during the motions is 20cm (the maximum length of the robot step) and the images from the camera were taken every second. The quality of the images presented in Figure 5.10 is different due to the changes in camera performance and light conditions. However, the algorithm does not require any specific adjustment for each case.

The presented results were obtained without assuming any constrains for the robot movement. If such constrains are applied, it can be seen that almost all wrong associations (showed in red in Figure 5.10) can be eliminated.

5.5 Fusion of Odometry and Vision System

The Kalman filter is widely used in case of sensor fusion for mobile robot localization [104, 105, 99, 98]. There are basically two approaches for using the Kalman filter for sensor fusion: state-vector fusion and measurement fusion [104]. In the first case the state is tracked by every sensor separately and then the state estimates are fused using their covariance and cross covariance matrixes. The measurement fusion combines data from different sensors when the measurement vector is formed. One approach can be to obtain a weighted average of the measurements from different sensors and then to track this combined measurement by the Kalman filter. In this case the sensors should provide the measurements of the same parameter. Another approach is to form an augmented measurement vector by merging the measurement vectors obtained from each sensor. The last strategy allows a greater flexibility and does not require the sensors to provide measurements of all parameters. Thus it is used in this work as the base strategy.

The state vector to estimate determines the 2D position of the robot

$$x = \begin{bmatrix} robotx \\ roboty \\ \alpha \end{bmatrix}$$

There are three sensors which provide the measurement vectors:

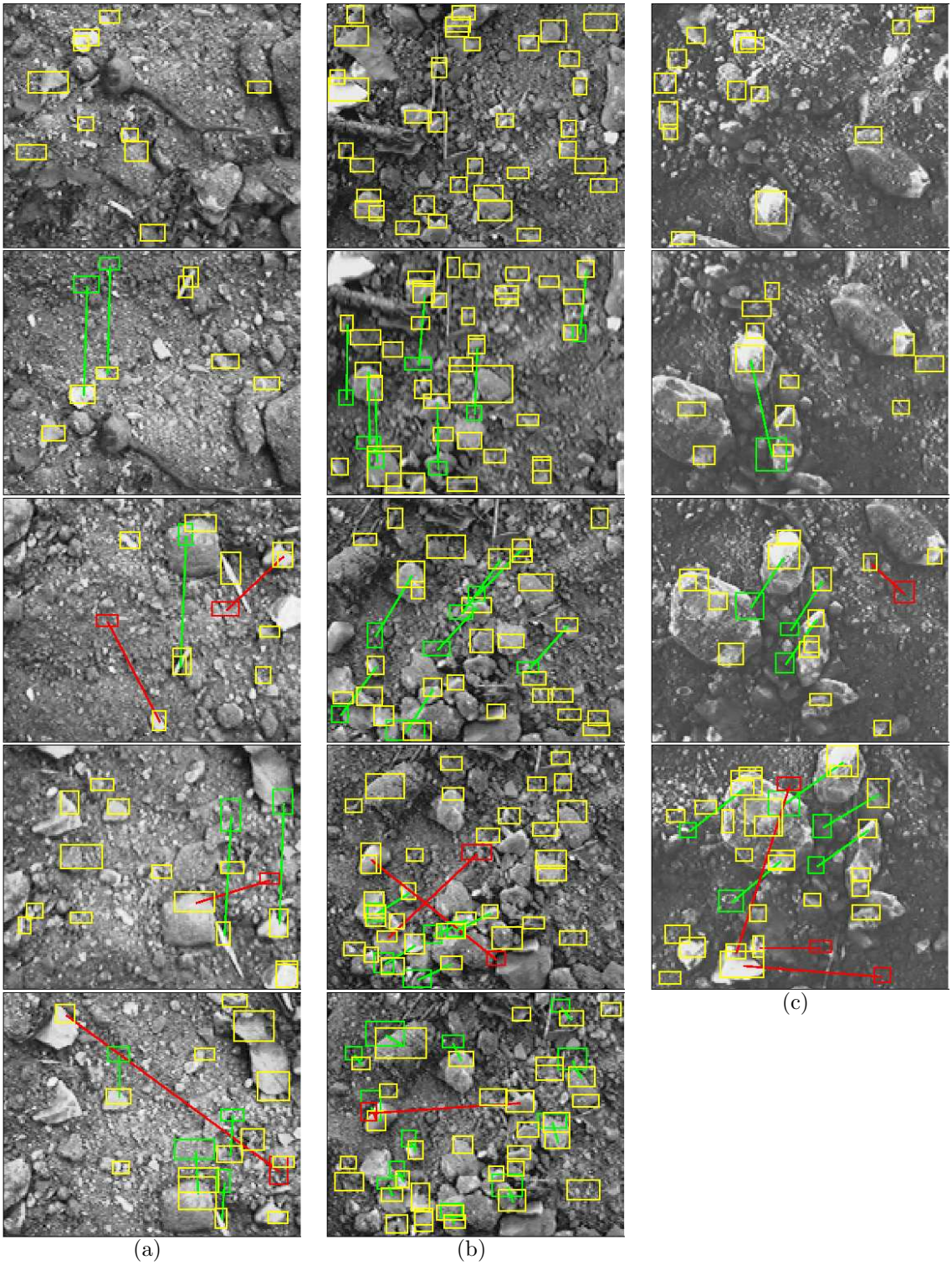


Figure 5.10: Detection and association of landmarks during linear movement (a) and rotation (b) and (c), yellow - landmarks detected in the current frame, green - landmarks detected in the previous frame and correctly associated with the new ones, red - wrong association, green and red lines connect the associated landmarks

- Odometry of the robot provides estimation of relative displacement along X and Y axes, the measurement vector is formed as
  $z_k^1 = \begin{bmatrix} robotx_{k-1} + \Delta robotx_k^{odometry} \\ roboty_{k-1} + \Delta roboty_k^{odometry} \end{bmatrix}$
- Vision system provides estimation of relative displacement along X and Y axes and rotation angle
  $z_k^2 = \begin{bmatrix} robotx_{k-1} + \Delta robotx_k^{vision} \\ roboty_{k-1} + \Delta roboty_k^{vision} \\ \alpha_{k-1} + \Delta\alpha_k^{vision} \end{bmatrix}$
- Electronic compass provides estimation of the heading
  $z_k^3 = \begin{bmatrix} \alpha_k^{compass} \end{bmatrix}$

According to the measurement fusion concept the measurement vector is formed as follow

$$z_k = \begin{bmatrix} z_k^1 \\ z_k^2 \\ z_k^3 \end{bmatrix}$$

The measurement model is represented by equation

$$z_k = Hx_k + \upsilon,$$

where υ is the measurement noise and H is the matrix:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

The system model is determined as follows:

$$x_k = x_{k-1} + u_{k-1} + \omega,$$

where ω is the system noise and u is the control input which is updated according to the current movement of the robot as follows:

$$u_k = \begin{bmatrix} \Delta robotx_k^{control} \\ \Delta roboty_k^{control} \\ \Delta\alpha_k^{control} \end{bmatrix}$$

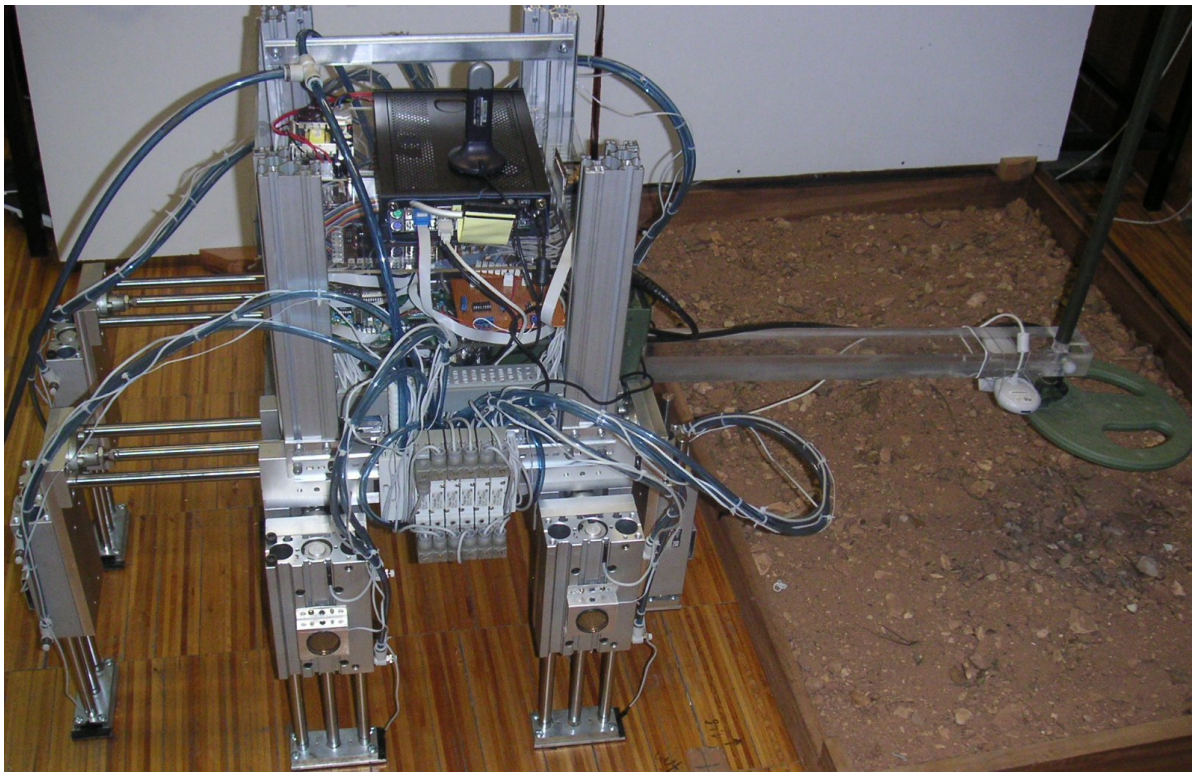


Figure 5.11: Landmine detection platform on the test setup

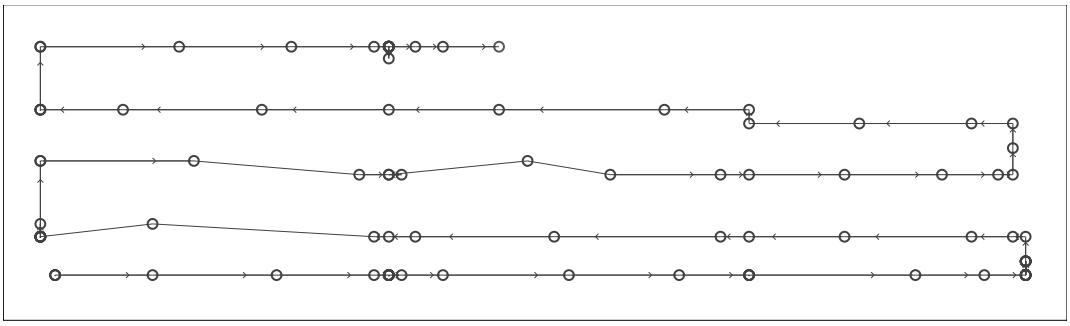
An important problem to be solved in the case of sensor fusion using Kalman filter is the determination of the process and measurement noise. In the case of the measurement noise, it can be done experimentally. The process noise is obtained based on the current state of the robot which determines the probability of failure for the considered sensors and the system model. The following conditions are considered:

- Starting or ending of the movement: the probability of slippage is high, thus the system noise is high
- The robot is moving or stopped: the system noise is low

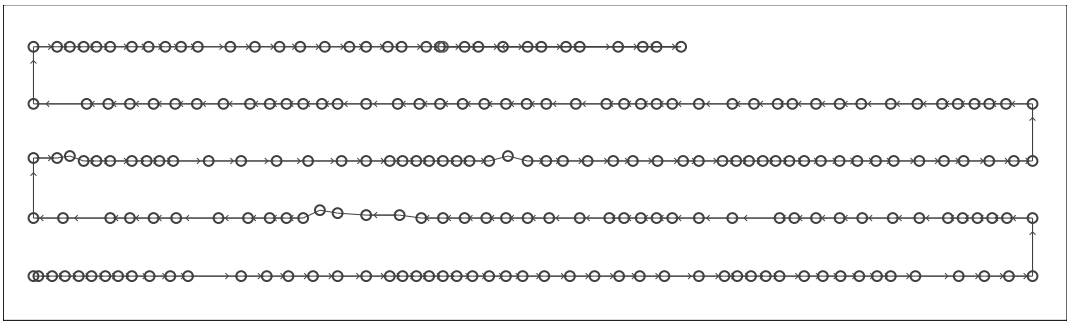
The sensors used in this work provide measurements at different time intervals due to the specificities of the hardware. When the sensor value is not available, it is ignored by setting the appropriate measurement noise covariance to infinity, forcing the filter not to consider the data at this time. The same process is performed for the vision system if there are not enough landmarks for determining the robot displacement.

5.6 Experimental Results

An experimental testing of the proposed system was done in an indoor environment using a box with soil to provide the input information for the vision system (Figure 5.11).



(a)



(b)

Figure 5.12: Trajectories obtained by the robot during the experimental test: a) - "true" trajectory measured by an external camera, b) - trajectory obtained by the developed system

During the tests the robot covered an area of 600x140 mm^2 with back-and-forth motions. The true position of the robot was monitored by an external camera mounted above the test side using an easy to detect color mark on the robot. The trajectories obtained by the proposed system and external camera are shown in Figure 5.12.

5.7 Implementation

The proposed low-level algorithms for robot positioning are implemented in the *low-level control program* which runs on the *main control board* of the robot.

The additional vision-based system is implemented in the *control program* which runs on the onboard PC of the robot. The Kalman filter sensor fusion is implemented as a *decision maker*. The odometry and vision systems are implemented as *sensors*. The detection of natural landmarks obtained from the video camera was employed for the suspicious objects detection, so here, the already developed class `CGridROIExtractor` is used as a base class. Please refer to Appendix G for more details.

5.8 Summary

Several possibilities to improve the positioning of the mobile scanning platform, from low-level control algorithms to a vision-based system, were investigated in this chapter. The combined system of vision and odometry has an ability to detect the deformations of the robot path, like slippage. Unfortunately, this approach cannot be adequately applied on the current platform prototype, since correction of small errors is not possible: the discrete cylinders cannot provide precise positioning and pure rotations are not possible in principle. The largest problem is originated from the small rotations of the robot body which introduce a constantly increasing error to its trajectory. However, the improvement of low-level control allows to minimize these situations, although not eliminating them completely.

If an absolute localization system is available it can be used in complementary to the developed system by means of data fusion based on Kalman filter. However, it can be concluded that the problems resulted from small distortions of the robot path cannot be completely eliminated, because even being detected they cannot be corrected. To allow the corrections, the mechanical structure of the platform has to be substituted by a more appropriate design as suggested, for example, in Section 8.3.

Chapter 6

Unknown Area Coverage

In this chapter the task of a high-level path planning for the mobile scanning platform is considered. This task is known as *unknown area coverage*. After a review of prior art in Section 6.1, the problem statement is formulated in Section 6.2. The developed approach is presented in general in Section 6.3. The basic elements which compose the algoithm are described in detail in Section 6.4. Section 6.5 is dedicated solely to the sensor processing required by the execution of the algorithm. Then, it is followed by some details about its implementation (Section 6.6) and the simulation and experimental results (Sections 6.7 and 6.8).

6.1 State of the art

The task of unknown area coverage is a well-known challenge for mobile robotics. Area coverage is a stategy which should lead the robot to visit every point of an area at least once. In contrast to the search strategy, where the visiting of all the area is not required to complete the test, here it is the most important condition. It is considered in general that the robot is not given with any information about the configuration of the space which should be covered. There are two basic approaches which can be used: *approximate cellular decomposition* when the area is represented by a grid map, and *exact cellular decomposition* where the area is represented as a set of simple subareas. There are also other strategies, for example, the concept of neural networks can be used to plan the path [106, 107, 108].

Examples of using approximate cellular decomposition with rectangular grid cells for online coverage can be found in [15, 109, 110]. A common specificity of such algorithms is a high number of turns in the provided path which makes the robot trajectory not regular, and therefore not suitable for landmine recognition.

The exact cellular decomposition [111, 112, 20, 19, 18, 113] seems a more appropriate strategy for a demining robot because it can produce an optimized path in terms of reg-

ularity. Choset et al. propose an algorithm to perform an exact cellular decomposition of the free area using a *virtual line* (also called *slicing line*) that moves across the map splitting it into smaller areas [113]. Cells are formed when the line finds an event (i.e. when it makes a tangent with the bound of an obstacle which can be detected by range sensors). Then each cell is covered using back-and-forth motions. For the online implementation of this algorithm it is important to find all points on the obstacle bounds where the cells must be formed. Acar et al. propose two ways to detect these points using robot range sensors [19]. There are also similar approaches which, however, do not use the exact cellular decomposition directly. Hert et al. suggested an algorithm which incrementally explores the area and covers it using back-and-forth motions [114]. Here the environment is represented as a combination of inlets and islands which are investigated during the exploration. Wong and MacDonald proposed a topological coverage algorithm which is based on a topological map representation of the environment and landmarks sensed on the obstacle boundaries [115].

Most of the existing unknown area coverage algorithms heavily rely on a high precision of the robot localization. Considering both problems this becomes a problem of SLAM. SLAM solutions are still not reliable enough for a large and almost empty environment ([116]), which is a typical environment for a minefield. In most cases, the coverage algorithm has to consider the odometry of the robot as the main source for localization dealing with its unreliability leading to algorithm failures [19].

6.2 Problem Statement

Path planning algorithm should work for two main purposes: assisting the data gathering for landmine detection algorithms and visiting all the points of the specified area, thus providing it's complete coverage. First goal imposes a restriction on the shape of the robot path. It should mimic a path followed by a stationary scanning device. Then, the sensor data can be properly mapped enabling the landmine detection. In order to guaantee the safety of the robot any non-scanning trajectory (any free path between two points) should be always located inside the already covered area. Such intermediate paths should be as short as possible to prevent accumulation of large localization errors which again lower the safety of the robot and affect the landmine detection. Such requirements were not considered in the previous works. However, they can improve the performance of the algorithm itself making it less dependent on the localization. Instead of assuming that a source of global localization is available, the path should be planned relying as much as possible on the environmental features, as implyed by SLAM strategies, increasing the reliability of the algorithm.

6.3 Approach

The developed coverage algorithm has a hybrid behavior-based architecture: it is based on a situated activity design which incorporates simple reactive behaviors [117]. The robot behaviors which compose the algorithm also serve as building blocks of the associative memory where the positions of obstacles are recorded. Instead of memorizing coordinates of a point, a sequence of several basic behaviors required to arrive to that point from the current position is recorded. In the case of a coverage algorithm the amount of possible situations is relatively small allowing a practical implementation of the proposed strategy. Thus, the algorithm operates with situations and sequences of behaviors needed to navigate between them, avoiding the use of global positioning information. The only positioning data which is considered to be available globally is the directional information (heading of the robot). However, this information, in any case, is required to maintain a regular coverage pattern.

To fulfill the requirements for the coverage algorithm, the most suitable strategy seems to be the exact cellular decomposition when the area is divided into cells of different sizes and shapes in such a way that the total area of the cells is equal to the initial area. This allows to implement a regular coverage pattern, back-and-forth motion, and simplifies the planning. The approach developed in this work is inspired by [113] and [114].

Being based on the cellular decomposition the proposed coverage algorithm should assure the coverage of all cells in which the environment is decomposed. There is always one cell, called *current cell*, being covered at a given moment. While proceeding with the coverage of the *current cell* the environment is observed in order to reveal conditions for further decomposition. This process provides a set of cells and paths between them which are executed when the coverage of one cell is finished and another one is chosen.

6.4 Basic Elements of the Coverage Algorithm

Figure 6.2 shows a general structure of the algorithm. The algorithm is based of several simple behaviors which are used in all different situations. These behaviors, in turn, compose more complex ones which provide the ability to sense the critical points and deadlocks. Combinations of simple and complex behaviors provide the required functionality which is logically devided in three processes: area decomposition, cell coverage and path between cells. These processes are even more complex basic elements which finaly compose the algoithm as shown in Figure 6.2.

The basic elements, starting from the premitive behaviours, are described in more details below.

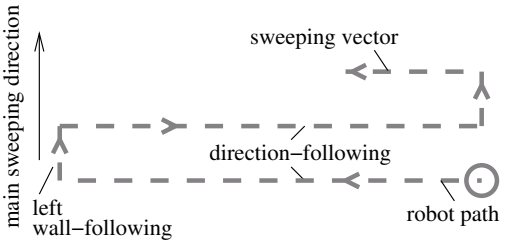


Figure 6.1: Basic coverage pattern

Behaviors

There are two basic behaviors used by the algorithm:

- *Direction-following* is a forward movement of the robot in a specified direction. The robot direction is constantly corrected by a feedback from the heading sensor of the robot. The behavior is terminated if an obstacle is encountered on the way.

- *Wall-following* is an adaptive movement of the robot at a safe distance along the boundary of the closest obstacle. The distance to the obstacle is measured by the robot sensors and used to control the robot actuators in order to keep it constant. There are two types of *wall-following* motion defined by the side of the robot which is faced to the obstacle (*wall-following side*): *left* or *right*.

The basic coverage pattern (back-and-forth path) is performed by switching between the *direction-following* and the *wall-following* behaviors (Figure 6.1):

Direction-following along the *sweeping vector* until an obstacle is encountered

⇕

Wall-following until an intersection with the *sweeping vector*

When the behaviors are switched, the *sweeping vector* and the *wall-following side* are corrected appropriately. This procedure implies the using of coordinate information (while moving the *sweeping vector*). However, this information is used only locally.

The associative memory of obstacle positions is organized using the following modifications of the basic behaviors:

- *Directed wall-following* is a *wall-following* executed until a specified direction is empty from obstacles, which signifies that a *direction-following* can be started in this direction.

- *Find obstacle* behavior is a combination of the two basic behaviors which is used to reach the specified obstacle. The "location" of the obstacle is defined by its number from the current position of the robot and by the direction in which the *direction-following* should be performed in order to discover it. Intermediate obstacles are overtaken by the *wall-following*.

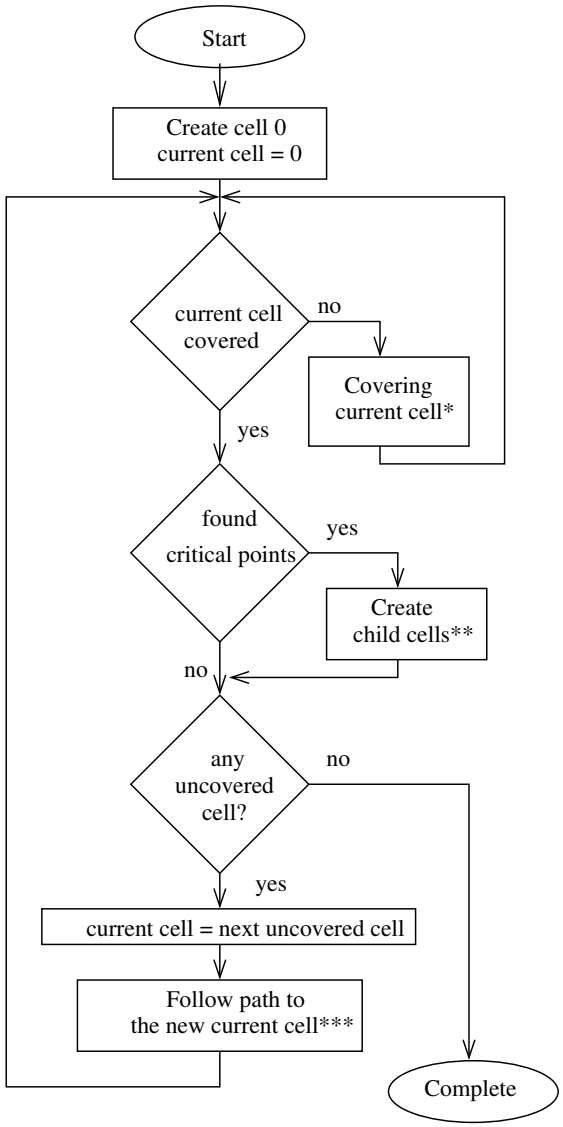


Figure 6.2: General structure of the coverage algorithm. * and ** - Section 6.4.2, *** - Section 6.4.3

These behaviors are used to guide the robot between the cells of the decomposition.

Critical Point Sensing

A *critical point* is a part of the obstacle which does not allow the robot to complete the coverage by simple back-and-forth paths (Figure 6.3). At this point the area has to be split into two cells so that each of them can be covered separately (for a strict mathematical explanation see, for example, [18]). A *critical point* is called *forward* if it is located in the *main sweeping direction* (Figure 6.1), and *backward* otherwise (Figure 6.3 shows an example of a *forward critical point*).

Range sensors of the robot (sonars) are utilized in this work to detect the *critical points*. For a *forward critical point*, the sensing process is started during the *direction-*

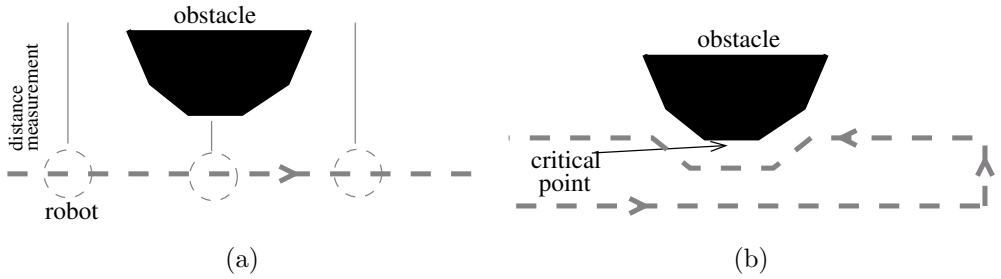


Figure 6.3: *Forward critical point* sensing: (a) sensing the distance during the *direction-following*, (b) confirming the *critical point* during the *wall-following*

following (Figure 6.3(a)) by measuring the range to the obstacle in the *main sweeping direction*. A local minimum of distance along the robot path signifies the presence of a *critical point*. However, the distance information is not very reliable and, thus, the *critical point* is said to be found only when it is *followed* by the robot during the consequent *wall-following* (when the robot returns to the previous *sweeping vector* as shown in Figure 6.3(b)). The information obtained from the range sensors during the *direction-following* is utilized in order to determine the *wall-following side* when the obstacle is encountered: the *side*s are opposite for the border and a *forward critical point*, and they are the same for the border and a *backward critical point*. Sensing of a *backward critical point* is similar considering the whole process in the direction opposite to the *main sweeping direction*.

Deadlock Sensing

An obstacle bounding the area in the *main sweeping direction* provides a situation when a back-and-forth path of the robot comes to a deadlock. The deadlock is sensed in the same way as a *critical point*: it may occur if the robot returns to the previous *sweeping vector* during the *wall-following*. In this situation, the direction of the robot displacement during the *wall-following* is analysed in order to distinguish between a deadlock and a *critical point* (a direction opposite to the direction of the *sweeping vector* signifies the deadlock). If a dealock is sensed the covering of the current sell is finished, however, no new cells are created (as it would happen in the case of a *critical point*)

6.4.1 Area Decomposition

Every *critical point* encountered as described in Section 6.4 signifies a discontinuity of the environment. Thus, the decomposition is performed by creating two new cells for each found critical point (in a general case, the number of created cells is equal to $n+1$, where n is the number of *critical points* found on the same *sweeping vector*). After that, the *current cell* is considered to be covered and the next cell is chosen (Figure 6.2). The

created cells are considered to be *child* cells and the cell which contains the *critical point* - *parent* cell. The direction of the *critical point* defines the direction of the created cell (*forward* or *backward*). *Forward* and *backward* cells have the opposite *main sweeping direction*.

This process leads to a decomposition structured as a tree (see simulation example in the end of this section and Figure 6.7(c)). The cells are chosen for coverage in the order in which they are created and the tree structure is used for the planning of the paths between the cells. The process of creation of the new cells is controlled by several logical assumptions which do not allow to double-cover the same area in the case of a pair of *backward* and *forward* cells which represent the same part of the environment.

6.4.2 Cell Coverage

The coverage of each cell is performed by the back-and-forth path as shown in Figure 6.4. One of the following situations completes the cell coverage and may result in the creation of new cells:

- *forward critical point* is found ⇒ new *forward* cells are created
- *backward critical point* is found ⇒ new *backward* cells are created
- deadlock ⇒ no new cells are created

The *critical point* which caused the cell creation is called a *starting point*. The coverage of the cell is started from the *starting point* by a *directed wall-following*. The parameters of the latter (the *wall-following side* and the direction) are made unique for all created *child* cells when the coverage of the *parent* cell is completed in one of the cases mentioned above. For example, if the coverage of the current cell is completed due to the descovery of a *forward critical point*, two new cells will be created, one with right *wall-follwing side* and another - with left.

6.4.3 Path to the Next Cell

The path between two cells of the decomposition is based on the idea of associative "location" of *critical points* relatively to each other. The paths are updated every time the *current cell* is covered: the path from the cell ending situation to its *starting point* is added to the end of the path. The *directed wall-following* is used to navigate around the same obstacle, and the *find obstacle* behaviour is used to move between the neighboring obstacles.

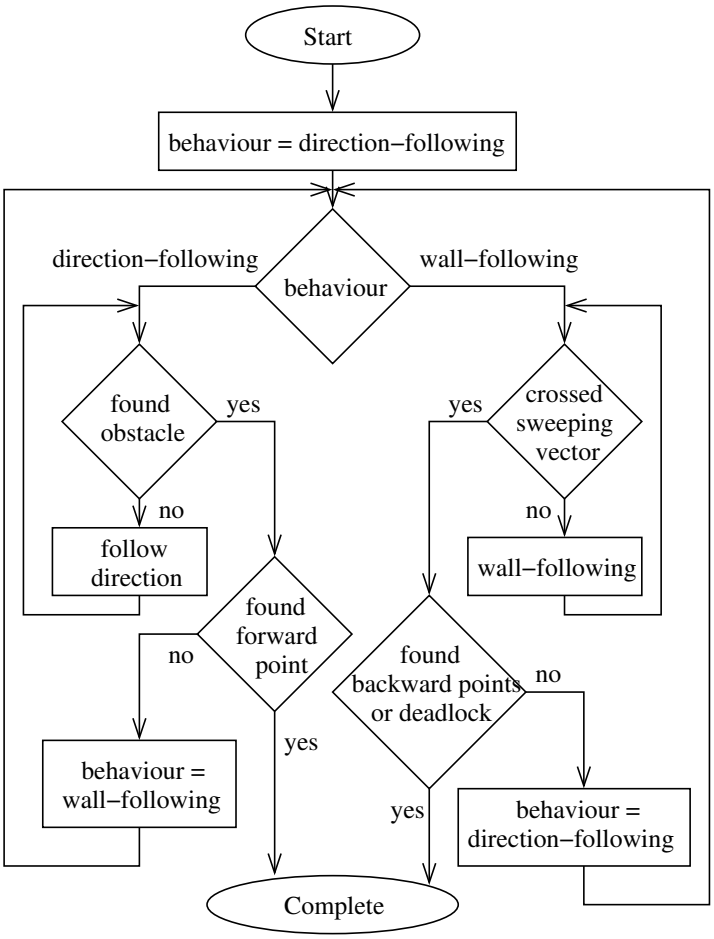


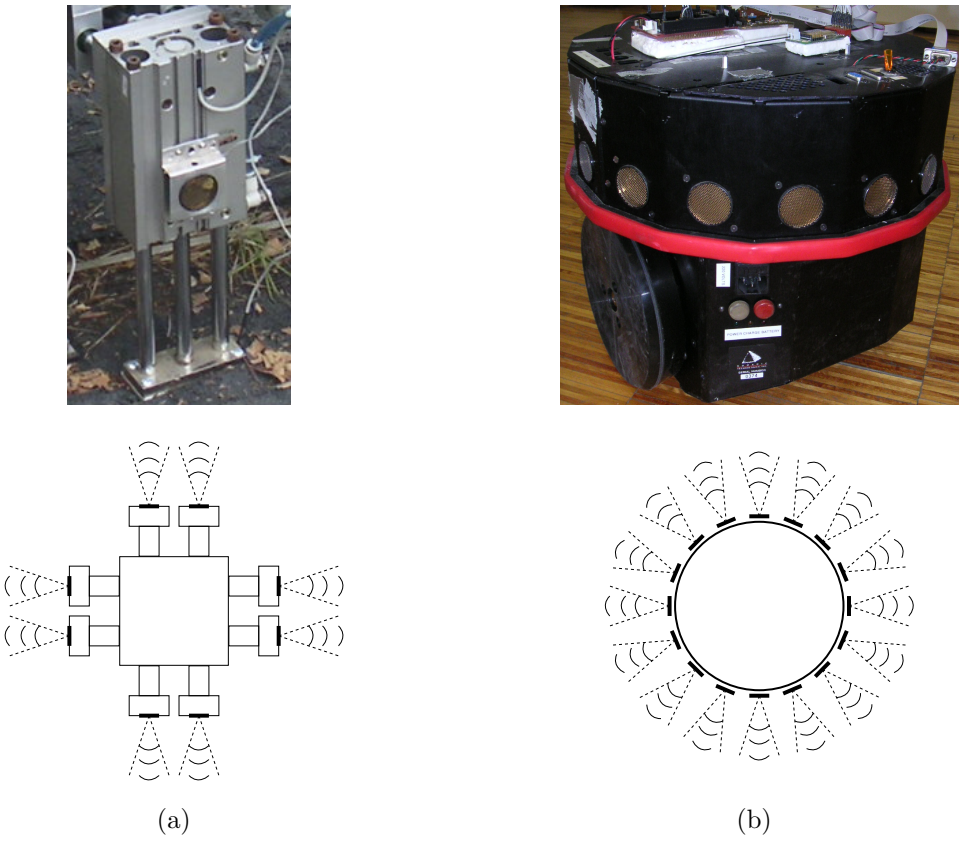
Figure 6.4: Cell coverage

6.5 Sensor Processing

The developed coverage algorithm relies on the sensor data as a primary source of information about the environment (in contrast to a map-based approach which would rely on a predefined map). It implies the existence of range sensors which report the distance to the nearest obstacles from the robot in different directions. In this work the sonars, usually used for mobile robots, were considered for this purpose. The Nomad Super Scout robot is equipped with 16 sonars which provide 360° field of view as shown in Figure 6.5. On the mobile scanning platform one sonar is installed on each leg (see Figure 6.5) providing the view in four directions. Sonars confer acceptable performance being relatively cheap. However, there are a number of possible failures which commonly occur in the system of several sonars:

- A *crosstalk* occurs when the beam emitted by one sonar is received by another one. In this case the value obtained by the second sonar is unpredictable.

- *Multiple paths* situation happens when the beam is reflected from an obstacle at a high angle, so it does not arrive back to the sonar. Instead, the beam is



(a) (b)

Figure 6.5: Location of sonars on the mobile scanning platform (a) and Nomad Super Scout (b)

reflected from other surfaces and finally arrives back with a large delay leading to overestimated value of distance.

One of the possible strategies which helps to reduce the influence of the mentioned situations is the mapping of the sensor reading to a grid-map, called *occupancy grid* [118]. Experiments performed in this work showed that this approach confers significant reduction of the sonar errors.

A probabilistic approach is used in this work to form the *occupancy grid*. A probability of being occupied by an obstacle is assigned to every grid cell. Every time a new sensor reading is available it is analyzed using a simplified model of the sonar beam as shown in Figure 6.6. Each grid cell which falls into the cone of the beam (shown in dark gray) is updated according to the following rule:

$$p_k = \frac{\widetilde{p_k} p_{k-1}}{\widetilde{p_k} p_{k-1} + (1 - \widetilde{p_k})(1 - p_{k-1})}, \tag{6.1}$$

where p_k is the probability of the cell to be occupied at time moment k, $\widetilde{p_k}$ is the probability of the cell to be occupied given the current sensor reading. Probability $\widetilde{p_k}$ is assigned according to the area in which the cell is located: 1, 2 or 3 in Figure 6.6:

1. $\widetilde{p_k} = 0.4$ (most probable these cells are not occupied)

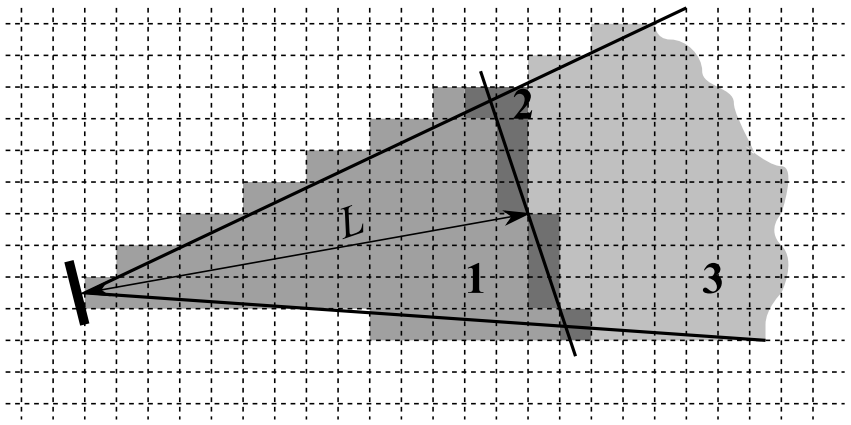


Figure 6.6: Updating of the occupancy grid from a sonar reading, L - the value of distance reported by the sonar (the shortest distance to the target surface inside the cone of measurements)

2. $\widetilde{p_k} = 0.45\frac{L_{max}-L}{L_{max}} + 0.5$ (most probable the cells are occupied but the reliability of this assumption decreases with distance)

3. $\widetilde{p_k} = 0.5$ (the state of these cells is unknown)

The obtained occupancy grid is used for the detection of *critical points* when the *direction following* is finished. In this case there is enough data obtained during the *direction following* which are constantly incorporated into the grid providing a reliable map.

It should be mentioned that the *wall following* behavior is performed based on the raw sonar data in order to reduce the computation time. Still the performance of the *wall following* is acceptable due to the reactive nature of the behavior.

6.6 Implementation

The coverage algorithm is implemented in the *control program* by the class `CExplorationBeh` representing a *behavior* in the object model described in Chapter 2. It contains the basic behaviors, as shown in Section 6.4, and performs by switching between them. All behaviors are implemented also as *behavior* objects, so they are nested by the main behavior :

- *Wall-following* is implemented by the class `CWallMotion`
- *Directed wall-following* and *Direction-following* are implemented by the class CForwardBeh
- *Find obstacle* behavior is implemented by the class `CFindObstacle`

Several objects are used to detect specific situations when the behavior should be switched. These objects are implemented as *decision makers*:

- The finishing of the current cell coverage are analyzed by `ACellEndDetector`, `AWallDirDetector` and `ACellEndAfterWallDetector`
- Critical points are detected by `COccupancyGrid` class (this class also peforms the processing of the occupancy grids)

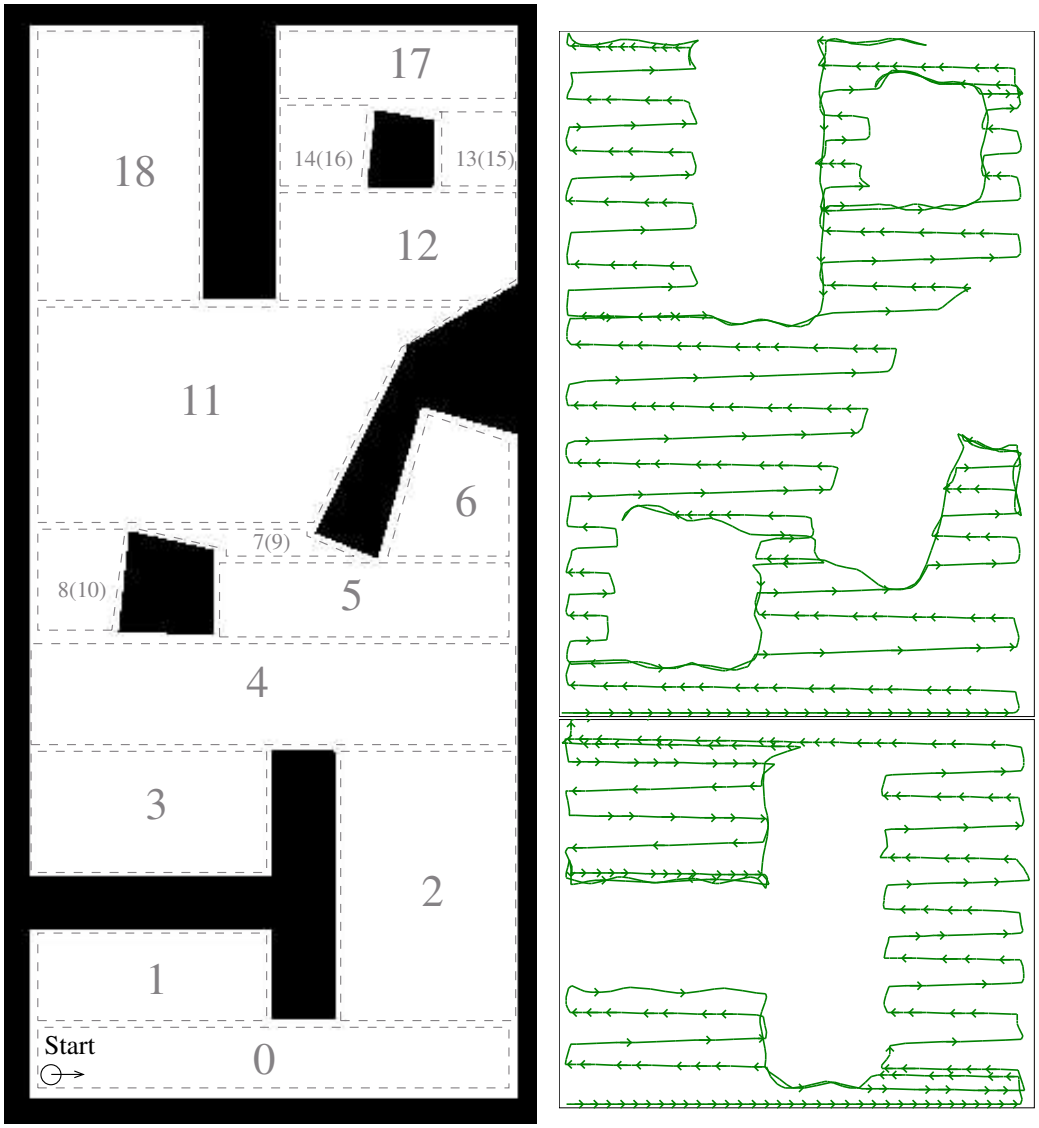
The exact cellular decomposition which is constantly constructed by the behavior is implemented as a *map* object by `CAssociatedED` class. Finally, the top-level behavior controls the robot using a hardware-independent interface so the algorithm can run both on the Nomad Super Scout and on the demining robot prototype without changes. The robots are represented by `CTwoWheelsRobot` and `CDemineRobot` classes respectively. These classes implement the necessary abstraction interface to allow the requied transparant transition from one platform to another. Please refer to Appendix H for more details.

6.7 Simulation Results

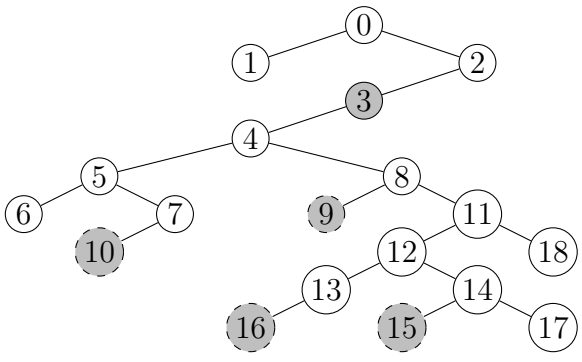
A simulation test was performed in order to evaluate the performance of the coverage algorithm. The used simulation environment is provided by Nomand to emulate the Super Scout mobile robots. Simulation performance of the robot is realistic. However, many problems cannot be tested with this simulation, like the accumulation of the odometry errors which is tested further in this chapter during the real experiments. Figure 6.7(a) shows the map of the simulated environment which was covered by the robot. Figure 6.7(c) presents the obtained decomposition tree (the numbers of cells are shown also in Figure 6.7(a)) and Figure 6.7(b) shows the trajectory followed by the robot during the coverage.

Using this example the following features of the coverage algorithm described earlier can be seen:

- Coverage of cells 1, 3, 6, 17 and 18 was finished by a deadlock
- Backward cells 9, 10, 15 and 16 were not covered because the corresponding forward cells 7, 8, 13 and 14 were successfully determined
- Cells 0, 4, 5, 11 and 12 each created two child forward cells



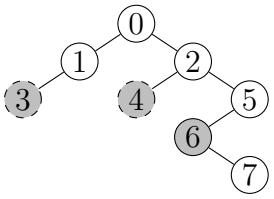
(a) (b)



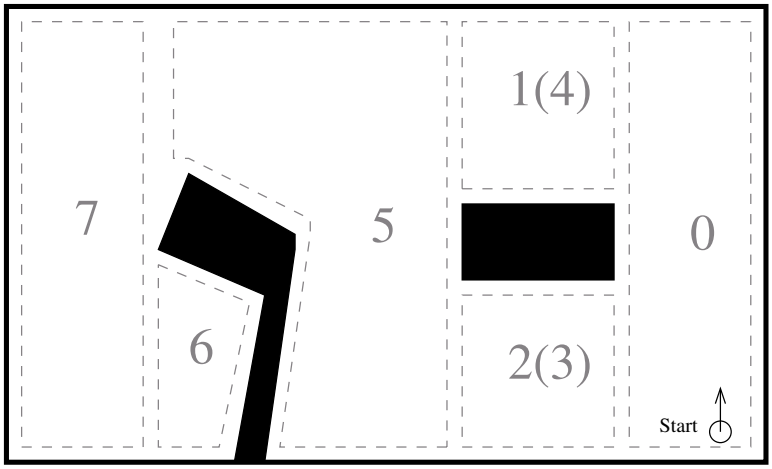
(c)

Figure 6.7: Simulation testing (a) simulation environment, the cell numbers of the created decomposition are shown in gray; (b) trajectory of the robot during the simulation test; (c) obtained decomposition tree (white - forward cell, gray - backward cell, dashed border - cell not covered due to a corresponding forward cell)

(a)



(c)



(b)

Figure 6.8: Experimental setup (a), (b) - map of the experimental setup (the obtained cell decomposition is shown in gray), (c) - decomposition tree created during the coverage (white - forward cell, gray - backward cell, dashed border - cell not covered due to a corresponding forward cell)

6.8 Experimental Results

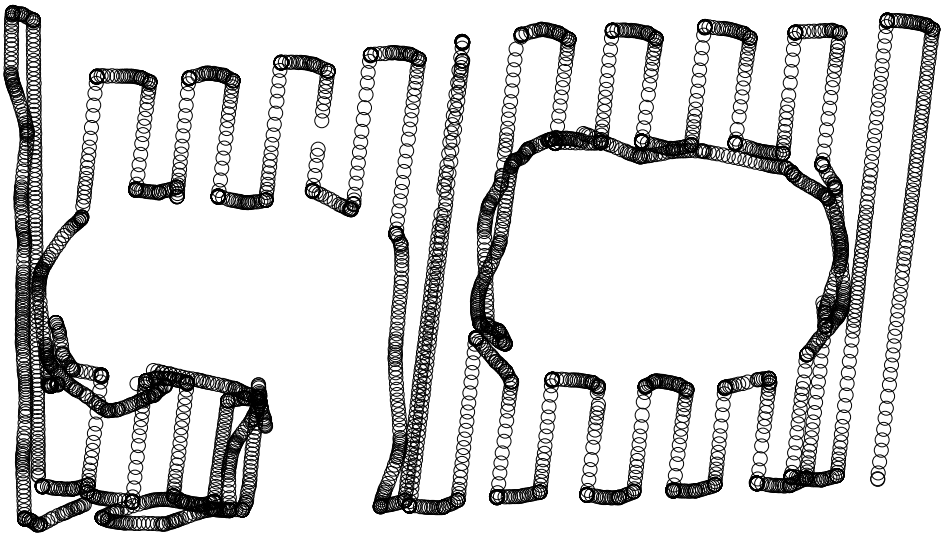
Indoor testing of the path planning algorithms using the demining robot prototype is complicated due to its large size so large areas are needed for testing. In turn, covering of the large area leads to the accumulation of the directional error due to unavoidable small rotations of the robot which does not allow to complete the path. Thus, experimental evaluation of the proposed algorithm was performed using only Nomad Super Scout mobile robot. The robot covered a bounded testing area with two obstacles shown in Figure 6.8(a) while creating a cellular decomposition presented in Figure 6.8 (b) and (c). A complete coverage was achieved. Figure 6.9 shows the trajectory recorded by

the robot odometry sensors (Figure 6.9(a) shows the trajectory corrected for odometry errors which is close to the real trajectory followed by the robot).

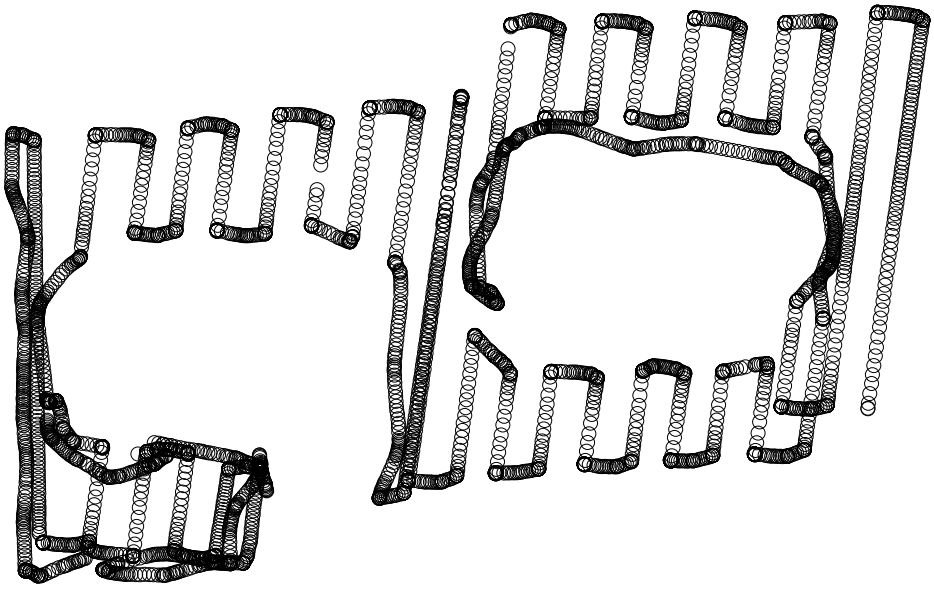
It can be seen from Figure 6.9(b) that during the coverage large odometry errors were accumulated. Such position errors would probably disable the possibility to perform a coordinate-based coverage. However, the proposed algorithm successfully covered the area. The measurement of direction in this experiment was obtained from the robot odometry which also accumulated the error. This error obviously affects the performance of the coverage because it does not allow a high-quality control of the direction for the *direction-following*. An absolute direction sensor might be installed on the robot as part of future work.

6.9 Summary

The algorithm for complete online coverage described in this chapter meets the requirements given by the landmine detection system: it utilizes a regular coverage pattern and provides a safe path. Because this task was considered in a broad sense, the algorithm also has a feature of low dependency on the robot localization which showed its reliability during the experiments. The usage of this strategy is not limited to demining only, but can be also employed in other fields of mobile robotics, like cleaning, grass cutting, agriculture, etc.



(a) Trajectory corrected for visible odometry errors



(b) Trajectory as recorded by odometry

Figure 6.9: The trajectory of the robot obtained during the experiment shown in Figure 6.8

Chapter 7

LADERO Experimental Testing

This chapter presents the results of the experimental testing of the mobile scanning platform LADERO performed on a test minefield in real conditions. The testing included the gathering of the sensor data from a test minefield and landmine detection data processing. Section 7.1 presents the details about the data acquisition process. Then, the obtained sensor data were processed by the landmine detection algorithms developed in this work as described in Section 7.2.

7.1 Data Acquisition

Having the landmine detection and path planning algorithms developed in this work, there is a question about their performance in real conditions when they are assisted by a mobile scanning platform. Therefor, the LADERO prototype with control software based on the developed algorithms was tested on a test minefield located at Meerdaal bomb disposal unit in Belgium. Figure 7.1 demonstrates the robot on the test field during the scanning.

The data were acquired from 27 objects using metal detector and two IR sensors installed on the platform. Figure 7.2 shows the layout of the test minefield. Each object on the field is located in a separated rectangle with size $50 \times 100cm$ and surrounded with an empty area to allow easy access. Not all of the available objects could be scanned by the robot because of its relatively large size. Each object was scanned in a separated experiment conferring reduction of the positioning errors which would be accumulated on a larger area. Examples of the data can be seen in Figures 7.3-7.5.

The metal detector used on the robot showed a good performance on the landmine targets located on the test field. It detected all low-metal plastic landmines buried in 0-15 cm depth, except one. Even with the relatively low quality scanning provided by the platform, allows us to obtain representative data from the metal detector due to the large size of the sensor head which compensates for the position inaccuracy of

Figure 7.1: Robot on the test field

the platform. The performance of the infrared sensors is acceptable only in few cases depending on the time of day and weather conditions (the tests were performed from 8 a.m. until 8 p.m.). Unfortunately, it is questionable if such sensors can be used on the platform because of the small area of the soil that they cover providing large dependence on the accuracy of the platform. It has to be concluded that the IR data are ambiguous due to the small spots provided by the sensors on the ground and, thus, only the data from metal detector were used for the classification.

The scanning performed by the robot is very unreliable in terms of slippage and small rotations. However, the possibility to use a similar platform for landmine detection in general can be tested using its current version. The testing requires constant assistance by a researcher in order to correct the small disturbances of the robot trajectory which cannot be corrected automatically due to the mechanical limitations. The speed of the robot is limited by the time required for sensor data acquisition. In its current version an average scan 100x50 cm requires about 40-50 minutes (this was an average time required for scanning one object on the Meerdaal test fields) due to the speed of *scanning cylinders* adjusted to be approximately $40mm/s$ in order to provide reliable motions as described in Section 5.2.

The data obtained during the tests are unique in a sense of being acquired by a mobile platform from two sensors at the same time in real conditions. The condition

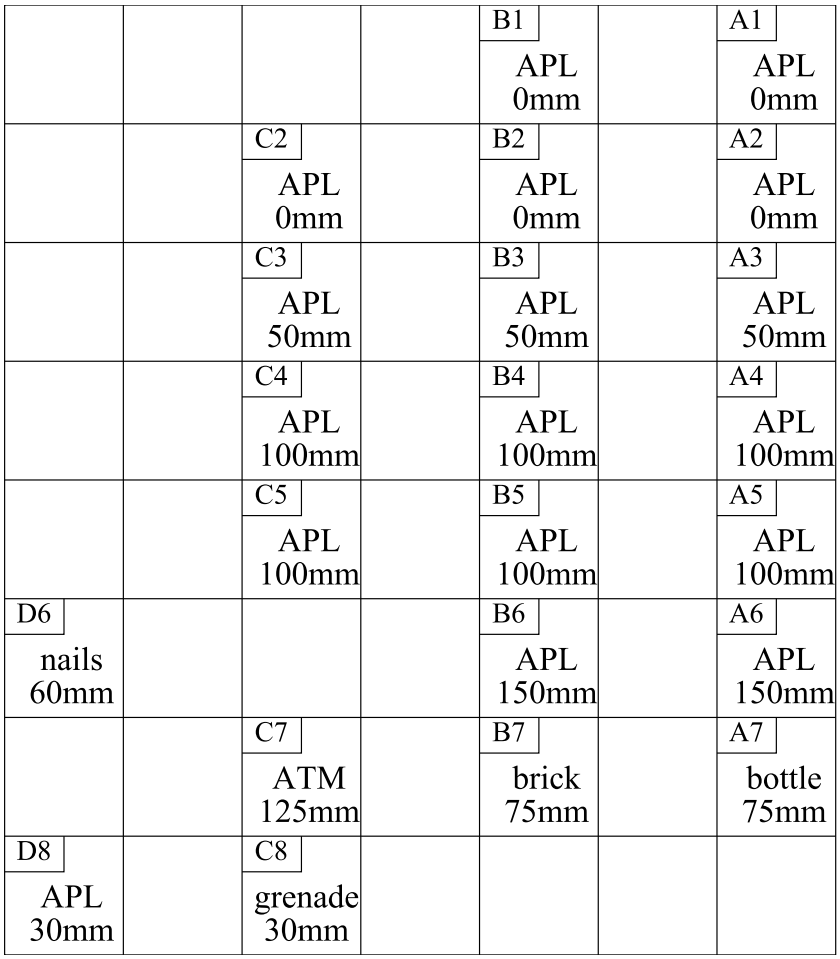
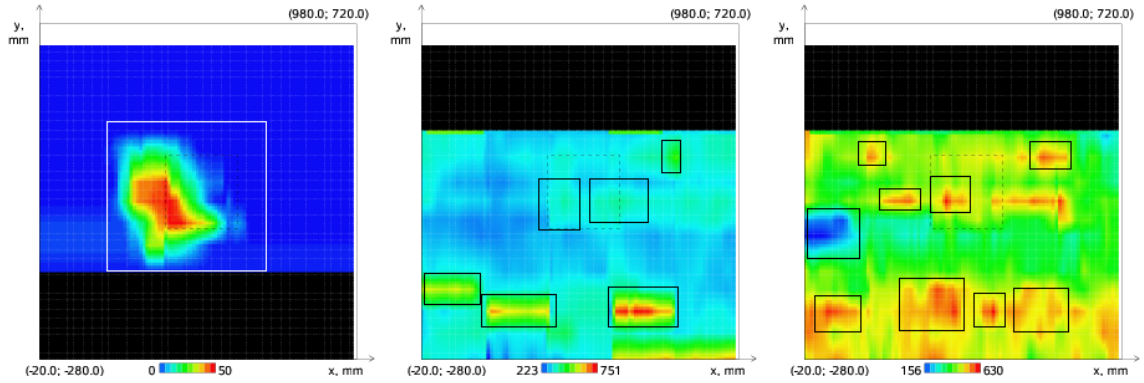


Figure 7.2: Layout of the test field (APL - antipersonnel landmine, AT - antitank landmine), sections An contain APL M35BG, sections Bn contain APL PMN-2, section C contains APL PRB409

of the field is close to natural because it was not used for a long time. The usage of these data represents a real ability of employing a mobile robot for landmine detection in contrast to the case when the data obtained by high precision scanning devices are used.

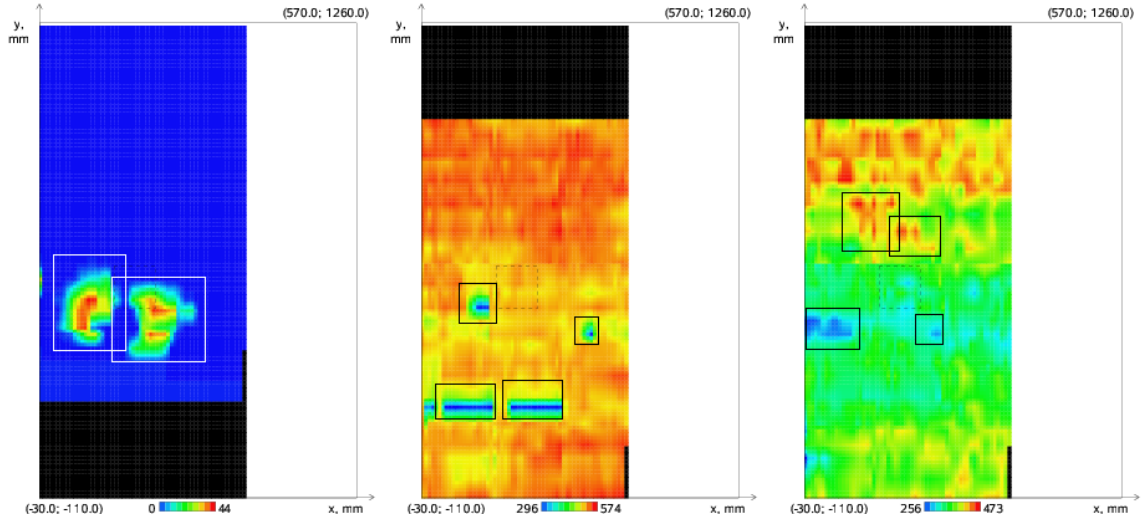
7.2 Landmine Detection

The data obtained by the mobile platform during scanning were used for landmine detection. Taking into account the low spatial resolution of the data it is reasonable to consider only a two-stage classification as shown in Figure 7.6. The first stage of the detection, namely the detection of suspicious objects, showed good results providing the detection of all objects. If the detection process is terminated after this stage (all the detected objects are considered to be landmines) the value of the false alarm rate would be 5.4 FA/m^2.



(a) Metal detector (b) IR0 (c) IR1

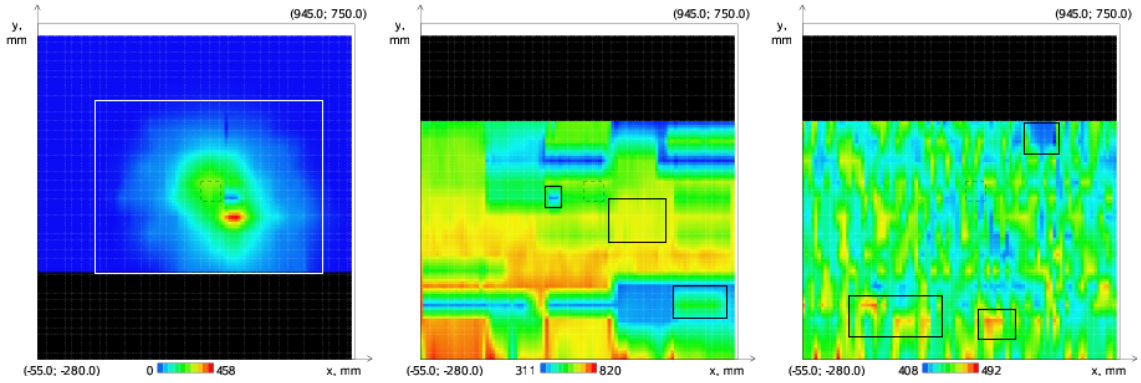
Figure 7.3: Antitank landmine VS 1.6, x = 500 mm, y = 250 mm, depth = 0 mm; IR0 - 6.6-14 μm, IR1 - 8-14 μm; the landmine is marked with a dashed rectangle



(a) Metal detector (b) IR0 (c) IR1

Figure 7.4: Antipersonnel landmine PMN-2; x = 250 mm, y = 500 mm, depth = 0 mm; IR0 - 6.6-14 μm, IR1 - 8-14 μm; the landmine is marked with a dashed rectangle

A possibility to perform further classification is tested using the concept of *dominant class*. This concept assures that the failure of the classification step does not lead to a significant change of the detection rate leaving the false alarm rate also unchanged. After the ROI detection stage for these experimental data the obtained data set has 70 objects, including the 27 real objects, several double detections and clutter. The size of this data set is obviously not enough for performing training and evaluation of a classifier. Therefore, it was decided to investigate a possibility to train the classifier using experimental data from [5]. This task is especially important because the situation of not having enough training data can easily happen in reality. The main obstacle to this approach is the fact that the sensors installed on LADERO are not considered in the MsMs database [5]. This means that the classifier has to be trained on the data



(a) Metal detector (b) IR0 (c) IR1

Figure 7.5: Antipersonnel landmine M35BG; x = 500 mm, y = 250 mm, depth = 100 mm; IR0 - 6.6-14 μm, IR1 - 8-14 μm; the landmine is marked with a dashed rectangle

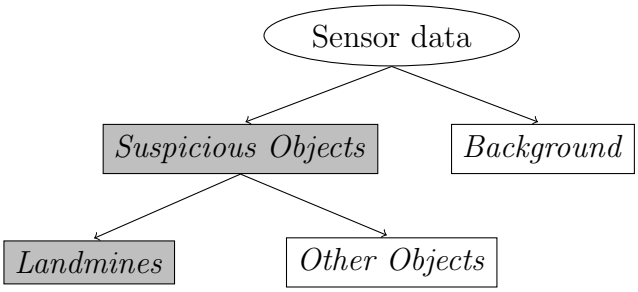


Figure 7.6: Two-step landmine detection performed for the obtained experimental data (classes highlighted with gray background can be associated with landmines)

obtained from different sensors.

The only possibility to perform such training is to reveal classification features which have weak dependency on the nature of the sensor. Such features could be fused for signatures of the same object obtained from different sensors using the idea of *combined features* introduced in Chapter 4. To choose the classification features which are general enough to be used for the created classifier a measure of *feature generality* in respect to the dominant class is introduced. This measure intends to estimate the difference between the distributions of the feature for the dominant class obtained from different sensors. The closer are the distributions of the same feature for different sensor to each other, the more general the feature is. Only the distributions for the dominant class are considered here allowing the feature to be not general in respect to the other class. The measure is calculated as follows:

$$FG = \frac{1}{N}\sum_{k=1}^{N}\frac{\Delta\mu_{ref} - \left|\mu_{comb(class1)} - \mu_{k(class1)}\right|}{\Delta\mu_{ref}} \tag{7.1}$$

$$\Delta\mu_{ref} = \left|\mu_{comb(class1)} - \mu_{comb(class2)}\right|, \tag{7.2}$$

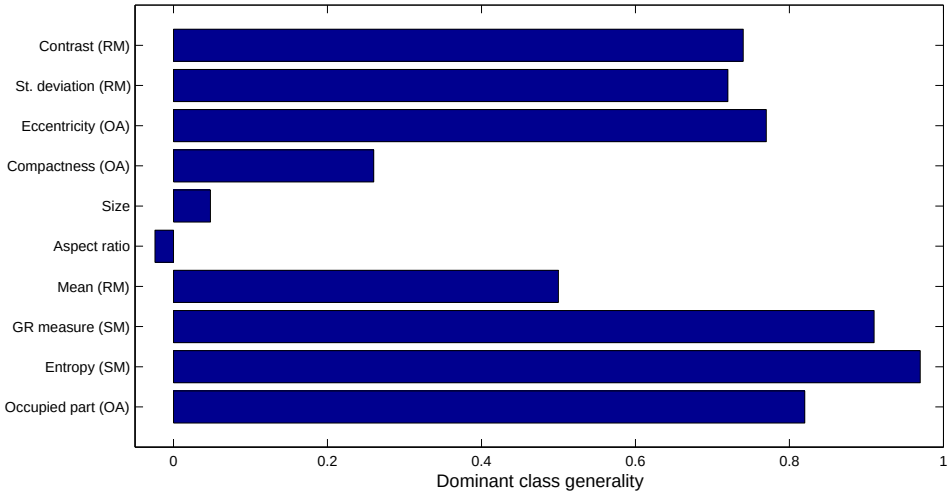


Figure 7.7: Value of *dominant class generality* for different classification features

Table 7.1: Landmine detection results over the stages of two-step detection process (the evaluation set contains 19 landmines)

	Classification step	
	ROIs extraction	Classification *Landmines/ Other Objects*
Landmines detected	19	18
DR	100%	95%
FA/m^2	5.4	2.9

where N is the number of sensors, $\mu_{comb(class1)}$ is the mean of the combined feature distribution for $class1$, $\mu_{k(class1)}$ is the mean of the feature distribution for $class1$ obtained by sensor k, and $class1$ is the dominant class. The measure shows how far the individual features calculated for each sensor are from the averaged feature normalized by the distance between the means of the classes (after processing the distributions to account for the dominant class). It should change from 0 to 1, but can also be negative showing a very high nongenerality. Only the features which have the value of FG close to 1 will be considered for the construction of the classifier.

Figure 7.7 shows a comparison of the classification features in terms of *feature generality* in respect to the dominant class. Now three features with highest values of FG are chosen:

- combined Occupied part (OA)
- combined Entropy (SM)
- combined GR measure (SM)

The created classifier is then trained on the database experimental data from [5] using the largest available data set which includes data from four sensors and counts

2574 samples. The evaluation of the classifier is performed on the data obtained by LADERO. After the classification one of the previously detected landmines is missed and the FA/m^2 is reduced from 5.4 to 2.9. The results of landmine detection through the classification stages are shown in Table 7.1.

The achieved possibility to employ a classifier trained on the data obtained in different conditions which do not match to the conditions where it is used, is very important since such situation can easily arise during the real operation. It should be noticed that this is possible thanks to the abstract classification features.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

The research conducted in the present work confirmed once again that automated landmine detection is a very challenging task. All stages of the development of such a system contain important unsolved problems, from low-level positioning of a mobile landmine detection platform in unstructured environment, to the high-level recognition task complicated by nonselectivity of the landmine detection sensors. Several of the most critical problems were considered in this work and possible solutions were developed.

The central problem addressed in this work is the **automatic landmine detection** using the sensor data obtained by a mobile scanning platform. Therefore, the available scanning platform LADERO was sufficiently improved in terms of installing necessary control hardware and sensors and implementing the newly developed control software. The landmine detection task was analyzed in detail, first, using the experimental data available from public databases of landmine signatures. This analysis revealed several specific properties of this task and allowed to develop an optimal strategy consisting in a multi-stage approach. Such division of the detection process accounts for the fact that the considered landmine detection sensors are not selective for the landmines. Instead, each sensor distinguishes objects with differing values of some physical property which can be also present in a landmine. This may lead to a situation where the data quality is not enough to perform any recognition. The multi-stage strategy allows to overcome this problem by terminating the detection process after an appropriate stage and safely considering all undistinguishable objects to be landmines. Thus, the detection process can be viewed as a step by step reduction of the false alarm rate where the next improvement is possible if the quality of the sensor data allows it.

The first stage of the landmine detection process consists in the detection of all suspicious objects against the background. A special algorithm was developed for this purpose in order to provide the features essential for the processing of the available

sensor data, characterized as follows:

- The algorithm is operating online which means that a suspicious object is detected right after it is fully scanned by the scanning platform. This property is crucial to enable a fully automated landmine detection on a mobile robot because there is no predefined knowledge about the locations of the suspicious objects. Having this feature, the robot is able to detect the object in advance and plan a safe path to avoid landmines.

- The developed algorithm does not use the shape of the object as a vital characteristic for the detection process. Even if it is possible to specify the shape of the objects located in a specific area, they can be deformed due to positioning failures of the scanning platform making an assumption of a certain shapes to be not reliable.

- Operation of the algorithm can be performed in different modes conferring detection of the objects with different contrast in relation to the background. Thus, the specificity of each sensor data can be taken into account by choosing an appropriate mode of operation, for example, detecting only objects with positive contrast from pulsed metal detector data, and objects with both negative and positive contrast from IR camera data.

- The algorithm is based on the analysis of relative changes of the sensor signal. Thus, it is not sensitive to the changes of environmental conditions which cause the constant changes of the background level.

The performance of the suspicious object detection algorithm was tested against simulated and experimental data obtained from the public database and from the experiments carried out with the mobile scanning platform LADERO. The obtained results clearly demonstrate that the presented algorithm provides a reliable basis for the subsequent stages of the landmine detection process.

The **next stages** of the landmine detection provide the recognition of landmines among the detected suspicious objects. Analysis of large amounts of experimental data showed that this task should be divided in two stages in order to provide better separability of the classes on each stage: first, the suspicious objects are classified to be man-made or natural, and then the landmines are recognized among the detected man-made objects. A feature-based sensor fusion approach is used for the classification considering the Bayesian classifier and multivariate Gaussian model for representation of the feature parameters. A number of new classification features, which reveal the nature of the object, are proposed, and several other features are adapted from other fields of pattern recognition. The main goal for the development of new classification

features is the improvement of their independence from the nature of a particular sensor. Several of the proposed features meet this requirement and, thus, their values can be fused for the signatures of the same object obtained from different sensors by averaging, conferring a *combined feature* with better characteristics. The considered classification features were analyzed in detail with the help of mutual information and Hausdorff distance measures in order to research the possibilities to improve the classification results. The main obstacle in this direction is a very high overlapping of the classes in the feature space. This problem naturally follows from the fact that the utilized sensors are not specific for the landmine detection. The separation of the classification in two stages is one step toward this goal. However, the classification process was improved further as summarized below.

The modification of the classifier training process is proposed in this work aiming in improvement of the separability of the analyzed classes in case of landmine detection. Therefor, the concept of ***selective training*** was developed here. This approach benefits from the fact that some of the classification features have bimodal distributions for one of the classes where one of the maxima is better separated from the distribution of the other class. In this case the ignorance of the highly overlapped maximum is acceptable, since the second maximum confers better separability of classes. The model of the classifier is then changed to an additive Gaussian model accounting for the new relation between the features when some parts of the distributions are ignored. However, only the presence of several classification features with this behavior can assure improvement of the classification results: if the better separated maxima characterize different sets of objects, a combination of several features can allow to classify the whole set of objects even if large parts of the distributions are ignored. This result can be explained by the fact that the highly overlapped areas of the distributions do not provide any useful information for the classifier. Instead, they can even confuse the classification and reduce its performance.

Although, the *selective training* remarkably enhances the classification results in comparison to the naive approach, the concept of ***dominant class*** was additionally introduced here in order to account for one more specifics of the landmine recognition task. The *selective training* reduces the classification error without any specific restrictions on the values of the resulting detection and false alarm rates, similar to the most approaches in pattern recognition. However, the case of landmine detection is specific in this sense because it is essential to keep the highest possible level of the detection rate (ideally 100%). In order to overcome this problem the concept of *selective training* was further modified to provide a possibility to specify one of the classes to be a *dominant class* modifying the distribution of the classification features before the training of the classifier. In this case the distribution of the *dominant class* is unchanged, and the overlapping part of the other class distribution is ignored. The remaining parts of

the nondominant class distribution now represent the whole distribution allowing the *dominant class* to prevail during the classification. This simple concept leads to a different level of performance consisting in higher detection rates and still conferring the reduction of the false alarm rate. The *dominant class* approach is especially useful if the quality of sensor data is low. The fine training of the classifier using the *selective training* only can be complicated in this case due to lack of information. The introduction of a *dominant class* makes the classification possible and in some cases reduces the false alarm rate.

The developed landmine detection algorithms were implemented on the mobile scanning platform LADERO after being tested on a large amount of experimental data from the public database. The specificities of the data acquired by the platform include low spatial resolution and deformations of the object signatures due to the positioning failures of the platform. Having low-quality data, the concept of *dominant class* was used for the classification. The important result is that the data acquisition problems do not affect the performance of the suspicious object detection algorithm. The data set obtained during the experiments was sufficient only for the evaluation of the classification and was not large enough to use a part of it for the classifier training. Thus, the classifier was trained using the data available from the public database., although, the sensors installed on LADERO were not employed in that database. It is evident that such process is not possible in the case of the classification features which are highly dependent on the sensor characteristics. However, there is a possibility to use the features which are relatively independent from the sensor as *combined features*. A special measure for estimating the degree of generality of the feature in respect to the *dominant class* was developed. Based on this measure, only the three best features were chosen to form the classifier. After being trained on the experimental data from the public database obtained from four different sensors, the classifier showed good performance on the experimental data obtained by LADERO reducing the false alarm rate by the factor of 1.9. Such operation is only possible thanks to the special classification features which include a high degree of abstraction from the nature of sensor. This achievement proves the possibility to train a classifier for landmine detection using experimental data obtained in different conditions which do not match the conditions where it will be used.

The landmine detection task raises a number of adjacent problems which have to be solved in order to enable the automated operation of a demining robot. First of all, spatially mapped sensor data are required for the functioning of the suspicious object detection algorithm. According to the idea of the project DEMINE the data are obtained by the mobile scanning platform LADERO which has a simple mechanical design based on pneumatic cylinders arranged to form a cartesian structure. The experiments performed with the platform revealed several positioning problems when operating on a rough terrain, which cause unintentional rotation of the platform and slippage of the

legs. Several **low-level control algorithms** were developed in order to solve some of these problems, namely the the parallel cylinders uneven speed and poor foot ground contact. However, several unsolved problems still remain and probably can be solved only by reconsidering the mechanical design of the platform. An additional **vision system** for the detection of positioning failures was developed and integrated with the odometry system of the robot.

Stepping from the low-level control of the platform to its behavior on the level of a minefield, the problem of **high-level path planning** is next addressed. In general, the area where the robot is operating is completely unknown in advance which means that an obstacle can be situated on the robot path. To perform the scanning of the area in case of obstacles, the problem of **complete area coverage** has to be solved. The algorithm developed here for this purpose utilizes the concept of sensor-based exploration of an unknown area. During the scanning path of the robot the environment is sensed using sonars and decomposed in several cells if an obstacle is encountered. The decomposition is performed in a way that each cell can be completely covered by the simple scanning pattern of the robot. An important feature of the developed algorithm is its weak dependence on the quality of the robot localization system assuring higher safety of the robot in case of failures. Other properties consist in the regular path generated by the algorithm allowing the scanning task to be performed and in the restriction for any path taken by the robot to be situated inside of the already covered area. The developed algorithm implemented on the test mobile robot demonstrated good performance during the experimental tests conferring complete coverage of the given area.

Summarizing the results gained during this research it can be concluded that the application of a mobile scanning platform for automatic landmine detection is promising. The algorithms developed for the platform allow:

- effective landmine detection promising high detection rates;
- improved positioning of the LADERO platform leading to more reliable data acquisition;
- complete coverage of an unknown area assuring high safety of the robot;

The main pitfall seems to be the local positioning of the used experimental platform which has to be improved on the mechanical level or by redesigning the concept of the platform. The promising results of experimental trials on test minefields using the LADERO platform equipped with the developed algorithms confirms their applicability for autonomous landmine detection robots.

It should be also mentioned that the landmine detection task raises many challenging problems which interlap with other areas of robotics and pattern recognition. For

example, the unknown area coverage algorithm developed in this work can also be used for similar tasks, like cleaning or grass cutting.

8.2 Original contributions and key publications

- Combination in one framework of the landmine detection and the path planning algorithms assuming the restrictions of one to another.
- A multi-stage pattern recognition strategy specially dedicated for landmine detection [82].
- A novel online automatic ROI extraction algorithm with low number of parameters and general object model [81].
- Novel classification features for better reflection of shape and nature properties of the objects [83].
- Dedicated concepts for landmine recognition, *selective training* and *dominant class*, which allow to benefit from the bimodal distributions of classification features.
- A concept of *mixed features* which allows to use the idea of signal-level sensor fusion for landmine recognition [82].
- A number of solutions for increasing of the reliability of the cartesian pneumatic platform and a vision system for slippage control based on the detection of natural landmarks on the ground [84].
- An approach for *unknown area coverage* which satisfies the restrictions applied by the demining task: the coverage path must be regular to allow acquiring data from the sensor and the path of the platform should always lie in the already covered area [86, 85].
- Implementation of the developed algorithms for landmine detection and path planning in one software framework which allows a transparent transition between different mechanical platforms.
- A graphical operator interface with the ability of simulation based on the acquired experimental data.

List of key publications:

- [119] S. Larionova, N. Almeida, L. Marques, and A. T. de Almeida. Olfactory coordinated area coverage. Autonomous Robots, 20(3):251-260, 2006.

- [82] S. Larionova, L. Marques, and T. de Almeida. Multi-stage sensor fusion for landmine detection. In Int. Conf. on Intelligent Robots and Systems (IROS), Beijing, China, 9-15 October, pages 2943-2948, 2006.

- [84] S. Larionova, L. Marques, and T. de Almeida. Detection of natural landmarks for mapping by a demining robot. In Int. Conf. on Intelligent Robots and Systems (IROS), Beijing, China, 9-15 October, pages 4959-4964, 2006.

- [85] S. Larionova, L. Marques, and T. de Almeida. Path planning for a demining robot. In Int. IARP workshop RISE, Brussels, Belgium, June, 2006.

- [83] S. Larionova, L. Marques, and T. de Almeida. Features selection for sensor fusion in a demining robot. In Int. Conf. on Robotics and Automation, ICRA, pages 3175-3180, 2005.

- [81] S. Larionova, S. Larionova, L. Marques, and T. de Almeida. Toward practical implementation of sensor fusion for a demining robot. In Int. Conf. on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September-2 October, volume 3, pages 3039-3044, 2004.

- [120] Feature-level sensor fusion for a demining robot. In IARP Int. Workshop Robotics and Mechanical Assistance in Humanitarian Demining and Similar Risky Interventions, Brussels, Belgium, 16-18 June, 2004.

- [86] S. Larionova, N. Almeida, L. Marques, and T. de Almeida. Olfactory coordinated area coverage. In IEEE Int. Conf. on Automation and Robotics (ICAR), Coimbra, Portugal, 30 June-3 July, pages 501-506, 2003.

8.3 Recommendations for Future Work

- Improvement of the mobile scanning platform is a necessary step toward the real implementation of automated demining technology. Based on the developed landmine detection strategy it seems that the most appropriate structure of the robot consists in two layers:

    1. A platform with good adaptability to move in rough terrain which can provide a reliable trajectory and a course scanning during the movement. The sensor data obtained from the platform can be processed by the algorithm for suspicious objects detection. If an object is detected, the platform should switch to the second layer.

    2. A precise scanning device able to scan an area roughly equal to the size of landmines signatures. This device is carried by the main platform and

activated in case an object is detected. These data are then of high quality and can be processed by the landmine detection algorithms.

- The concept of *selective training* can be tested on other classification tasks with a similar problem of highly overlapped classes. There is evidence that the classification features with a bimodal distribution for one of the classes may be present in other pattern recognition problems. For example, considering a task to recognize between pictures of cats and dogs, one of the classification features can be the size of the body. The distribution of the size for the dogs class most probably would be bimodal with one maximum overlapped with the cats class.

- The online coverage algorithm can be improved for other robotic applications, including its usage for mobile robot cleaning in indoor environment. The improvement should be mostly related with localization sensors of the robot, particularly the direction sensor. Some work has been done in the topic of using an electronic compass indoor (which is complicated by the existence of the metallic parts of the buildings). The preliminary results show that this can be made possible by fusing the values of the compass with the odometry direction sensor (for example, angular position of the wheeled robot monitored by its encoders) if the fusion process is controlled according to the current behavior of the robot. For example, if the robot moves forward (which means, its wheels are driven with the same speed) than the odometry sensor is expected to provide reliable value, while the compass can be influenced by the changing conditions along the robot path. When instead the robot rotates without changing the position, the situation is opposite. The fused value can benefit from the best sensor in each moment.

Bibliography

[1] Landmine monitor report 2006 (online). International Campaign to ban Landmines (ICBL), http://www.icbl.org/lm/2006/ (13.02.2007).

[2] J. Ishikawa, M. Kiyota, and K. Furuta. Evaluation of test results of gpr-based anti-personnel landmine detection systems mounted on robotic vehicles. In *IARP Int. Workshop on Robotics and Mechanical Assistance in Humanitarian Demining, Tokyo, Japan, 21-23 June*, 2005.

[3] Joint Research Centre (JRC) Landmines Signature Database (online). http://apl-database.jrc.it/Home/ (13.02.2007).

[4] M. A. Neves, R. R. Gomes, and R. M. Costa. Robô com pernas para desminagem humanitária. Diploma thesis. University of Coimbra, 2003.

[5] Joint Multi-Sensor Mine Signature Database MsMs (online). http://demining.jrc.it/msms/ (13.02.2007).

[6] The Convention on the Prohibition of the Use, Stockpiling, Production and Transfer of Anti-Personnel Mines and on Their Destruction (online). International Campaign to ban Landmines (ICBL), http://dev.icbl.org/treaty/ (13.02.2007).

[7] Geneva International Center for Humanitarian Demining (online). http://www.gichd.ch (17.11.2006).

[8] International Mine Action Standard (IMAS) 09.10. Clearance requirements (online). United Nations Mine Action Service (UNMAS), http://www.mineactionstandards.org/IMAS_archive/Amended/Amended1/IMAS_0910_1.pdf (01.01.2003).

[9] ORDATA database (online). http://www.maic.jmu.edu/ordata/ (13.02.2007).

[10] *Landmines must be stopped.* International Committe of the Red Cross, Geneva, 1997.

[11] ELADIN, High Pressure Waterjet Laboratory, Rock Mechanics and Explosive Research Center, University of Missouri - Rolla (online). http://eladin.umr.edu/ (13.02.2007).

[12] B. Haight. Cleaning up an explosive problem: project Eladin uses water to detect, expose and neutralize abandoned land mines. *Diesel Progress North American Edition*, (12), 2002.

[13] S. Hirose, S. Yokota, A. Torii, M. Ogata, S. Suganuma, K. Takita, and K. Kato. Quadruped walking robot centered demining system - development of TITAN-IX and its operation-. In *IEEE Int. Conf. on Robotics and Automation(ICRA), Barcelona, Spain, 18-22 April*, pages 1296–1302, 2005.

[14] E. U. Acar, Y. P. Zhang, H. Choset, M. Schervish, A. G. Costa, R. Melamud, D. C. Lean, and A. Graveline. Path planning for robotic demining and development of a test platform. In *Int. Conf. on Field and Service Robotics, Helsinki, Finland, 11-13 June*, pages 161–168, 2001.

[15] E. Gonzalez, M. Alarcon, C. Parra, and Y. F. Zheng. BSA: A coverage algorithm. In *IEEE Int. Conf on Intelligent Robots and Systems (IROS), Las Vegas, USA, 27-31 October*, volume 2, pages 1679 – 1684, 2003.

[16] K. Kato, S. Hirose, J. Estremera, E. Garcia, S. Suganuma, K. Takita, and K. Kato. Proposition of the humanitarian demining system by the quadruped walking robot. In *IEEE Int. Conf on Intelligent Robots and Systems (IROS), Takamatsu, Japan, 30 October-5 November*, pages 769–774, 2000.

[17] S. Hirose, K. Yoneda, H. Tsukagoshi, J. Guilberto, S. Suganuma, K. Takita, and K. Kato. TITAN VII: Quadruped walking and manipulating robot on a steep slope. In *IEEE Int. Conf. on Robotics and Automation (ICRA), Albuquerque, USA, April*, pages 494–500, 1997.

[18] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull. Morse decompositions for coverage tasks. *The International Jounal of Robotics Research*, 21(4):331–344, 2002.

[19] E. U. Acar, H. Choset, Y. P. Zhang, and M. Schervish. Sensor-based coverage of unknown environments: Incremental construction of morse decompositions. *The International Jounal of Robotics Research*, 21(4):345–366, 2002.

[20] E. U. Acar, H. Choset, Y. P. Zhang, and M. Schervish. Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *The International Jounal of Robotics Research*, 22:441–466, 2003.

[21] Y. P. Zhang, M. Schervish, E. U. Acar, and H. Choset. Probabilistic methods for robotic landmine search. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS), Maui, Hawaii, Nov.*, pages 1525–1532, 2001.

[22] M. YU. Rachkov, L. Marques, and A. T. de Almeida. Multisensor demining robot. *Autonomous Robots*, 18(3):275–291, 2005.

[23] G. A. Clark, S. K. Sengupta, D. Aimonetti, F. Roeske, and J. G. Donetti. Multispectral image feature selection for land mine detection. *IEEE Trans. on Geoscience and Remote Sensing*, 38(1):304–311, 2000.

[24] L. Kempen, A. Katartzis, V. Pizurica, J. Cornelis, and H. Sahli. Digital signal/image processing for mine detection. part1: Airborne approach. In *Euroconference on Sensor Systems and Signal Processing Techniques Applied to the Detection of Mines and Unexploded Ordance, Firenze, Italy, 1-3 October*, pages 48–53, 1999.

[25] J. Coronado-Vergara, G. Avina-Cervantes, M. Devy, and C. Parra. Towards landmine detection using artificial vision. In *IEEE Int. Conf. on Intelligent Robots and Systems, Alberta, Canada, 2-6 August*, pages 659–664, 2005.

[26] K. M. Dawson-Howe and T. G. Williams. The detection of buried landmines using probing robots. *Robotics and Autonomous Systems*, (4):235–243, 1998.

[27] D. Guelle, A. Smith, A. Lewis, and T. Bloodworth. *Metal Detector Handbook for Humanitarian Demining, http://www.itep.ws/pdf/metal_detector_handbook.pdf*. Office for Official Publications of the European Communities, Luxembourg, ISBN 92-894-6263-1.

[28] C. Bruschini and H. Sahli. Phase angle based emi object discrimination and analysis of data from a commercial differential two frequency system. In *Proc. of SPIE, Detection and Remediation Technologies for Mines and Minelike Targets V, Orlando, USA, 24-28 April*, volume 4038, 2000.

[29] L. S. Riggs, L. T. Lowe, and D. Weaver. Discrimination between buried metallic mines and metallic clutter using spatial symmetry, signal energy, and exponential decay rates. In *Research on Demining Technologies Joint Workshop, Ispra, Italy, 12-14 July*, 2000.

[30] K. C. Ho, L. M. Collins, L. G. Huettel, and P. D. Gader. Discrimination mode processing for emi and gpr sensors for hand-held land mine detection. *IEEE Trans. on Geoscience and Remote Sensing*, 42(1):249–263, 2004.

[31] M. Peichl, S. Dill, H. Suess, and P. D. Gader. Application of microwave radiometry for buried landmine detection. In *Int. Workshop on Advanced Ground Penetrating Radar, TU Delft, Delft, 14-16 May*, pages 172–176, 2003.

[32] W. Messelink, K. Schutte, A. Vossepoel, F. Cremer, J. Schavemaker, and E. Breejen. Feature-based detection of landmines in infrared images. In *Proc. of SPIE, Detection and Remediation Technologies for Mines and Minelike Targets VII, Orlando, USA, 1-5 April*, volume 4742, pages 108–119, 2002.

[33] F. Cremer, W. Jong, K. Schutte, A. G. Yarovoy, and V. Kovalenko. Feature level fusion of polarimetric infrared and gpr data for landmine detection. In *Int. Conf. on Requirements and Technologies for the Detection, Removal and Neutralization of Landmines and UXO, Brussels, Belgium, Sep.*, pages 638–642, 2003.

[34] H. Frigui, P. D. Gader, J. M. Keller, and K. Schutte. Fuzzy clustering for land mine detection. In *Conf. of the North American Fuzzy Information Processing Society - NAFIPS, USA, 20-21 August*, pages 261–265, 1998.

[35] M. Roughan, D. W. McMichael, K. C. Ho, A. G. Yarovoy, and V. Kovalenko. A comparison of methods of data fusion for land-mine detection. In *Int. Workshop on Image Analysis and Information Fusion, Adelaide, Australia, 6-8 November*, 1997.

[36] G. A. Jarrad and D. W. McMichael. Improving multispectral mine detection methods by compensating for clutter. In *Australian-American Joint Mine Warfare Conference, Sydney, Australia, July*, 1999.

[37] O. G. Merino. Image analysis of Infrared Polarization measurements of landmines. Master thesis. Vrije Universiteit Brussel (VUB)-ETRO department, 2001.

[38] F. Cremer, W. Jong, K. Schutte, and A. Bibaut. Fusion of polarimetric infrared features and gpr features for landmine detection. In *Int. Workshop on Advanced Ground Penetrating Radar, TU Delft, Delft, 14-16 May*, pages 222–227, 2003.

[39] F. Cremer, W. Jong, K. Schutte, W.-J. Liao, and B. Baertlein. Detectability of surface-laid landmines with a polarimetric ir sensor. In *Proc. of SPIE, Detection and Remediation Technologies for Mines and Minelike Targets VIII, Orlando, USA, 21-25 April*, volume 5089, pages 505–516, 2003.

[40] F. Cremer, W. Jong, K. Schutte, W.-J. Liao, and B. Baertlein. Processing of polarimetric infrared images for landmine detection. In *Int. Workshop on Advanced Ground Penetrating Radar, Delft, The Netherlands, 14-16 May*, pages 216–221, 2003.

[41] B. Hosgood and G. Andreoli. Experiments with microwave heat sources for thermal stimulation of anti-personnel landmines. Technical Report SPL0251, HS Unit/IPSC/JRC Ispra, 2001.

[42] J. J. Bromenshenk, C. B. Henderson, R. A. Seccomb, S. D. Rice, R. T. Etter, S. F. A. Bender, P. J. Rodacy, J. A. Shaw, N. L. Seldomridge, L. H. Spangler, and J. J. Wilson. Can honey bees assist in area reduction and landmine detection? *MAIC Journal of Mine Action*, (7), 2003.

[43] A. D. Hibbs. *Alternatives for Landmine Detection*, chapter Nuclear Quadrupole Resonance, pages 169–189. Rand, Skokie, 1999.

[44] C. Sheehy. Fast neutron technology used for explosive detection. *National Defense*, (6), 2003.

[45] C. Bruschini. Commercial systems for the direct detection of explosive ordnance disposal tasks. *Subsurface Sensing Technologies and Applications*, 2(3):299–336, 2001.

[46] C. Wu. Digging in the dirt: chemical and biological sensors could aid the search for hidden land mines. *Science News*, (3), 1998.

[47] M. A. Gibs. Will every rat have its day? can wasps, moths, pigs, and rats be as useful as dogs in detecting dangerous chemicals? researchers say yes. *Security Management Magazine*, (1), 2006.

[48] S. Milius. Rescue rat: could wired rodents save the day? *Science News*, (5), 2002.

[49] P. Gwin. Weeding out land mines: gene-altered plant blushes when roots sense explosives. *National Geographic Magazine*, (3), 2005.

[50] S. W. Kercel, R. S. Burlage, D. R. Patek, C. M. Smith, A. D. Hibbs, and T. J. Rayner. Novel methods for detecting buried explosive devices. In *Proc. of SPIE, Detection and Remediation Technologies for Mines and Minelike Targets II, Orlando, USA, 21-24 April*, volume 3079, pages 467–477, 1997.

[51] R. S. Burlage, M. Hunt, J. DiBenedetto, M. Maston, R. T. Etter, S. F. A. Bender, P. J. Rodacy, J. A. Shaw, N. L. Seldomridge, L. H. Spangler, and J. J. Wilson. Bioreporter bacteria for the detection of unexploded ordnance. *Humanitarian Demining Web Site at the University of Western Australia* `http://www.mech.uwa.edu.au/jpt/demining`, 2002.

[52] Y. P. Zhang, L. M. Collins, H. Yu, C. E. Baum, and L. Carin. Sensing of unexploded ordance with magnetometer and induction data: Theory and signal processing. *IEEE Trans. on Geoscience and Remote Sensing*, 41(5):1005–1015, 2003.

[53] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Interscience Publication, New York, 2000.

[54] L. M. Collins, P. Gao, D. Schofield, J. P. Moulton, L. C. Makowsky, D. M. Reidy, and R. C. Weaver. A statistical approach to landmine detection using broadband electromagnetic induction data. *IEEE Trans. on Geoscience and Remote Sensing*, 40(4):950–962, 2002.

[55] L. M. Collins, P. Gao, L. Carin, J. P. Moulton, L. C. Makowsky, D. M. Reidy, and R. C. Weaver. An improved bayesian decision theoretic approach for land mine detection. *IEEE Trans. on Geoscience and Remote Sensing*, 37(2):811–819, 1999.

[56] P. Gao, S. L. Tantum, L. M. Collins, D. Weaver, J. P. Moulton, and L. C. Makowsky. Statistical signal processing techniques for the detection of low-metal landmines using emi and gpr sensors. *Geoscience and Remote Sensing Symposium, Hamburg, 28 June-02 July*, 5:2465–2467, 1999.

[57] P. Torrione and L. M. Collins. Performance comparison of automated induction-based algorithms for lanmine detection in a blind field test. *Subsurface Sensing Technologies and Applications*, 5(3):121–150, 2004.

[58] S. Perrin, E. Duflos, P. Vanheeghe, and A. Bibaut. Multisensor fusion in the frame of evidence theory for landmines detection. *IEEE Trans. on Systems, Man and Cybernetics, Part C*, 34(4):485–498, 2004.

[59] N. Milisavljevi, I. Bloch, L. G. Huettel, and P. D. Gader. Sensor fusion in anti-personnel mine detection using a two-level belief function model. *IEEE Trans. on Systems, Man and Cybernetics, Part C*, 33(2):269–283, 2003.

[60] N. Milisavljevi, I. Bloch, M. Acheroy, E. Breejen, and K. Benoist. Characterization of mine detection sensors in terms of belief functions and their fusion, first results. volume 2, pages ThC3 – 15–22, 2000.

[61] R. J. Stanley, P. D. Gader, K. C. Ho, A. G. Yarovoy, and V. Kovalenko. Feature and decision level sensor fusion of electromagnetic induction and ground penetrating radar sensors for landmine detection with hand-held units. *Information Fusion Jornal*, 3(3):215–223, 2002.

[62] A. Filippidis, L. C. Jain, and P. Lozo. Degree of familiarity art2 in knowledge-based landmine detection. *IEEE Trans. on Neural Networks*, 10(1):186–193, 1999.

[63] S. Sheedvash and M. R. Azimi-Sadjadi. Structural adaptation in neural networks with application to landmine detection. In *Int. Conf. on Neural Networks, 9-12 June*, volume 3, pages 1443–1447, 1997.

[64] F. Cremer, J. Schavemaker, W. Jong, and K. Schutte. Comparison of vehicle-mounted forward-looking polarimetric infrared and downward-looking infrared sensors for landmine detection. In *Proc. of SPIE, Detection and Remediation Technologies for Mines and Minelike Targets VIII, Orlando, USA, 21-25 April*, volume 5089, pages 517–526, 2003.

[65] A. Filippidis, L. C. Jain, N. Martin, E. Breejen, and K. Benoist. Multisensor data fusion for surface land-mine detection. *IEEE Trans. on Systems, Man and Cybernetics, Part C*, 30(1):145–150, 2000.

[66] E. Breejen, K. Schutte, F. Cremer, K. Schutte, and K. Benoist. Sensor fusion for anti personnel landmine detection: a case study. In *Proc. of SPIE, Detection and Remediation Technologies for Mines and Minelike Targets IV, Orlando, USA, Apr.*, volume 3710, pages 1235–1245, 1999.

[67] R.C. Luo and M.G. Kay. *Multisensor Integration and Fusion for Intelligent Machines and Systems.* Ablex Publishing Corporation, Norwood, ISBN:0-89391-863-6, 1995.

[68] E. Waltz and J. Llinas. *Multisensor Data Fusion.* Artech House, Norwood, 1990.

[69] A. Gunatilaka, B. Baertlein, K. C. Ho, A. G. Yarovoy, and V. Kovalenko. Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 23, pages 577–589. 2001.

[70] A. Gunatilaka, B. Baertlein, K. C. Ho, A. G. Yarovoy, and V. Kovalenko. Comparison of pre-detection and post-detection fusion for mine detection. In *Proc. of SPIE, Detection and Remediation Technologies for Mines and Minelike Targets III, Orlando, Florida, 5-9 April*, volume 3710, pages 1212–1223, 1999.

[71] B. Baertlein, A. Gunatilaka, K. C. Ho, A. G. Yarovoy, and V. Kovalenko. Improving detection of buried land mines through sensor fusion. In *Proc. of SPIE, Detection and Remediation Technologies for Mines and Minelike Targets, Orlando, Florida, April*, volume 3392, pages 1122–1133, 1998.

[72] G. A. Clark, S. K. Sengupta, M.R. Buhl, R. J. Sherwood, P. C. Schaich, N. Bull, R. J. Kane, M. J. Barth, D. J. Fields, and M. R. Carter. Detecting buried objects by fusing dual-band infrared images. In *27th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, USA, 1-3 November*, volume 1, pages 135–143, 1993.

[73] G. A. Clark, J. E. Hernandez, S. K. Sengupta, R. J. Sherwood, P. C. Schaich, M.R. Buhl, R. J. Kane, M. J. Barth, N. K. DelGrande, and M. R. Carter. Computer vision and sensor fusion for detecting buried objects. In *Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, USA, 1-3 October*, pages 466–471, 1992.

[74] C. Bruschini, B. Gros, F. Guerne, P.-Y. Piece, and O. Carmona. Ground penetrating radar and induction coil sensor imaging for antipersonnel mines detection. In *Int. Conf. on Ground Penetrating Radar, Sendai, Japan, 30 September-3 October*, pages 211–216, 1996.

[75] F. Cremer, K. Schutte, J. Schavemaker, E. Breejen, and K. Benoist. Towards an operational sensor-fusion system for anti-personnel landmine detection. In *Detection and Remediation Technologies for Mines and Minelike Targets V, Orlando, USA, 24-28 April*, volume 4038, pages 792–803, 2000.

[76] F. Cremer, K. Schutte, J. Schavemaker, and E. Breejen. A comparison of decision-level sensor-fusion methods for anti-personnel landmine detection. *Information Fusion Jornal*, (2):187, 2001.

[77] J. Schavemaker, E. Breejen, F. Cremer, K. Schutte, and K. Benoist. Depth fusion for anti-personnel landmine detection. In *Detection and Remediation Technologies for Mines and Minelike Targets VI, Orlando, USA, Apr.*, volume 4394, pages 1071–1081, 2001.

[78] M. Fritzsche, O. Lohlein, M. Acheroy, E. Breejen, and K. Benoist. Sensor fusion for the detection of landmines. *Subsurface Sensing Technologies and Applications*, 1(2):247–267, 2000.

[79] R. Kacelenga, D. Erickson, D. Palmer, and P. D. Gader. Voting fusion adaptation for landmine detection. *AES Magazine*, 18(8):13–19, 2003.

[80] R. Kacelenga, D. Erickson, D. Palmer, and P. D. Gader. Voting fusion adaptation for landmine detection. In *IEEE Int. Conf. on Information Fusion, Annapolis, USA, 811 July*, volume 1, pages 333–340, 2002.

[81] S. Larionova, L. Marques, and T. de Almeida. Toward practical implementation of sensor fusion for a demining robot. In *Int. Conf. on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September-2 October*, volume 3, pages 3039–3044, 2004.

[82] S. Larionova, L. Marques, and T. de Almeida. Multi-stage sensor fusion for landmine detection. In *Int. Conf. on Intelligent Robots and Systems (IROS), Beijing, China, 9-15 October*, pages 2943–2948, 2006.

[83] S. Larionova, L. Marques, and T. de Almeida. Features selection for sensor fusion in a demining robot. In *Int. Conf. on Robotics and Automation, ICRA*, pages 3175–3180, 2005.

[84] S. Larionova, L. Marques, and T. de Almeida. Detection of natural landmarks for mapping by a demining robot. In *Int. Conf. on Intelligent Robots and Systems (IROS), Beijing, China, 9-15 October*, pages 4959–4964, 2006.

[85] S. Larionova, L. Marques, and T. de Almeida. Path planning for a demining robot. In *Int. IARP workshop RISE, Brussels, Belgium, June*, 2006.

[86] S. Larionova, N. Almeida, L. Marques, and T. de Almeida. Olfactory coordinated area coverage. In *IEEE Int. Conf. on Automation and Robotics (ICAR), Coimbra, Portugal, 30 June-3 July*, pages 501–506, 2003.

[87] Test minefield at meerdaal bomb disposal unit (online). `http://www.itep.ws/facilities/Belgium/belgium.htm` (13.02.2007).

[88] G. Smith, R. Smith, and A. Wardhani. Software reuse across robotic platforms: Limiting the effects of diversity. In *Australian Software Engineering Conference, Brisbane, Australia, 29 March-1 April*, pages 252–261, 2005.

[89] F. Pont, R. Siegwart, and A. Wardhani. A real-time siftware framework for indoor navigation. In *IEEE Int. Conf on Intelligent Robots and Systems (IROS), Alberta, Canada, 2-6 August*, pages 2085 – 2090, 2005.

[90] H. Utz, S. Sablatnog, S. Enderle, and G. Kraetzschmar. Miro - middleware for mobile robot applications. *IEEE Trans. on Robotics and Automation*, 18(4):493–497, 2002.

[91] R. Volpe, I. A. D. Nesnas, T. Estlin, D. Mutz, R. Petras, and H. Das. CLARAty: Coupled Layer Architecture for Robotic Autonomy. JPL Technical Report D-19975, 2000.

[92] D. J. Miller, R. C. Lennox, T. Estlin, D. Mutz, R. Petras, and H. Das. An object-oriented environment for robot system architectures. *IEEE Control Systems Magazine*, 11:14–23, 1991.

[93] G. A. Clark, J. E. Hernandez, N. K. DelGrande, R. J. Sherwood, S. Lu, P. C. Schaich, and P. F. Durbin. Computer vision for locating buried objects. In *25th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, USA, 4-6 November*, volume 2, pages 1235–1239, 1991.

[94] J. M. Stiles, A. V. Apte, and B. Beh. A group-theoretic analysis of symmetric target scattering with application to landmine detection. *IEEE Trans. on Geoscience and Remote Sensing*, 40(8):1802–1814, 2002.

[95] N. Sarkar and B. B. Chaudhuri. An efficient differential box-counting approach to compute fractal dimension of image. *IEEE Trans. on Systems, Man, and Cybernetics*, 24(1):115–120, 1994.

[96] N. Kwak and C.-H. Choi. Input feature selection for classification problems. *IEEE Trans. on Neural Networks*, 13(1):143–159, 2002.

[97] S. Piramuthu. The hausdorff distance measure for feature selection in learning applications. *Int. Conf. on System Sciences, Maui, Hawaii, USA, 5-8 January*, 1999.

[98] B. Gassmann, F. Zacharias, J. M. Zollner, and R. Dillmann. Localization of walking robots. In *IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain, 18-22 April*, pages 1483–1488, 2005.

[99] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe. Gps/imu data fusion using multisensor kalman filtering: Introduction of contextual aspects. *Information Fusion Journal*, (7):221–230, 2006.

[100] G. N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002.

[101] E. Todt, C. Torras, and S. M. Abdallah. Detection of natural landmarks through multiscale opponent features. In *15th Int. Conf. on Pattern Recognition, Barcelona, Spain, 3-8 September*, volume 3, pages 976–979, 2000.

[102] D. C. Asmar, J. S. Zelek, and S. M. Abdallah. Seeing the trees before the forest. In *2nd Canadian Conf. on Computer and Robot vision, Victoria, Canada, 9-11 May*, pages 587–593, 2005.

[103] R. C. Luo, H. Potlapalli, and S. M. Abdallah. Fractal based outdoor landmark recognition system for the navigation of a mobile robot. In *IEEE Int. Conf. on Robotics and Automation, San Diego, USA, May*, volume 3, pages 1973 – 1978, 1994.

[104] J. B. Gao, C. J. Harris, J. M. Zollner, and R. Dillmann. Some remarks on kalman filters for the multisensor fusion. *Information Fusion Jornal*, (3):191–201, 2002.

[105] Q. Gan, C. J. Harris, J. M. Zollner, and R. Dillmann. Comparison of two measurement fusion methods for kalman-filter-based multisensor data fusion. *EEE Trans. on Aerospace and Electronics Systems*, 37(1):273–279, 2001.

[106] X. Qiu, S. Liu, and S. X. Yang. A rolling method for complete coverage path planning in uncertain environments. In *IEEE Int. Conf. on Robotics and Biomimetics (RoBio04), Shengyang, China, 22-26 August*, pages 146–151, 2004.

[107] C. Luo, S. X. Yang, and X. Yuan. Real-time area-covering operations with obstacle avoidance for cleaning robots. In *IEEE Int. Conf on Intelligent Robots and Systems (IROS), Lausanne, Switzerland, 30 September-4 October*, volume 3, pages 2359–2364, 2002.

[108] C. Luo, S. X. Yang, and M. Q.-H. Meng. Real-time map building and area coverage in unknown environments. In *IEEE Int. Conf. on Robotics and Automation (ICRA), Barcelona, Spain, 18-22 April*, pages 1748–1753, 2005.

[109] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra, and C. Bustacara. Bsa: A complete coverage algorithm. In *IEEE Int. Conf. on Robotics and Automation(ICRA), Barcelona, Spain, 18-22 April*, pages 2040–2044, 2005.

[110] A. Pirzadeh, W. Snyder, C. Parra, and Y. F. Zheng. A unified solution to coverage and search in explored and unexplored terrains using indirect control. In *IEEE Int. Conf. on Robotics and Automation (ICRA), Cincinnati, USA, May*, volume 3, pages 2113–2129, 1990.

[111] E. U. Acar, H. Choset, and J. Y. Lee. Sensor-based coverage with extended range detectors. *IEEE Trans. on Robotics*, 22(1):189–198, 2006.

[112] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset. Limited communication, multi-robot team based coverage. In *IEEE Int. Conf. on Robotics and Automation, New Orleans, USA, 26 April-1 May*, volume 2, pages 3462–3468, 2004.

[113] H. Choset, P. Pingnon, A. A. Rizzi, P. N. Atkar, and D. Hull. Coverage path planning: The boustrophedon cellular decomposition. In *Int. Conf. on Field and Service Robotics, Canberra, Australia, Dec.*, 1997.

[114] S. Hert, S. Tiwari, and V. Lumelsky. A terrain-covering algorithm for an auv. *Autonomous Robots*, (3):91–119, 1996.

[115] S. C. Wong and B. A. MacDonald. A topological coverage algorithm for mobile robots. In *IEEE Int. Conf on Intelligent Robots and Systems (IROS), Las Vegas, USA, 27-31 October*, volume 2, pages 1685 – 1690, 2003.

[116] S. Thrun. *Exploring Artificial Intelligence in the New Millenium*, chapter Robotic Mapping: A Survey, pages 1–35. Morgan Kaufmann, San Fancisco, ISBN:1-55860-811-7, 2002.

[117] R.C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, 1998.

[118] S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2):111–127, 2003.

[119] S. Larionova, N. Almeida, L. Marques, and A. T. de Almeida. Olfactory coordinated area coverage. *Autonomous Robots*, 20(3):251–260, 2006.

[120] Feature-level sensor fusion for a demining robot. In *IARP Int. Workshop Robotics and Mechanical Assistance in Humanitarian Demining and Similar Risky Interventions, Brussels, Belgium, 16-18 June*, 2004.

Appendices

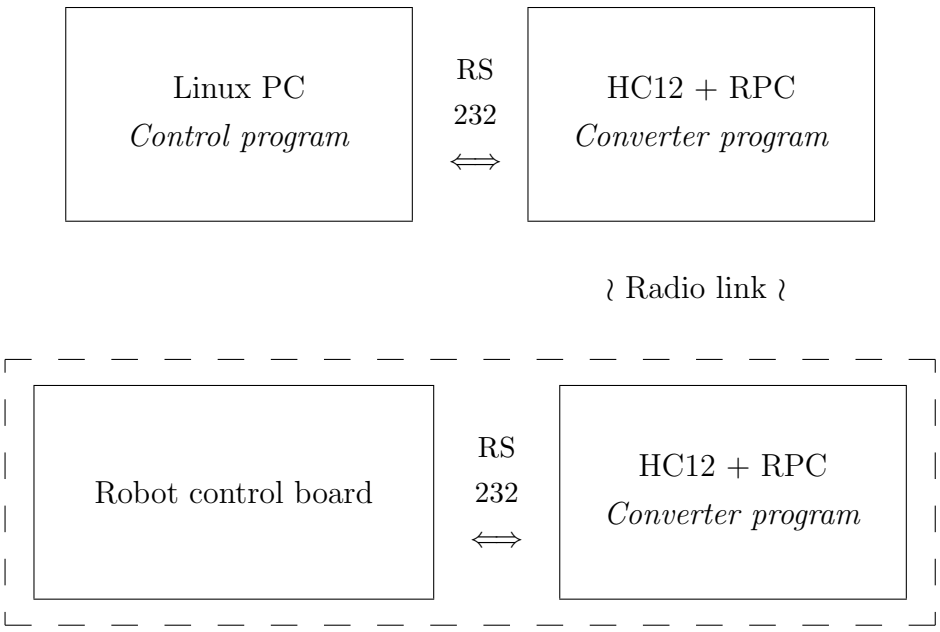


Figure A-1: Structure of control software/hardware

A Nomad Super Scout control software and hardware

A Nomad Super Scout robot is used in the experiments related to the path planning algorithms developed in Chapter 6. In order to provide a convenient development environment the robot is controlled from a remote PC via wireless link. Even so there is a possibility to install an onboard PC on the robot it was not used because the setup described here showed to be more appropriate.

The control board of the Nomad Super Scout provides low level control of the robot by means of commands transferred via a serial link. An onboard computer would be connected to the serial port of the control board. Instead, in the current setup a wireless transmitter is connected. It consists of a Card12 board with MC912D60A microcontroller and a Radiometrix RPC radio transmitter. The serial port of the microcontroller is then connected to the robot control board or to the remote PC serial port. Having two pairs of these devices (one for the robot and one for the remote PC) provides a wireless connection which is transparent for the control programm running on the PC: the programm is not aware that it is running on the remote PC and not on the robot. Figure A-1 shows an overall structure of software/hardware of this setup.

The Radiometrix RPC radio is controlled by the HC12 using an 8-pin interface as shown in Figure A-2 and the *converter program* which is listed below. The main purpose of the *converter program* is to transfer all the data between the serial link and the radio connection without any changes. Besides this function the program includes a feature of parsing the Nomad Super Scout commands in order to transfer additional data from the robot, for example, information from additional sensors. The latest possibility was used in the experiments to acquire data

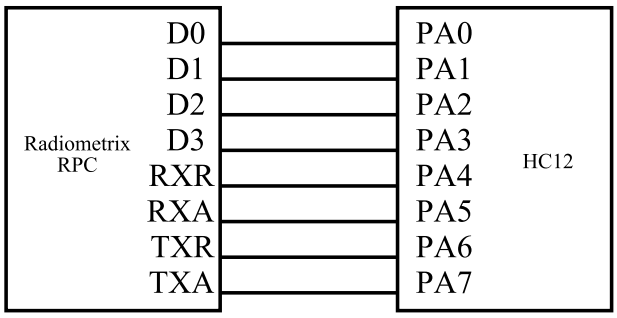


Figure A-2: Interface between the Radiometrix RPC transmitter and Motorola MC912D60A microcontroller

from an electronic compass connected to HC12 via SPI [4].

The format of the Nomad Super Scout commands was found to be the following:

,

where CC - 1-byte command code (e.g. *gs* - 0x01)
N - 2-byte length of data in bytes
CS - 1-byte check sum (sum of all bytes: command code, data length and data, limited to one byte)

In the current implementation the command $0x01$ (*gs* command which returns all data of robot) is parsed in order to include the data from the electronic compass. Two unused values of the state vector *State[]* are used to store the compass data: $STATE_VEL_TURRET$ for X component, $STATE_COMPASS$ for Y component. For more details about the robot interface refer to Nomad Super Scout manual.

A.1 Converter program

The converter program is run on the Motorola HC12 microcontroller and its primer goal is to transfer data between radio and serial links. Moreover, it performs parsing of the Nomad Super Scout commands in order to include additional sensor data. In the setup presented here the data from electronic compass developed in [4] are obtained from a SPI link and included in the command $0x01$ as described above. The full listing of the program, named *rpcprog* can be found on the attached CD.

Main control and command parsing

```
#define FilaSize 200
char fila[FilaSize];
#define SCOUT 1
```

```
int main(void)
{
 static INT8U valor1;
 static int ptw=0;
 static int pstart=0;
 static int flag = 0;
 static int flag_count = 0;
 int flag_len = 0;
 INT16U chk_sum = 0;
 int cc = 0;
 int diff=0;

 asm("sei");
  _PseudoVectorTbl();// Create pseudo vector table
 DDRP = 0;
 PWCTL = 0;
 setSerial(BAUD38K); // Set speed of serial connection to 3800
 InicializaSPI();
 PORTP &= 0x7f; // select board
 asm("cli");

 DDRA=0x60;
 PORTA|=0x60;

 // initialization of RPC
 send_byte(0xC0);
 send_byte(0x33);

 if (!RXR) IRQ_Handler();

 SendThroughSPI(0x55,0);

 while(1) {
  if(getcharSerial_t(&valor1)) {
   fila[ptw] = valor1;

   if (SCOUT) {
    // parse Scout command
    if ((valor1 == 0x5c || flag_count > 100) && flag == 6) {
     flag = 0;
```

```
    flag_count = 0;
    flag_len = 0;
    chk_sum = 0;
   }
   else if (flag == 5) {
    if (flag_count == (flag_len-1)) {
     fila[ptw] = (chk_sum&0xff);
     flag = 6;
    }
    flag_count ++;
   }
   else if (flag == 4 && flag_count >= 32) {
    if (new_vals == 1)
     fila[ptw] = cur_vals[flag_count-32+1];
    else
     fila[ptw] = 0xff;
    flag_count ++;
    if (flag_count > 35) {
     flag = 5;
     new_vals = 0;
    }
   }
   else if (flag == 4)
    flag_count ++; // counting the data bytes
   else if (flag == 3) { // passing the high byte of length (
      ...because the low byte >= 0x26)
    flag = 4; // the length is large enough
    flag_len |= ((valor1<<8)&0xff00);
   }
   else if (flag == 2) { // high byte of length
    flag_len |= ((valor1<<8)&0xff00);
    if (valor1 > 0)
     flag = 4; // the length is large enough
    else
     flag = 0;
   }
   else if (flag == 1) { // low byte of length
    flag_len = valor1;
    flag = 2;
    if (valor1 >= 0x26)
     flag = 3;
```

```
    }
    else if (flag == 0 && valor1 == 0x01) { // beginning byte 0x01
     flag = 1;
     flag_count = 0;
     flag_len = 0;
     chk_sum = 0;
    }
    if (flag > 0 && flag < 6)
     chk_sum += fila[ptw];
   }
   if (fila[ptw] == ENDCHAR || (ptw-pstart) >= 23 || (ptw < pstart
      ..&& ptw >= 23)) {
    if (ptw >= pstart) {
     send_packet(fila+pstart,ptw-pstart+1,1);
    }
    else {
     send_packet(fila+pstart,FilaSize-pstart,1);
     send_packet(fila,ptw+1,1);
    }
    if (ptw+1 < FilaSize)
     pstart = ptw+1;
    else
     pstart = 0;
   }
   ptw = (ptw < (FilaSize-1))?(1+ptw):0;
  }
 }
}
```

RPC control

```
#define TXA ((PORTA & 0x80) >> 7 )
#define RXA(x) x==1?(PORTA|=0x20):(PORTA&=0xdf)
#define TXR(x) x==1?(PORTA|=0x40):(PORTA&=0xbf)
#define RXR ((PORTA & 0x10) >> 4 )

#define BROADCAST 0
#define ADDRESS 1
#define BASE 7
#define ENDCHAR 0x5c
```

```
typedef struct {
 int dest;
 int orig;
 int size;
 }comand_struct;

int byte_count = 0;
char last_nibble=0;
char control_expected=1;
char address_expected=0;
char Dest_OK=0;
unsigned char RF_byte;

INT8U vals[5]={0,0,0,0,0};
int vali = 0;
INT8U cur_vals[5]={0,0,0,0,0};
INT8U coms[5]={0x08,0x88,0x10,0x90,0x55};
int time = 60;
int new_vals=0;
comand_struct RF_comm;
int tc = 0;

#define LWAIT 1000000
#define SWAIT 5000

int send_byte(unsigned char byte) {

 TXR(0); // request a data transfer

 if (SCOUT) {
  if (vali == 0 && vals[2] <= 0x03 && vals[4] <= 0x03) {
   cur_vals[0] = vals[0];
   cur_vals[1] = vals[1];
   cur_vals[2] = vals[2];
   cur_vals[3] = vals[3];
   cur_vals[4] = vals[4];
   new_vals = 1;
  }
  if (tc == (time+22))
   PORTP &= 0x7f; // select board
 }
```

```
unsigned long c = 0;
while (TXA && c < LWAIT) c++; // wait for request accept
if (c >= LWAIT) {
 TXR(1);
 return 0;
}
DDRA = 0x6f; // RXA and TXR as outputs
PORTA &= 0xf0; // reset data bits to 0
PORTA |= byte & 0x0f; // LS nibble
TXR(1);
c = 0;
while (!TXA && c < SWAIT) c++; // wait when request accept becomes
    ... 1
if (c >= SWAIT)
 return 0;
TXR(0);
c = 0;
while (TXA && c < LWAIT) c++; // wait for request accept
if (c >= LWAIT) {
 TXR(1);
 return 0;
}
PORTA &= 0xf0; // reset data bits to 0
PORTA |= (byte & 0xf0) >> 4; // MS nibble
TXR(1);

if (SCOUT) {
 // read data from SPI electronic compass
 if (tc == (time+30)) {
  if (vali >= 0 && vali < 5) {
   vals[vali] = SendThroughSPI(coms[vali],0);
   if (vali == 4)
    time = 60;
   else
    time = 20;
  }
  if (vali < 4)
   vali ++;
  else
   vali = 0;
```

```
    tc = 0;
   }
   else
    tc++;
 }

 c = 0;
 while (!TXA && c < SWAIT) c++; // wait when request accept becomes
    ... 1
 if (c >= SWAIT)
  return 0;
 DDRA = 0x60; // RXA and TXR as outputs

 return 1;
}

void send_packet (unsigned char *dados, int tamanho, int destino) {
 int i;
 unsigned char b_control, enderecos;

 b_control = 0;
 b_control |= (tamanho+2) & 0x3f;
 send_byte(b_control);

 enderecos = 0;
 enderecos |= destino;
 enderecos |= ADDRESS << 3;
 send_byte(enderecos);

 for (i=0; i < tamanho; i++) {
  if (!send_byte(dados[i]))
   break;
 }
}

// IRQ on RXR
void IRQ_Handler(void) {
 IRQCR = IRQCR & 0xBF;
 asm("cli");
 // request of data transfer from RPC
```

```
DDRA=0x60; // A5 and A6 - outputs
 if (RXR) {
  IRQCR = IRQCR | 0x40; //reactivate interrupt
  return;
 }
 RXA(0); // acknowledge the data transfer

 unsigned long c = 0;
 while(!RXR && c < SWAIT) c++; // waiting when RXR becomes high
 if (c == SWAIT) {
  RXA(1); // acknowledge that the data was received
  IRQCR = IRQCR | 0x40; //reactivate interrupt
  return;
 }
  RF_byte = PORTA & 0x0F; // LS nibble
  last_nibble=1;
  RXA(1);
  c = 0;
  while(RXR && c < LWAIT) c++;
  if (c == LWAIT) {
   IRQCR = IRQCR | 0x40; //reactivate interrupt
   return;
  }

  RXA(0);
  c = 0;
  while(!RXR && c < SWAIT) c++; // waiting when RXR becomes high
  if (c == SWAIT) {
   RXA(1); // acknowlendge that the data was received
   IRQCR = IRQCR | 0x40; //reactivate interrupt
   return;
  }
  RF_byte = RF_byte | ((PORTA & 0x0f) << 4); // MS nibble
  last_nibble=0;

  if (control_expected) {
   RF_comm.size=RF_byte & 0x1F; // size of packet
   RF_comm.size -= 2; // number of data bytes which will come next
   Dest_OK = 0;
   if (RF_byte & 0xE0)
    control_expected = 1; // no data
```

```
   else {
    control_expected = 0;
    address_expected = 1;
   }
  }
  else if (address_expected) {
   address_expected = 0;
   RF_comm.dest = RF_byte & 0x07;
   RF_comm.orig = (RF_byte >> 3) & 0x07;
   if (RF_comm.dest == ADDRESS || RF_comm.dest==BROADCAST)
    Dest_OK = 1;
   else
    Dest_OK = 0;
   byte_count = 0;
  }
  else {
   if (Dest_OK == 1) {
    if(!qFull(TXbuff)) {
     poeFila(TXbuff,RF_byte);
     SC0CR2 |= 0x80;      // Enable Tx interrupt
    }
   }
   byte_count ++;
   if (byte_count == RF_comm.size) {
    control_expected = 1;
    address_expected = 0;
    byte_count = 0;
    Dest_OK = 0;
   }
  }
  if (RF_byte == ENDCHAR) {
   if (!control_expected) {
    // packet is broken
    if(!qFull(TXbuff)) {
     poeFila(TXbuff,'*');
     SC0CR2 |= 0x80;      // Enable Tx interrupt
    }
   }
   // reset
   control_expected = 1;
   address_expected = 0;
```

```
    byte_count = 0;
    Dest_OK = 0;
   }

 RXA(1); // acknowledge that the data was received
 IRQCR = IRQCR | 0x40; //reactivate interrupt
}
```

Interface with SPI and serial port

```
INT8U SendThroughSPI(INT8U i, // data to send
 INT8U slave) { // slave to select (if 0 no selection is made)
 // send and receive data
 SP0DR = i; //Registo para dados
 while(!(SP0SR & 0x80)); // wait while all data is sent
 i = SP0DR; // data
 PORTP |= 0x80;
 return(SP0DR);//return data from SPI register
}

void InicializaSPI(void) {
 DDRP = 0x80; // PP7 - output
 PWEN = 0;
 PORTP |= 0x80;

 DDRS = 0xFF;
 SP0CR1 = 0x5C;
 SP0CR2 = 0x00;      // Normal (not bi-directional) mode
 SP0BR = 0x02;    // 1MHz
}
```

```
 #define qInit(x)  ((x).in = (x).out = 0)
 #define qEmpty(x) ((x).in==(x).out)
 #define qFull(x)  ((((x).in+1) & QMASK) == (x).out)
 #define poeFila(Q,REG) \
     (Q).q[(Q).in] = REG; \
     (Q).in = (++((Q).in)) & QMASK
 #define tiraFila(Q) \
     (Q).q[(Q).out]; \
     (Q).out = (++(Q).out) & QMASK
 #define initSci(BAUD) \
```

```
    SC0BD = BAUD; \
    SC0CR2 |= 0x2C
typedef struct {
    INT8U in;
    INT8U out;
    INT8U q[QSIZE];
 } Queue;
Queue RXbuff, TXbuff;

void setSerial(BaudRate baud) {
    qInit(RXbuff);
    qInit(TXbuff);
    initSci(baud);
}

void sci_isr(void) {
 INT8U ssr, sdr;

 ssr = SC0SR1;
 sdr = SC0DRL;

    if((ssr & SCI_RDRF) &&   // byte received
        !qFull(RXbuff)){
            poeFila(RXbuff, sdr);}

 if(ssr & SCI_TDRE)
 {
    if(!qEmpty(TXbuff))    /* Existem bytes para enviar */
  {
   SC0DRL = tiraFila(TXbuff);
  }
  else
   SC0CR2 &= 0x7F;    // Turn off TIE interrupt
 }
}

int getcharSerial_t(char *c) {
  INTR_OFF();
  if (qEmpty(RXbuff))
  {
   INTR_ON();
```

```
    return 0;
  }
  *c = tiraFila(RXbuff);
  INTR_ON();
  return(1);
}
```

B Control Software Implementation Details

High-level *control program* of the demining robot is implemented as a set of C++ classes which allow to execute the developed sensor fusion and path planning algorithms providing the necessary object abstraction (the full listing of this program, named *lowpr*,can be found on the attached CD). All the classes are derived from one basic class `CControlledObject` which defines a basic interface allowing to control all the objects in the `working loop`. This interface includes the following abstract functions:

- `void ProcessFast(CControlledObject* p)` is called for each object each cycle of the `working loop`. The objects reimplement this function to include the specific processing (which should be fast not to block the whole control loop).
- `bool Init(CControlledObject* p)` is called before making any processing to allow initialization of the object. After this function returns `true` it is not called any more and the processing is started.
- `void ArchiveFast()` is called every cycle of `working loop` and should implement a code for archiving the object information (or can be empty if the object does not need to archive anything).

Having these two functions implemented the object is controlled automatically by the following code which is called every cycle of the `working loop`:

```
if (!IsInit())
{
  SetInitFlag(Init(p));

  if (IsInit() && m_archive_unit < 0)
    this->InitArchive(p);
  if (IsInit())
      Start();
}
if (IsInit() && m_is_on)
{
  this->ProcessFast(p);
  if (this->GetArchiveUnit() >= 0 &&
      (g_manager.m_archive.CheckTime(this->GetArchiveUnit()) ==
          ...true) ||
      g_manager.m_archive.IsNeedUpdate(this->GetArchiveUnit()))
  {
    if (IsChanged()  ||
        g_manager.m_archive.IsNeedUpdate(this->GetArchiveUnit()))
      this->ArchiveFast();
    else
```

```
        this->ArchiveEmpty();
      this->Archived();
    }
  }
```

For some of the objects the possibility to perform fast processing in `ProcessFast` is not enough as they incorporate some processing which may take significant time. Possibility to perform slower processing is implemented by an additional class `CSlowObject` whose interface consists in abstract function `ProcessLong()`. The slow part of the object is run in a separated thread and is started as follow:

```
void CSlowObject::Start()
{
  if (Started())
    return;
  m_must_stop = false;
  int res = pthread_create(&m_process_th,NULL,&slow_loop,this);
  if (res < 0)
  {
    printf("Error while creating thread for slow processing\n");
    m_process_th = 0;
  }
}
```

Both the slow and the fast parts of the object need to share the same data with each other. It is well known that the access of the same data from different threads is prone to variety of errors which are difficult to discover. Therefor, a unified interface for the data access was implemented here. If a part of data needs to be accessed by slow and fast parts of the object it is organized as a class derived from `ProtectedData` class presented below:

```
class ProtectedData
{
  template <class T, class P> friend class ControlledAccess;
public:
  virtual ~ProtectedData(){};
protected:
  ProtectedData();
private:
  /**Index in @ref CControlledObject::m_prot_parts. The derived
     ...class must provide
  static function GetNumber() which returns this index to allow
     ...ADD_PD macro to work*/
  static int m_number;
```

```
  /**Locking access to this object for @p obj*/
  bool Lock(CControlledObject* obj);
  /**Unlocking access to this object for @p obj*/
  void Unlock(CControlledObject* obj);

  /**Threads which locked the object*/
  pthread_t m_locked_threads[5];
  int m_num;

  /**Mutex which is used to control access*/
  pthread_mutex_t m_mutex;

  pthread_mutex_t m_help_mutex;
};


ProtectedData::ProtectedData()
{
  m_num = 5;
  int i;
  for (i=0;i<m_num;i++)
    m_locked_threads[i] = 0;
  pthread_mutex_init(&m_mutex,NULL);
  pthread_mutex_init(&m_help_mutex,NULL);
}

bool ProtectedData::Lock(CControlledObject* obj)
{
  pthread_mutex_lock(&m_help_mutex);
  int i;
  int i_0=-1;
  pthread_t self = pthread_self();
  for (i=0;i<m_num;i++)
  {
    if (pthread_equal(m_locked_threads[i],self) != 0)
    {
      pthread_mutex_unlock(&m_help_mutex);
      return false;
    }
    else if ((m_locked_threads[i] == 0) && (i_0 < 0))
      i_0 = i;
```

```
  }
  if (i_0 >= 0)
  {
    m_locked_threads[i_0] = self;
    pthread_mutex_unlock(&m_help_mutex);
    pthread_mutex_lock(&m_mutex);
    return true;
  }
  printf("locking, no free place\n");
  pthread_mutex_unlock(&m_help_mutex);
  return false;
}

void ProtectedData::Unlock(CControlledObject* obj)
{
  pthread_mutex_lock(&m_help_mutex);
  int i;
  pthread_t self = pthread_self();
  for (i=0;i<m_num;i++)
  {
    if (pthread_equal(m_locked_threads[i],self) != 0)
    {
      m_locked_threads[i] = 0;
      pthread_mutex_unlock(&m_mutex);
      break;
    }
  }
  pthread_mutex_unlock(&m_help_mutex);
  return;
}
```

The object of class `CControlledObject` has an array of `ProtectedData` and only these data can be accessed from inside the fast and slow parts of the object. The access is provided by creation of an object of class `ControlledAccess`:

```
/**
  \brief Provides controlled access to the protected parts of the
     ...object.

  T defines the class of the controlled object (derived from @ref
     ...CControlledObject),
```

```
  and P - class of its protected part to control (derived from @ref
     ... ProtectedData).
  Creation of the class object initiates the access, and its
     ...deletion terminates it.

  @author svetlana.larionova@gmail.com
  \ingroup control
*/
template <class T, class P>
class ControlledAccess
{
public:
  /**Initiates controlled access to the protected part (specified
     ...by P) of the object @p obj or its common object (@ref
     ...CControlledObject::m_common_obj) if it exists*/
  ControlledAccess(CControlledObject* obj);
  /**Terminates the access*/
  ~ControlledAccess();
  /**Provides access to the members of P*/
  P* operator->();
protected:
  /**Controlled object*/
  CControlledObject* m_obj;

  /**Object which asked for controlled access*/
  CControlledObject* m_cobj;
  int m_number;
};


template <class T,class P>
ControlledAccess<T,P>::ControlledAccess(CControlledObject* obj)
{
  if (obj == 0)
    return;
  m_cobj = obj;
  m_obj = obj->m_common_obj;
  if (m_obj == 0)
    m_obj = obj;

  m_number = P::GetNumber();
```

```
  if (m_obj != 0)
  {
    if (m_obj->m_prot_parts[m_number]->Lock(obj) == false)
      m_cobj = 0; // it was already locked by this thread
  }
}

template <class T, class P>
P* ControlledAccess<T,P>::operator->()
{
  return (P*)((T*)m_obj)->m_prot_parts[m_number];
}

template <class T, class P>
ControlledAccess<T,P>::~ControlledAccess()
{
  if (m_obj != 0 && m_cobj != 0)
    m_obj->m_prot_parts[m_number]->Unlock(m_cobj);
}
```

The access to a protected part is performed by creation of an object of `ControlledAccess` and terminets when the object is deleted. In between the data of the protected part can be only access through the `ControlledAccess` object using usual member access operators: `.` and `->`. The objects can initialize their protected data using the flowing function:

```
void CControlledObject::AddProtectedData(int n,ProtectedData* data)
{
  m_prot_parts[n] = data;
}
```

or the macro:

```
#define ADD_PD(cl) AddProtectedData(cl::GetNumber(),new cl);
```

C Demining Robot Communication Protocol

Command format:

:	ADR	CC	Data	CS	0x10	0x13

,

where ADR - 1-byte address of the device (0x01 for the robot)
CC - 1-byte command code
CS - 1-byte check sum (sum of all bytes: command code, data length and data, limited to one byte)

Commands:

0x04 Read multiply registers (according to Modbus specification)

0x10 Write multiply registers (according to Modbus specification)

0x41 Do step. Data = 2-byte step mode:
LSB:

```
- --- ----
   |  ||||___ step back
   |  |||____ step front
   |  ||_____ step left
   |  |______ step right
   |_________ number of sensor s
```

MSB:

```
________ 0 - step until sensor s starting from end-position
     |||___ complete step with half-step return
     ||____ step from sensor s
     |_____ step from current position
```

0x45 Calibrate foot sensors. Moves legs up and down and measures extreme values of foot sensors.

Robot memory:

Modbus address	Value (2 bytes)
0x0000	X coordinate high byte
0x0001	X coordinate low byte
0x0002	Y coordinate high byte
0x0003	Y coordinate low byte
0x0004	Azimuth
0x0005	Pitch
0x0006	Roll
0x000A	Axes end-sensors: $\overline{B2left}\|\overline{B2right}\|\overline{B1left}\|\overline{B1right}\|\overline{C2front}\|\overline{C2back}\|\overline{C1front}\|\overline{C1back}$
0x000B	Legs end-sensors: Leg is up Leg is down $\overline{8}\|\overline{7}\|\overline{6}\|\overline{5}\|\overline{4}\|\overline{3}\|\overline{2}\|\overline{1}$ $\overline{8}\|\overline{7}\|\overline{6}\|\overline{5}\|\overline{4}\|\overline{3}\|\overline{2}\|\overline{1}$

0x000D	Current step state:
	`____ ____ ____ ____`
	`|||| ||||  \_\__ current position of the body`
	`|||| |||____ moving back (or rotating left)`
	`|||| ||\_____ moving front (or rotating right)`
	`|||| |\_____ moving left`
	`|||| |_____ moving right`
	`||||_________ preparing for the step -`
	`|||  moving legs`
	`|||__________ preparing for the step -`
	`||  moving axes to the starting positions`
	`||__________ step is finished -`
	`|  waiting when all data are transmitted`
	`|___________ step was not finished (something blocked)`

0x0030	Metal detector radio signal
0x0031	Metal detector raw signal
0x0040	IR sensor 1
0x0041	IR sensor 2
0x0060	Sonar of leg 1
0x0061	Sonar of leg 2
0x0062	Sonar of leg 3
0x0063	Sonar of leg 4
0x0064	Sonar of leg 5
0x0065	Sonar of leg 6
0x0066	Sonar of leg 7
0x0067	Sonar of leg 8

D Suspicious objects detection implementation

Detection of suspicious objects is implemented using several classes:

- `CKalmanFilter` - implementation of iterative filter
- `CTimeROIDetector` - detection of interesting points using two iterative filters: providing the value for the *Segmented Map*
- `CGridROIExtractor` - implementation of extrema searching and region growing
- `CStoreGridMap` - data storage for ROI grid maps: *Data Map*, *Object Area*, *Segmented Map*
- `CPlaneGridMap` - data storage for the grid map obtained from the sensor data during scanning; implementation of the grid regularisation
- `CPlaneSegmGridMap` - data storage for the grid map where the *Segmented Map* value obtained by `CTimeROIDetector` is mapped spatially
- `CObjectsAssociation` - implementation of the object association algorithm
- `CROIsCollection` - collection of ROIs which represent the same object as obtained after object association

This appendix contains short descriptions of the main functions according to the stages of the algorithm. The program listing can be found in the project *lowpr* on the attached CD.

D.1 Detection of interesting points

Detection of interesting points is performed by the class `CTimeROIDetector` which is derived from `CDecisionMaker`. The used sensor value (`m_sensors[0]`) is first processed by two iterative filters implemented by the class `CKalmanFilter`. The parameter `m_threshold` is initialized to the same value as the system noise of the fast filter. The main processing is performed by the function:

```
template <class T> void CTimeROIDetector<T>::ProcessRobot(CRobot*);
```

D.2 Extrema searching

Extrema searching is performed by the class `CGridROIExtractor` inside the *Segmented map* whose pointer is stored in protected data `Data::m_segm_map`. The following functions performs the extrema searching:

```
template <class MT, class ST>
void CGridROIExtractor<MT,ST>::SlowPart::FindMinMax(
        MaxPoint* maxpoints, // array of extrema
        int end_yi); // maximum Y
```

D.3 Region growing

During region growing the border of the object being detected is followed in order to determine new segments which might be joined. The following function of `CGridROIExtractor` performs the core processing of region growing.

```
template <class MT, class ST>
int CGridROIExtractor<MT,ST>::SlowPart::
  FollowBorderAndFindMoreROIs(
        CStoreGridMap<MT>* storemap, // Object Area of the
            ...processed ROI
        MT* count,int end_yi);
```

The following function of `CGridROIExtractor` determines a region of the same value inside the *Segmented Map*.

```
template <class MT, class ST>
int CGridROIExtractor<MT,ST>::SlowPart::FindSameValue(
        MT val, // value to detect
        int i,int j, // starting point of search
        int* i_min,int* i_max, // limits of the found region
        int* j_min,int* j_max, // limits of the found region
        CStoreGridMap<MT>* storemap, // Object Area map
        MT mapvalue, // value to set to Object Area for the
            ...detected region
        int end_yi, // maximum Y
        bool is_wait_for_valid);
```

E Suspicious objects database

The detected suspicious objects are stored in the database using the following table hierarchy:

rois_groups stores data of objects: name of data file, index of sensor group (*sensors_groups.indx*), index of experiment (*experiments.indx*)

sensor_types defines the types of sensors

sensors defines the used sensors according to their types

experiments define parameters of the experiments at which the data were obtained

expr_sensors define parameters of the experimental data for a particular sensor (*sensors.indx*) and experiment (*experiments.indx*)

The objects are stored in text-files according to the following format:

1. Number of sensors whose ROIs are represented in the file (`Number of sensors`)
2. For each sensor:
   - Type of sensor, which corresponds to *sensor_types.address*+*sensors.indx_in_type* (`Sensor indx`)
   - Index of experimental data from which this ROI was obtained, corresponds to *expr_sensors.indx* (`DB expr_sensor index`)
   - Number of maps for this ROI, usually 3 which means: Raw Map, Object Area, and Segmented Map (`Layers`)
   - Coordinates of the ROI:
     `XStart=...  XStep=...  XNum=...`
     `YStart=...  YStep=...  YNum=...`
   - For each map: data of the map where columns correspond to X and rows correspond to Y

An example of a data file is presented below:

```
Number of sensors 2
Sensor index 66
DB expr_sensor index 14
Layers 3
XStart=1700.000000 XStep=10.000000 XNum=9
YStart=4270.000000 YStep=10.000000 YNum=5
  37751   36847   37938   37867   37218   37120   37623   37902   37396
  37899   37077   37062   37063   37851   37889   38107   37288   37188
  38544   37110   37187   37021   36942   36751   37343   36576   36741
  38332   37271   37104   36757   37070   37153   37059   37093   37004
  38132   36462   36838   37001   37122   37022   36879   37126   36875
      0       0       0       0       0       0       0       0       0
      0       0       0       1       1       1       1       1       1
      1       1       1       1       1       1       1       1       1
      0       0       0       1       1       1       1       1       1
      0       0       0       0       0       0       0       0       0
      1       1       1       1       1       1       1       1       1
      1       1       1      15      15      15      15      15      15
      5       5       3       3       1       1       1       1       1
     91      91      91       1       1       1       9       9       9
      1       1       1       1       1       1       1       1       1
Sensor index 49
DB expr_sensor index 156
Layers 3
XStart=1710.000000 XStep=10.000000 XNum=9
YStart=4250.000000 YStep=10.000000 YNum=4
  32764   32764   32765   32765   32765   32767   32767   32769   32768
  32765   32765   32765   32766   32766   32767   32767   32767   32768
  32765   32766   32766   32766   32766   32768   32768   32770   32769
  32769   32770   32770   32771   32772   32773   32773   32774   32775
      0       0       0       0       0       0       0       1       1
      0       0       0       1       1       1       1       1       1
      0       0       0       0       0       1       1       1       1
      1       1       1       1       1       1       1       1       1
      1       1       1       1       1       1       1      12      12
      1       1       1      12      12      12      12      12      12
      1       1       1       1       1       1      12      12      12
      1       1       1       1      12      12      12      12      12
```

The suspicious object database created during this work can be found on the attached CD. In order to view the database or to use it for the testing of landmine detection algorithms the interface program *inttest* should be used with the project file *fa.prj* located on the CD.

F Landmine recognition Implementation

The following classes were developed in order to implement the landmine detection algorithms proposed in this work:

- The classification features are implemented by the classes derived from `CGridMapFeature`, for sensor based features, and from `CROIsCollectionFeature`, for multi-sensor features. The combined features are implemented by `CAverageFeature` class. Each particular feature is implemented in a separate class (for example, `CFractalDimentionFeature`, `CGRMeasureFeature`). The features are calculated on the objects of `CStoreGridMap` class obtained after suspicious object detection algorithm (see Appendix D)
- `CFeaturesCollection` - represents a collection of features used for the classification
- `CAdditiveBayesClassifier` - implementation of the combined classifier which includes the possibility to use the concepts of *selective training* and *dominant class*
- `CDecisionTree` - implementation of a decision tree classifier
- `CFeatureEvaluation` - implementation of the feature analysis techniques which wee used in this work (this class is used by the interface program *inttest* to allow the feature analysis)

The classifiers have the following main functions to perform all the processing:

- Training of the classifier is performed by:

```
void Train(CFeaturesCollection** fc, int fc_num);
```

- When an object has to be classified the following function is used:

```
void Classify(CFeaturesCollection** fc, int fc_num);
```

The program listing for these classes can be found in the project *lowpr* on the attached CD.

G Implementation of vision- and odometry-based positioning system

The odometry of the robot is implemented in the low-level program which run on the Motorola MC12 microcontroller located on the robot. This program contains the developed algorithms for improvement of the robot movements (compensation for uneven speeds of parallel cylinders and adjusting the ground contact). The full listing of the program, named *dprog_gcc*, can be found on the attached CD.

The vision based localization system is implemented in the control program *lowpr* as a set of classes:

- The detection of natural landmarks located on the ground is performed by the class `CLandmarkDetector` which is derived from the previously developed ROI extraction algorithm (class `CGridROIExtractor`).
- `CCorrelationFeature` - calculation of the correlation between landmarks
- `CShapeCorrelationFeature` - calculation of the shape correlation between landmarks
- `CContourCorrelationFeature` - calculation of the contour correlation between landmarks, while the contour of the object is obtained by

```
int CStoreGridMap::GetObjPerimeter(bool is_count);
```

- `CCombinedCorrelationFeature` - calculation of the combined measure used for the landmark association
- `CKalmanFusion` - implementation of the Kalman sensor fusion for combining the vision and the odometry systems

The program listing for these classes can be found in the project *lowpr* on the attached CD.

H Complete coverage implementation

The following classes are used for the implementation of the developed complete online coverage algorithm:

- `CExplorationBeh` - top-level behavior of the robot; this behavior contains other, lower-level, behaviors and controls the robot switching between them in the function:

```
void CExplorationBeh::NextMotion(int state,CRobot* rob);
```

- `CWallMotion` - wall-following behavior
- `CForwardBeh` - directed wall-following behavior (is implemented as a top-level behavior for `CForwardMotion` and `CWallMotion`)
- `CFindObstacle` - find obstacle behavior (is implemented as a top-level behavior for `CForwardMotion` and `CWallMotion`)
- `COccupancyGrid` - implementation of the occupancy grid map where the sonar readings are mapped in order to be used for detection of critical points
- `CAssociatedED` - the incremental cellular decomposition performed by the coverage algorithm stored as a map
- `ACellEndAfterWallDetector`, `ACellEndDetector`, `AWallDirDetector` - decision making in order to determine the finishing of the current cell

The program listing for these classes can be found in the project *lowpr* on the attached CD.