# 論文 / 著書情報
# Article / Book Information

| | |
|---|---|
| 題目(和文) | |
| Title(English) | Landmine Detection Rate andMetal Fragment Discrimination Performance Improvement Based on Robotic ArmScanning and Spatially Represented MetalMine Detector Signal Characterization |
| 著者(和文) | ｶﾈｺｱﾚｯｸｽ ﾏｽｵ |
| Author(English) | ALEX MASUO KANEKO |
| 出典(和文) | 学位:博士(工学),<br>学位授与機関:東京工業大学,<br>報告番号:甲第9454号,<br>授与年月日:2014年3月26日,<br>学位の種別:課程博士,<br>審査員:福島 E　文彦,大熊　政明,小田　光茂,松永　三郎,塚越　秀行 |
| Citation(English) | Degree:Doctor (Engineering),<br>Conferring organization: Tokyo Institute of Technology,<br>Report number:甲第9454号,<br>Conferred date:2014/3/26,<br>Degree Type:Course doctor,<br>Examiner:,,,, |
| 学位種別(和文) | 博士論文 |
| Type(English) | Doctoral Thesis |

# Landmine Detection Rate and Metal Fragment Discrimination Performance Improvement Based on Robotic Arm Scanning and Spatially Represented Metal Mine Detector Signal Characterization

A thesis presented

by

Alex Masuo Kaneko

to

The Department of Mechanical and Aerospace Engineering

in partial fulfillment of the requirements

for the degree of

PhD of Engineering

Tokyo Institute of Technology

Tokyo, Japan

December 2013

Thesis supervisors
**Edwardo F. Fukushima**

Author
**Alex Masuo Kaneko**

# Landmine Detection Rate and Metal Fragment Discrimination Performance Improvement Based on Robotic Arm Scanning and Spatially Represented Metal Mine Detector Signal Characterization

## Abstract

Landmines continue killing and injuring people even many years after the end of conflicts. It is estimated that there are 100 million mines buried in about 60 countries in the world. Current neutralization methods are costly, tedious, dangerous, time-consuming (centuries are required for complete clearance) and suffer from high false alarm rates (each 1000 alarms, only 1 is actually a mine).

Tokyo Institute of Technology developed a semi-autonomous mobile robot, called Gryphon, to assist the mine detection process. Its manipulator is able to automatically scan over rough terrain, record data and present the resulting sensor images to the operator who then can mark suspect spots. However, as most of the demining solutions based on metal mine detectors, Gryphon suffers from high False Alarm Rates, what makes necessary accurate and fast methods for discriminating landmines and metal fragments.

First, enhancements of landmine detection and marking tasks of the Demining Robot Gryphon in terms of time duration and human factors are presented. New landmine detection and marking methods were analyzed, and aiding tools were implemented in the user interface, such as: a) filters for enhancing targets visualization in the 2D graph and b) friendly interface for automatically pointing and marking detected targets, previously done manually by

human operators. Experimental results showed that time efficiency and operator workload were greatly improved compared to the former methods.

Second, this thesis introduces a new, fast, accurate and on-site method for metal fragments and landmines discrimination based on Spatially Represented Metal Mine Detection Signal. The method is simple, fast and powerful since metal detector signals are simplified into polynomials which are stored in a database and used for searching. Experimental results showed that the method can estimate depth with average error of 4 mm and maximum 39 mm. Using data from test fields in Croatia, FAR can be reduced to 50% without risks of False Negatives, what is achieved by setting safety margins. The estimation and discrimination are performed in less than 1 s, which comparisons with the database are done linearly permitting parallel searching, what can be said fast and suitable for demining operations.

Further improvements were also achieved by optimizing the scan and marking trajectory, greatly reducing time of the demining operation. Enhancements in the robot's vision were also verified with natural sunlight experiments, which features in images were enhanced in until 9% and generated depth maps in 29% with an on-site method, making it possible for the robot to accurately scan areas with light contrasts. An interface for capturing GPS data and integrating with databases was also developed and tested.

# Acknowledgments

I would like to thank Professors Edwardo Fukushima, Gen Endo and Shigeo Hirose for giving me the honor of joining Hirose Fukushima Lab and for supervising the project, guiding me with knowledge and patience. A special thanks to Professor Fukushima for following all the steps of the project and providing all necessary support.

Thanks to all staff, colleges and friends of the laboratory (including all ex-members who I had the pleasure of meeting), for providing a pleasant environment for research and stay in Japan.

I am also grateful for all the staff who were constantly working backstage for supporting our student life: Student Support Division, Student Division and others.

To my family, for the constant support and dedication (no words would be enough to describe my gratitude).

To the Japan International Cooperation Agency (JICA) and the Ministry of Education, Culture, Sports, Science and Technology (MEXT) for providing financial support for my study program.

Finally, my deepest gratitude to Emeritus Professor Kokei Uehara and Professor Paulo E. Miyagi from University of Sao Paulo (USP), who were great influences and motivation for me to come to Japan.

Five years have passed since I came to this exotic place called Japan, located in the very opposite side of the world from my home country Brazil. I knew it would be unforgettable times, but could never imagine it would actually be such a whole new and exciting life full of adventure and experiences.

First, as a naive and young boy suddenly getting to live alone in a foreign country, facing all the simple difficulties of daily life was already a challenge. (I **finally** noticed how great my parents are for doing these tasks every single day for me...).

Second, post-graduation itself is not something anyone can handle easily. Differently from other degrees, much more dedication, hard-work and somehow talent are necessary, or else one will never be able to graduate...

Few people know that less than 65% of all PhD students in the world can in fact graduate even after 10 years! Less than 8% can graduate within 3 years, the considered “ideal period”! May this PhD thesis be one small of the many achievements I acquire in my life.

Third, as a Brazilian student, I did my best to cope with both study life and volunteer activities of promoting cultural exchange between Brazil, Japan and other countries. I had the satisfaction of joining activities, which some of them I will briefly mention here: a) Tohoku area restoration, b) English tutor for kids, c) Portuguese teacher, d) NPO member, Counselor for supporting Brazilian community in Japan and for introducing Brazilian culture to the Japanese. I met many interesting people and learned principles and values that will definitely be fundamental in my life.

Furthermore, even though it is Japan, I had the honor of meeting people from many other countries, which I believe expanded my view and understanding of the world. Only in our laboratory, I could meet amazing people from: **Brazil, China, Germany, India, Italy, Japan, Korea, Mexico, Netherlands, Scotland, Switzerland, Russia, Taiwan, Tunisia and United States**.

Moreover, during all my stay in Japan, I also had the pleasure of meeting people from: **Angola, Australia, Austria, Belize, Bulgaria, Cameroon, Canada, Chile, Colombia, Cuba, Ecuador, Egypt, France, Honduras, Hungary, Indonesia, Jamaica, Malaysia, Mongolia, Mozambique, Myanmar, Pakistan, Paraguay, Peru, Portugal, Philippines, Romania, Saudi Arabia, Singapore, Spain, South Africa, Syria, Thailand, Turkey, Uruguay, Venezuela, Vietnam and Zambia**. I learned a lot (even about my own country!) by explaining and exchanging opinions with all these people. Now I comprehend and agree with the necessity of people actually understand each other and transcend all cultural barriers. The world is really huge, and unfortunately our knowledge about each other is still too small. Here I conclude a so called "high academic degree" (PhD), but there is still too much to learn...

Finally, I could make many amazing discoveries in Japan besides robotics (my primary research topic) and culture. After 5 years here I noticed that my greatest discovery was in fact about **myself**!

# Contents

Table of Contents . . . . . ix
List of Figures . . . . . xiii
List of Tables . . . . . xix

**1 Introduction** **1**
1.1 Structure of the Thesis . . . . . 2
1.2 Background on Humanitarian Demining . . . . . 4
1.2.1 Landmines . . . . . 5
1.2.2 Humanitarian Demining . . . . . 7
1.2.3 Mechanization and automation . . . . . 9
1.3 The POD and FAR Problem in Demining Operations . . . . . 10
1.3.1 Existing Approaches for Reducing FAR . . . . . 12
1.3.2 Gryphon POD and FAR . . . . . 13

**2 Description of the Robotic Demining System Gryphon** **15**
2.1 Mobile Platform . . . . . 15
2.2 Manipulator . . . . . 17
2.3 Stereo Vision Camera . . . . . 17
2.4 Metal Detector . . . . . 18
2.5 Marking System . . . . . 20
2.6 GPS . . . . . 21
2.7 Control Architecture . . . . . 22
2.8 Analysis and Limitations of the System . . . . . 26
2.8.1 Demining Tasks with Gryphon . . . . . 26
2.8.2 Mapping Task . . . . . 28

2.8.3 Scanning Task . . . . . 29
2.8.4 Marking Task . . . . . 29
2.8.5 System Analysis . . . . . 30

**3 Targets Detection Method Enhancements 33**
3.1 Landmine Detection Difficulties . . . . . 33
3.2 Sensor Pre-processing and Filtering . . . . . 34
3.2.1 Proposed Filters . . . . . 35
3.2.2 Filter Evaluation . . . . . 40

**4 Targets Marking Method Enhancements 43**
4.1 Targets Perimeter Searching Algorithm . . . . . 43
4.2 Targets Center Searching Algorithm . . . . . 44
4.3 Proposed Method Experiment Results . . . . . 44
4.4 Marking Trajectory and Sequence Enhancement . . . . . 45

**5 Proposed Curve Characterization Method 51**
5.1 Method Description . . . . . 51
5.1.1 Spatially Represented Metal Mine Detector Signals . . . . . 51
5.1.2 Main Axis and Main Characteristic Curves Definition . . . . 53
5.1.3 Searching Criterion . . . . . 55
5.1.4 Database Building . . . . . 55
5.1.5 Depth Interpolation for Characteristic Curves . . . . . 56
5.1.6 Metal Mine Detector Signal Conditioning . . . . . 57
5.1.7 Database Integrity . . . . . 59
5.2 Landmine/Metal Fragment Discrimination Method . . . . . 62
5.2.1 Basic Discrimination Scheme . . . . . 62
5.2.2 Practical Discrimination Process . . . . . 62
5.3 Experiments . . . . . 67
5.3.1 Material and Depth Estimation . . . . . 68
5.3.2 Landmine and Metal Fragment Discrimination . . . . . 70
5.3.3 Database Searching Time . . . . . 74
5.4 Experiments With Noisy Data . . . . . 74
5.4.1 Types of Noise . . . . . 74

5.4.2 Experiment Methodology . . . . . 77
5.4.3 Depth Estimation Analysis . . . . . 78
5.4.4 FAR Analysis . . . . . 78

**6 Further Enhancements in the System** **81**
6.1 Trajectory and Scan Line Step Optimization . . . . . 81
6.1.1 Scan Line Analysis for Targets Detection . . . . . 82
6.1.2 Scan Line Step Analysis for Discrimination . . . . . 83
6.1.3 New Scan Procedure . . . . . 84
6.2 New Artificial Vision Technique for Extreme Sunlight Conditions . 85
6.2.1 Implemented Solution . . . . . 87
6.2.2 Natural Sunlight Experiments . . . . . 90
6.3 GPS Signal and Map Integration . . . . . 99
6.3.1 Interface Enhancements for Database Integration . . . . . 103
6.3.2 Interface Experiment . . . . . 103

**7 Conclusions** **107**
7.1 Future Work . . . . . 109

**Bibliography** **113**

**A Implemented Algorithms** **119**
A.1 Transforming MMD Raw Signals Into Polynomials . . . . . 119
A.2 Discriminating Signals In the Database . . . . . 133
A.3 Program for Comparing Polynomials in Database . . . . . 145
A.4 Program for Setting and Calculating FAR from Database . . . . . 154
A.5 Calculating the Median . . . . . 157
A.6 Calculating the Half-peaks . . . . . 158
A.7 Potential Targets Searching Algorithm . . . . . 158
A.8 Local Maximum Searching Algorithm . . . . . 159

**B Achievements** **161**
B.1 Papers in refereed journals . . . . . 161
B.2 Conferences . . . . . 161
B.2.1 Refereed . . . . . 161

B.2.2 Non-refereed . . . . . 162

# List of Figures


1.1 Countries affected by landmines . . . . . 2
1.2 Landmines victims . . . . . 3
1.3 AT M15 landmine . . . . . 5
1.4 AP M16 landmine . . . . . 5
1.5 Difference in size between APs and ATs . . . . . 6
1.6 Dog for landmine detection . . . . . 8
1.7 Rat for landmine detection . . . . . 8
1.8 Conventional MMD used in demining operations . . . . . 8
1.9 Demining robot Pemex . . . . . 10
1.10 Walking robot TITAN . . . . . 10
1.11 ODIS platform . . . . . 10
1.12 Demining solution proposed by Yamanashi Hitachi (top) and possible combination with Gryphon for enhancing detection capabilities (bottom) . . . . . 11
1.13 Procedures taken for removing landmines. Operators lay down on the ground, dig objects slowly and with caution until they are completely removed . . . . . 12
1.14 Gryphon's POD and FAR obtained in experiments in testfields . . 14

2.1 Overview of Gryphon's main elements . . . . . 16
2.2 ATV for Gryphon's locomotion . . . . . 16
2.3 Arm model (side view) . . . . . 17
2.4 Arm model (top view) . . . . . 17
2.5 Wrist model (with MMD) . . . . . 18
2.6 Stereo vision camera attached to Gryphon . . . . . 18

2.7 Minelab F3, recently most used MMD in Gryphon . . . . . 19
2.8 Marking based on plate dispenser . . . . . 20
2.9 Marking based on paint . . . . . 21
2.10 Base antenna (immobile) . . . . . 23
2.11 Rover antenna (mobile) . . . . . 23
2.12 Trimble receiver used with Gryphon . . . . . 23
2.13 Yaesu modem for RTK . . . . . 23
2.14 Remote control unit . . . . . 23
2.15 Overview of the control architecture of Gryphon . . . . . 24
2.16 Overview of the Gryphon user interface "Visu" . . . . . 25
2.17 Flowchart of the demining operation with Gryphon . . . . . 27
2.18 Gryphon's main tasks and its sub categories . . . . . 27
2.19 Mapping task . . . . . 29
2.20 Scanning task . . . . . 30
2.21 Marking task . . . . . 31
2.22 Gryphon time duration analysis for three different methods . . . . 32

3.1 Example of signal and different filters . . . . . 34
3.2 Lines of a scanned area . . . . . 35
3.3 Resulting images of the filtering methods: a) Basic . . . . . 35
3.4 b) Half-peaks offset . . . . . 36
3.5 c) Half-peaks . . . . . 36
3.6 d) Median of an entire area . . . . . 36
3.7 e) Median of each line in an area . . . . . 37
3.8 Half-peaks algorithm . . . . . 38
3.9 Database information . . . . . 39
3.10 Algorithm of the best pre-evaluated filter "e" . . . . . 40
3.11 Experiment results with 95% confidence limits . . . . . 41

4.1 Examples of target images . . . . . 44
4.2 Targets perimeter searching algorithm . . . . . 45
4.3 Targets center searching algorithm . . . . . 46
4.4 Resulting image after applying the targets perimeter and targets center searching algorithms . . . . . 46

4.5 Comparison between the best result in the test field and automatic marking system . . . . . 47
4.6 Final output image displayed to the operator . . . . . 47
4.7 Trajectory between two consecutive marks . . . . . 48
4.8 Trajectory enhancement . . . . . 48
4.9 Sequence enhancement . . . . . 50
4.10 Sequence enhancement test . . . . . 50

5.1 Demining robot Gryphon and detected signals . . . . . 52
5.2 SRMMDS for different targets and conditions . . . . . 53
5.3 Cutting plane and main characteristic curves . . . . . 54
5.4 Targets used for building the database . . . . . 56
5.5 Example of depth interpolation for an AT and MF21 target type characteristic curves . . . . . 57
5.6 Database data depth error distribution . . . . . 60
5.7 Targets with 30-40 mm depth errors . . . . . 61
5.8 Basic discrimination cases (R1, R2, R3 and R4) and target distances according to *Err* . . . . . 63
5.9 Examples of MFs considered as "potential mines" by the $E_{closest}$ and $|dE| < dE_{threshold}$ criteria. Targets and corresponding depths are shown in parenthesis. Note that the ITOP conceived for an International Test Operations Procedure project as the metal content of larger stimulant mines shows a SRMMDS very similar to some landmines (such as PMN2) and it is also classified as "potential mine" by this criteria . . . . . 65
5.10 Resulting Errors of closest MFs and mines from each data. According to the adopted safety margins $dE_{threshold}$ and $E_{threshold}$ different FAR can be observed . . . . . 66
5.11 Variation of False Positive and True Positive values according to $E_{threshold}$ (discrete case) . . . . . 66
5.12 Variation of False Positive and True Positive values according to $E_{threshold}$ (interpolated case) . . . . . 67

5.13 Trade-off of adopted safety margins and FAR. For all cases, FAR is generated with no occurrence of False Negatives due to the discrimination criteria and safety margins adopted . . . . . 68
5.14 Target with maximum depth error (top) and examples of ITOP wrong estimation (bottom) . . . . . 70
5.15 Examples of FAR: fragment discriminated as "potential mine" (left) and fragment discriminated as "landmine" (right). Depth of each target is shown in parenthesis. MFX and MFY are two MFs from the test field, which size, shape and material are not available . . . 73
5.16 Average time for estimation of a target per number of data in the database (worst case) . . . . . 73
5.17 Noise caused by soil . . . . . 75
5.18 Noise balanced without changing its features. “Median” and “Ground Balancing” result in very similar offsets. In relation to the median method, the GB has an offset difference of 0.01%, while for no GB, -0.21% . . . . . 76
5.19 Experiments with targets in soil . . . . . 76
5.20 Noise caused by electrical devices . . . . . 77
5.21 Data with White Gaussian Noise . . . . . 79
5.22 Maximum depth error cases for each target, with depths in parenthesis. Blue lines represent noisy inputs and red ones the closest data in the database . . . . . 79
5.23 Resulting FAR according to adopted ST levels . . . . . 80

6.1 Auto-detection results of ITOP target at 3 cm depth, for various line steps . . . . . 82
6.2 Average maximum signal decay in function of scan steps for different $MMD_{max}$ ranges . . . . . 83
6.3 Obtained errors between targets in 0 and $135^o$ for each scan line step . . . . . 84
6.4 Proposed new scan procedure . . . . . 85
6.5 Demining time with Gryphon with the former and new scan procedures . . . . . 86

6.6 Effect of strong light contrast in the map calculation. Camera's left-right pair images (top) and calculated depth map displayed in the user interface (bottom) . . . . . 87
6.7 Six images with different exposures used for computing the compensated image in laboratory environment . . . . . 89
6.8 Implemented algorithm: compensated (left top) and non-compensated image (right top). Resulting depth map displayed in the user interface (middle images) and detected features for each case (bottom) . . . . . 91
6.9 Condition A: asphalt. No contrast images, detected features with SURF and calculated depth map (top); contrast images, detected features with SURF and calculated depth map shown in the user interface (bottom) . . . . . 95
6.10 Condition B: low vegetation with fallen leaves. No contrast images, detected features with SURF and calculated depth map (top); contrast images, detected features with SURF and calculated depth map shown in the user interface (bottom) . . . . . 96
6.11 Condition C: middle size vegetation. No contrast images, detected features with SURF and calculated depth map (top); contrast images, detected features with SURF and calculated depth map shown in the user interface (bottom) . . . . . 97
6.12 Condition D: soil. No contrast images, detected features with SURF and calculated depth map (top); contrast images, detected features with SURF and calculated depth map shown in the user interface (bottom) . . . . . 98
6.13 Six images with different exposures used for computing the compensated image in outdoors experiments . . . . . 99
6.14 Scan failure with the non-compensated image (bottom) and scan successfully performed with the compensated image (top). Depth maps detected by the camera (left) and calculated (including artificially filled empty spots) possible scan regions (right) . . . . . 100
6.15 Experiments results. Increases in features (blue) and depth maps (red) from non-compensated to compensated images . . . . . 101

6.16 Implemented interface for capturing GPS signals . . . . . 101
6.17 GPS signal plotted in database by KML file . . . . . 103
6.18 GPS experiments done nearby Mt. Mihara in Izu Oshima, Japan . 104
6.19 Hand GPS used for comparisons . . . . . 104
6.20 GPS data comparison between Gryphon' GPS (yellow) and a hand GPS (blue) . . . . . 105

7.1 Generated signal from two different targets scanned close to each other . . . . . 110
7.2 Chart obtained as sub product of the proposed discrimination method showing all metal fragments and landmines regions. Signals with similar features (similar polynomials) are located very closely . . . . . 111
7.3 General scheme of the self-calibration method: Internal/external points and sensors . . . . . 112

# List of Tables


2.1 Mapping task details . . . . . 28
2.2 Scanning task details . . . . . 30
2.3 Marking task details . . . . . 31

3.1 Filters comparative table . . . . . 39

4.1 Average time per $m^2$ . . . . . 44
4.2 Marking trajectory and sequence enhancements evaluation . . . . 50

5.1 Dimensions of the targets used for building the database . . . . . 58
5.2 Database integrity: material and depth estimation results . . . . . 60
5.3 Discrimination cases according to *Err (%)*. . . . . 64
5.4 Depth and material estimation experiment results . . . . . 70
5.5 Parameters adopted and results of the proposed method . . . . . 72
5.6 Depth estimation results . . . . . 80

6.1 Number of features detected in laboratory experiments . . . . . 99
6.2 Natural sunlight experiments average results . . . . . 102

# List of Abbreviations

| | |
|---|---|
| AP | Anti-Personnel |
| AT | Anti Tank |
| ATV | All Terrain Vehicle |
| FAR | False Alarm Rate |
| GPR | Ground Penetrating Radar |
| GPS | Global Positioning System |
| GUI | Graphical User Interface |
| MMD | Metal Mine Detector |
| POD | Probability of Detection |
| RTK | Real Time Kinematics |

# Chapter 1

# Introduction

Landmines are prominent weapons: effective, cheap, easy to make and lay. Mines can kill, injure, deny access to lands and its resources, and social and economic consequences are immense. However, the major malefaction of landmines is the fact the they remain active even after wars have ceased, killing and injuring innocent civilians.

It is estimated that worldwide (more than 68 countries, shown in Fig. 1.1) there are about 100 million landmines scattered, causing 26.000 victims every year (Fig. 1.2). Moreover, it is said that currently 2 to 5 millions of new mines continue to be laid every year, which is a much faster rate if compared to the clearance.

Demining operations are still very ineffective and costly. While the production of a mine can be around 3 to US$30, the cost for medical treatment and rehabilitation exceeds US$750 million, which is still a small fraction compared to clearing the existing mines [18].

Most of the existing solutions (which will be described in the next section) are based on Metal Mine Detectors (MMD), but the False Alarm Rate (FAR) is still too high. Tokyo Institute of Technology developed a robot called Gryphon (described in details in Chapter 2) for assisting in demining operations. According to experiments done in real minefields [41], the robot has the potential to be "as good as a human deminer", but it still suffers from FAR.

This thesis proposes new landmine detection, marking system and a fast

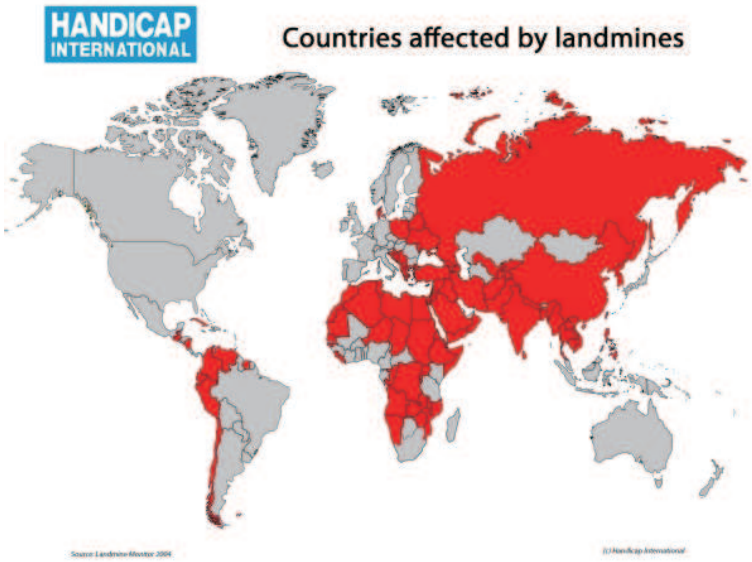


**Figure 1.1:** *Countries affected by landmines*

and accurate method for landmine and metal fragment (MF) discrimination for MMDs based on Curve Characterization. The method takes advantage of high precision scanning of the minefield using robotic manipulator, resulting in great decrease of FAR.

## 1.1 Structure of the Thesis

The thesis is structured to gradually introduce the reader through the very basic concepts of demining operations until the main achievements obtained in the work:

Chapter1: Introduction. Presents the structure of the thesis,very basic oncepts and information related to demining operations, such as landmines types and working principle, humanitarian demining and some existing solutions (using animals, metal detector, machines and robots). The limitations of these solutions are presented and the importance of Gryphon is highlighted, followed by the motivations for this thesisof enhancing landmine detection, marking and discrimination.

Chapter2: Description of the Robotic Demining System Gryphon. This chap-

**Figure 1.2:** *Landmines victims*

ter describes the demining robot Gryphon and its main parts: mobile platform (for carrying Gryphon along hard conditions of minefields), manipulator (for moving landmine sensors and tools near the ground), stereo vision camera (for detecting terrain information), metal detector (the main sensor used with Gryphon), marking system (for clearly pointing the location of detected landmines), GPS (for localizing itself along the field) and control architecture.

Chapter3: Targets Detection Method Enhancements. Introduces the main difficulties for landmine detection and the proposed solution in the project. The solution consists of implementing and evaluating different filters for processing the metal detector signal and permitting easier visualization by a human operator.Several filters based on different approaches such as median, moving window, half-peaks, etc were implemented and evaluated by some subjects, who marked potential targets after the signals being submitted to the filters.

Chapter4: Targets Marking Method Enhancements. Once targets can be detected more efficiently, new methods for marking with higher efficiency are proposed. First, an algorithm for searching the perimeter of the signals of detected targets is proposed, followed by a targets center searching algorithm. For

increasing time efficiency, improvements in the sequence and trajectory of the robot's manipulator during the marking task were also done. Besides pointing more accurately the correct targets and their centers, human operator's workload and operation time duration were greatly reduced, eliminating possible human factors.

Chapter5: Proposed Curve Characterization Method. This chapter presents the main landmine/metal fragment discrimination method developed in this project. Basic definitions as Spatially Represented Metal Mine Detector Signals and its Main Axis are introduced. All necessary steps for the method are detailed, such as the searching criterion, database, depth interpolation method, metal detector signal conditioning and database integrity experiment. Experiments in laboratory, soil, using data from test fields and inputting noisy data were also conducted, showing the robustness and potential of the method.

Chapter6: Further Enhancements in the System. Besides the main goals, some further enhancements were achieved in the system. First, the scan line step for a reliable detection is investigated, showing that the scan can be done in two steps, fast and reliably. Second, a new artificial vision technique for extreme sunlight conditions is presented. This enhancement permits the camera to correctly detect features in light contrast areas, increasing the reliability of demining operations. Finally an interface for integrating GPS signals with databases was developed and tested.

Chapter7: Conclusions. A summary of all achievements is done followed by the main results obtained. According to the experiments, the proposed landmine detection and marking methods reduce greatly workload and time of demining operations. Moreover, the method permits quick landmines and metal fragments discrimination with lower FAR, as well as accurate depth and material estimation. No false negatives can happen by setting correct safety margins, increasing the potential of the robot in real operations.

## 1.2 Background on Humanitarian Demining

This section provides some basic information about demining, such as landmines types, humanitarian demining, ladmine sensors, and examples of mech-

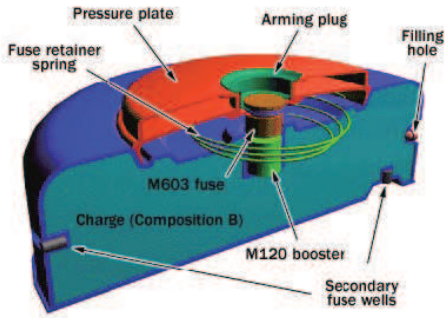


**Figure 1.3:** *AT M15 landmine*

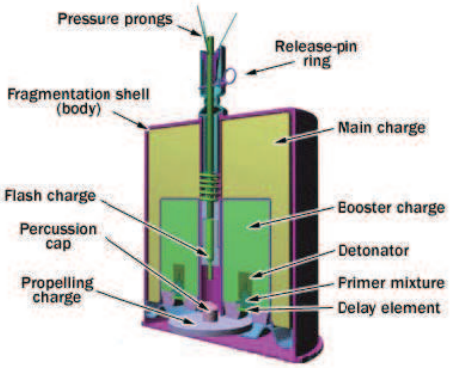


**Figure 1.4:** *AP M16 landmine*

anized solutions.

### 1.2.1 Landmines

Landmines exist in different sizes, shapes, colors and can be made by different materials (metal, plastic, wood, etc). Mines are usually laid in groups (minefields), to avoid enemies to enter certain areas, or to force them to pass through a specific region for tactical purposes. It is also possible to find a single landmine or small groups of them into an area, which is usually the worse scenario for a deminer. Even if a very large number of landmines exist today, they can be roughly divided in two main groups:

- Anti-tank mines (AT) (Fig. 1.3)

- Anti-personnel mines (AP) (Fig. 1.4)

Even though their functional behavior is the same, they are aimed at different targets, which requires some basic differences between them. AT mines are usually bigger and contains several times more explosive than AP ones (Fig. 1.5, taken from [49]), in order to destroy a tank or a car. The pressure required to trigger them is usually much higher than the one required for AP mines; a human body is usually not able to trigger it.

The cylindrical fuse is made of iron, which is the easiest (sometimes the

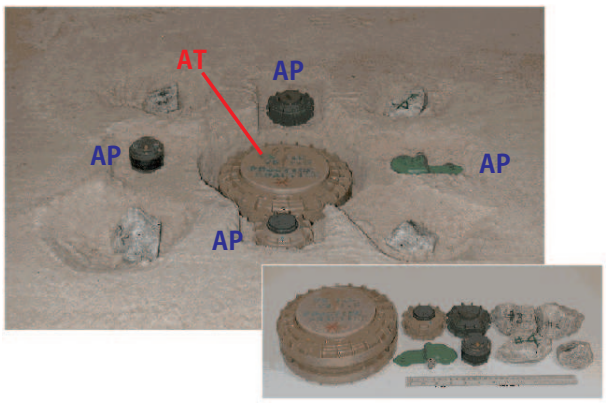


**Figure 1.5:** *Difference in size between APs and ATs*

only) part to be detected with a metal detector, and is attached to the pressure plate by a copper cover. When a tank rolls over the mine, it pushes down the pressure plate, which operates the particular spring retaining the fuse. The firing pin is then driven down to the detonator, which detonates and fires the M120 booster charge, which immediately sets off the main charge.

An example of AP can be seen in Fig. 1.4. The M16 Anti-personnel landmine can be either detonated in two ways: by applying pressure typical of a human body or by pulling the spring-loaded release pin. Either method causes the pin immediately to pull out of the fuse, releasing the striker and igniting the percussion cap.

As can be evinced by the figures, the overall metal amount is reduced to the minimum, but other well known tactics are usually aimed at keeping mines as dangerous as possible. For instance, it is common in war zones to find mines which were installed in piles, with the bigger one on the bottom: the purpose of this tactic is to avoid detection by using a very small landmine, which in turn triggers the bigger one.

The opposite can also be done, that is putting a smaller mine (AP) underneath a bigger one (AT); this configuration is suitable to kill demining personnel, which are expensive to be trained and exist in a small number with respect to other professionals, giving a dangerous tactical advantage when eliminated.

Eventually some low technology tricks can be used to fool demining, such as the insertion of wood sticks directly pointing to the trigger plate of the mine,

giving the ability to bury the mine deeper into the ground.

### 1.2.2 Humanitarian Demining

The mine clearance operations are classically divided in two major kinds: humanitarian demining and military demining. Military demining is usually oriented in cleaning a small area as fast as possible, disregarding of a complete coverage of the minefield, and is usually performed with ordnances or small exploding devices, which while not assuring complete reliability, are very fast to deploy and use.

Humanitarian demining, on the other hand, aims at totally recovering endangered areas from landmines, a process which can be very time-consuming, but which the purpose is to locate all mines so that the area can be returned to normal use. Even if only a small number of mines remain undetected, this can lead to an increase of civilian casualties.

It is important to notice that demining is usually a vital task to be performed before other humanitarian activities can take place. The procedures for landmine removal can vary greatly according to factors such as location, terrain, mine distribution, vegetation, etc. However, according to the UN standards, the humanitarian demining procedure can be divided in three levels [13]:

1. Level 1: Identification of minefields done using war documents and witnesses reports and improved with aerial surveillance and trained dogs.

2. Level 2: Coarse removal with big, heavy and well-armored vehicles that plow the ground detonating mines.

3. Level 3: Inspection to allow a safe return of the terrain to the local population. This is the most time-consuming step, which Gryphon is intended to be applied.

Well-trained dogs (Fig. 1.6) are able to sniff explosive chemicals like TNT in landmines, and are already used in several countries [48] [20]. Giant pouched rats (Fig. 1.7) [32] are also being trained to sniff out chemicals like TNT in land-

**Figure 1.6:** *Dog for landmine detection*

**Figure 1.7:** *Rat for landmine detection*

**Figure 1.8:** *Conventional MMD used in demining operations*

mines: these rats are currently working in minefields in Mozambique and are trained in Tanzania by humanitarian organizations.

Several modern methods are used and technologically refined day after day. Conventional metal detectors (Fig. 1.8) rely on electromagnetic signals with frequencies of the order of 10-100kHz, which are not sensitive to plastic or wooden mine bodies and the high explosive block itself. The only part of a low-metal mine that they may be able to detect is the detonator.

Much higher frequency signals (of the order of 1GHz) are employed in Ground Penetrating Radar (GPR), these signals are also sensitive to the non-metallic parts of the mine. Unfortunately, as they are also affected by innocuous objects

such as tree-roots and stones and by local changes in soil moisture, it is difficult to distinguish a mine on a GPR image. Both methods can be used by the Gryphon robot, subject of this thesis.

Finally, more exotic methods have also been developed, including a high variety of detection systems, such as honey bees, plants and even bacteria, while strictly technological means like nuclear and acoustic detection are also under refinement. The use of a highly experienced human deminer is still believed to be the optimal approach, and as will be clear in the next section, automatic devices usually try to imitate this behavior.

### 1.2.3 Mechanization and automation

Many attempts have been made to assist the demining operation by the use of mechanical systems. One example is the PEMEX robot (Fig. 1.9) [39], which is a two wheeled vehicle that carries a sensor for landmine detection and is light enough (16 kg) for not triggering hidden mines.

Some other approaches like walking robots were also made. TITAN IX (Fig. 1.10) is a quadruped walking robot that is able to adapt its gait to navigate on difficult terrain, and use one of its legs as a manipulator to scan the soil for mines, cut vegetation or even dig the ground [21].

In order to compensate the slowness, energy inefficiency, cost and complexity of walking robots, machines that roll on tracks or wheels were also proposed. One example is the semi-automatic detection machine named ODIS (Fig. 1.11), which is an imaging induction coil sensor developed at DASA-Dornier for the reconnaissance of buried metallic objects [5]. The ODIS module is fixed in front of another carrying vehicle and contains a matched set of rotating detectors which scan 1 m in width at about 0.5 m/s.

A promising method is the one developed by Yamanashi Hitachi, Japan (Fig. 1.12) [51]. This heavy machine contains a grapple with rotatory tool which explodes landmines. It is very fast but unfortunately it is still not able to exploded all landmines. Besides, it still can't cope with AT landmines, which explosion is so strong that can completely destroy the heavy machine. For this reason, collaborations with Gryphon have been discussed, which preliminary experi-

**Figure 1.9:** *Demining robot Pemex*

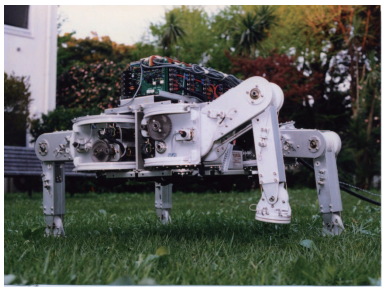

**Figure 1.10:** *Walking robot TITAN*

**Figure 1.11:** *ODIS platform*

ments showed great potential when combining the two systems.

## 1.3 The POD and FAR Problem in Demining Operations

When talking about efficiency of demining operations, two main parameters are used: Probability of Detection (POD) and False Alarm Rate (FAR). POD is related to the capacity of detecting landmines, calculated by eq. 1.1, which higher PODs indicate very efficient sensor/system. On the other hand, FAR represents the number of metal fragments wrongly detected as landmines in an area (eq. 1.2) and small values are ideal. Most of the existing demining solutions are able to achieve high PODs, but FAR tends to increase greatly. High FAR usually is related to "safe" operations, since all detected signals are treated as landmines and removal is done with long and exhausting standard procedures (Fig. 1.13). Moreover, for each hundred detected signals, only one is actually

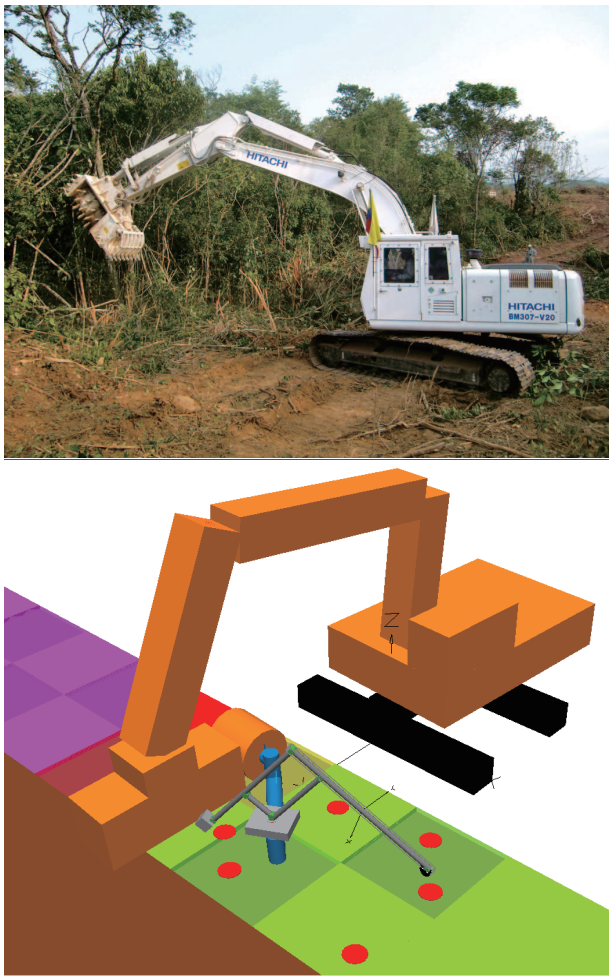


**Figure 1.12:** *Demining solution proposed by Yamanashi Hitachi (top) and possible combination with Gryphon for enhancing detection capabilities (bottom)*

a landmine, but since no efficient discrimination methods exist operations become too long and inefficient.

$$\text{POD} = \frac{\text{detected landmines}}{\text{total landmines}} \tag{1.1}$$

$$\text{FAR} = \frac{\text{detected non-landmines}}{\text{area}} \tag{1.2}$$


**Figure 1.13:** *Procedures taken for removing landmines. Operators lay down on the ground, dig objects slowly and with caution until they are completely removed*

### 1.3.1 Existing Approaches for Reducing FAR

The most commonly used landmine sensor is the Electromagnetic Indution based ones. At present, all commercially available MMDs deployed for use in the field cannot distinguish landmines from other metal fragments, and the idea that MMD alone cannot discriminate them is an accepted fact. However, as there are electromagnetic induction based detectors that can select metal types to be searched, such as gold detectors [34], there is an indication that MMDs can actually be used for discrimination of landmines and other metal fragments, as shown by many researches, such as: i) algorithms for evaluation of detected signals using model of physical phenomena [22] [44] [15], ii) feature extraction from MMD signals and classification of the data according to the metal type, size or depth of the metal fragments [30] [13] [38], and iii) algorithms that combine time domain analysis and frequency domain analysis [47] [7].

Unfortunately, none of these methods were successfully implemented for use in practical demining tasks yet. In [30] and [13] a database with targets information is used, but the discrimination is as limited as the number of data in the database. Some of the methods [10] [45] also rely on a dual-sensor ap-

proach, which is a combination of two sensors, a MMD with a GPR. The discrimination is based on the signal from the GPR instead of the MMD, but a high level of expertise is still necessary for proper evaluation of obtained data (image or sound), and discrimination is done considering large safety margin which keeps FAR still high. [38] uses image processing and MMD signal surface area and volume calculation estimating size and material, followed by depth estimation inclining the MMD in different angles. However, as work [13], it requires information from several depths (layers) for discrimination, considerably slowing down the demining operation.

In fact, MMDs signals are strongly dependent on targets physical properties, such as depth, material, size, shape and posture, as will be deeply explained in Chapter 5. Estimating one or all of these properties would lead to landmine/MF discrimination. Depth permits a quick discrimination of anti-personnel (buried in shallow depths with weaker explosions) and AT (deeper with very strong explosions) landmines. Moreover, depth information permits easier use of GPRs. GPRs can collect data of objects (mines or non-mines) in many depth ranges and operator's pre-knowledge of the buried location of targets influences significantly the detection for these kind of systems. The material which targets are composed is also a valuable information for discrimination, since landmines are in great part made of steel.

### 1.3.2 Gryphon POD and FAR

According to experiments in testfields [41], the demining robot Gryphon (detailed in Section 2) proved to be better than human operators in terms of correctly detecting landmines with high POD and low FAR (Fig. 1.14). However, too much time and workload are required for landmine detection and marking and further improvements are still desired for reducing FAR and new methods for solving these issues are proposed in this Thesis, as will be detailed in the following Chapters.

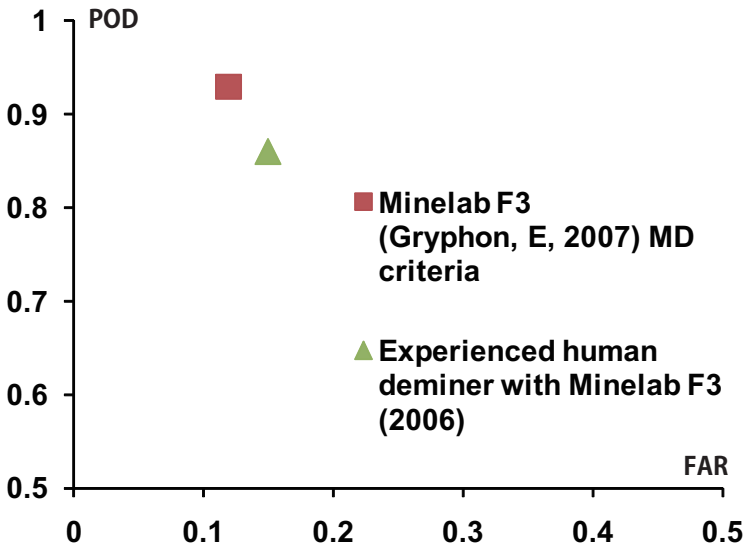


**Figure 1.14:** *Gryphon's POD and FAR obtained in experiments in testfields*

# Chapter 2

# Description of the Robotic Demining System Gryphon

Developing a device for demining is complex due to the diverse possible working environments, climates, soils and vegetation. Therefore, the device should be robust, adaptive, flexible, affordable, simple and moreover safe.

Based on these considerations, a mobile robot named Gryphon (Fig. 2.1) was developed to assist the mine detection operation. Several versions of the robot were developed in the past years, leading to a progressive refinement of both the hardware and software aspects of the operations.

The main parts of the latest version of Gryphon will be described in the following sections: mobile platform, manipulator, stereo vision camera, landmine sensor, marking system, Global Positioning System (GPS) and control architecture.

## 2.1 Mobile Platform

The mobile platform is a commercial 4-wheeled All Terrain Vehicle (ATV), model no. Grizzly 400 auto, 4x4, Yamaha Corp (Fig. 2.2). This vehicle (also called as "Buggy" in this thesis) can transport both the robot and operator along the minefield, and can be totally remotely controlled from a safe distance.

Another major advantage of this solution, is that no electric power is re-

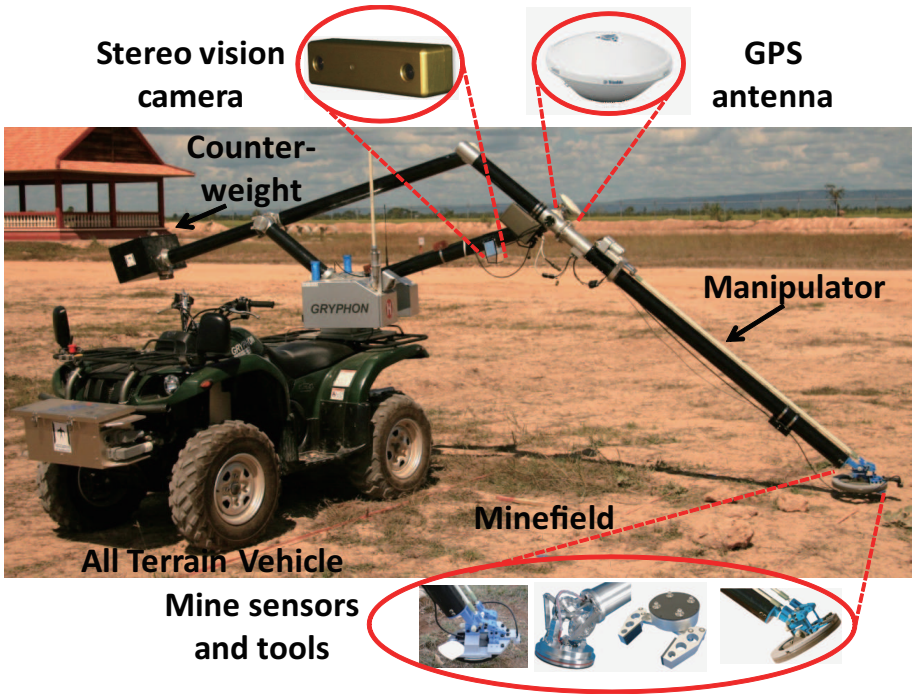


**Figure 2.1:** *Overview of Gryphon's main elements*

**Figure 2.2:** *ATV for Gryphon's locomotion*

quired to operate it. The electric power needed to operate the robotic arm, the control systems and the onboard PC comes directly from the combustion engine of the ATV, which is a vital requirement in rural areas typical of minefield locations.

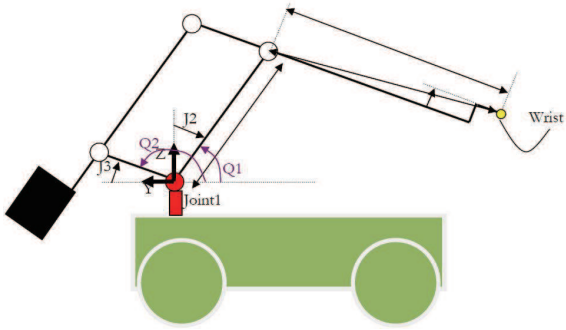


**Figure 2.3:** *Arm model (side view)*

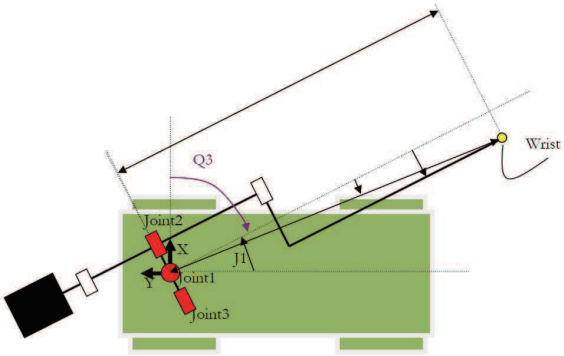


**Figure 2.4:** *Arm model (top view)*

## 2.2 Manipulator

The robot is equipped with a 3m-long arm, that can scan large areas at once from safe distances. Due to its length and weight, a counterweight on the opposite side with respect to the wrist has been installed, to both balance the arm in every configuration and allow the system to be more energy efficient and precise. Inside the counterweight, batteries are included to provide energy when the combustion engine is not ignited.

The schema of the arm is shown in Figs 2.3 and 2.4. The arm can be equipped with two kinds of wrist mechanisms, the first one, intended for small payloads, is able to carry a weight which is less than 10kg, built with nonmetallic and non-conductive material, operated by a 2 DoF parallel mechanism. The second wrist is intended for heavier payloads, and was built to allow a quick interchange between the two. It also features an additional degree of freedom and was built to also support fast and autonomous tools interchange. A schema of the first wrist mechanism, which was mostly used during the tests regarding the present project, is shown in Fig. 2.5

## 2.3 Stereo Vision Camera

In order to scan the terrain with efficiency and reliability, a device for capturing the terrain information is essential. A BumbleBee stereo vision camera [42] (Point Grey Research, Canada, Fig 2.6) is attached to the manipulator's

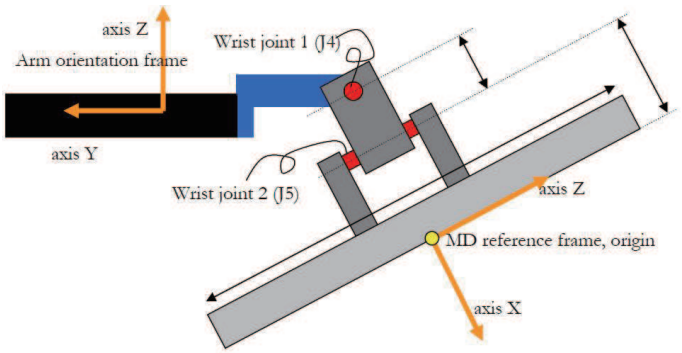


**Figure 2.5:** *Wrist model (with MMD)*

**Figure 2.6:** *Stereo vision camera attached to Gryphon*

first link, at a position that allows easy topographic data acquisition.

Stereo vision's operation principle is based on matching corresponding points from the left and right images. A simple triangularization permits determining (x,y,z) positions of a point relative to the cameras.

Bumblee is compatible with several different calculation libraries, allowing easy manipulation and correction of individual images. The final results are depth maps or collections of points in 3D space.

## 2.4 Metal Detector

Gryphon was originally designed to cope with a MMD and an optional GPR (which has not been used in the latest versions and will not be discussed in this thesis).

MMDs use electromagnetic induction to detect metals. The simplest form of a metal detector consists of an oscillator producing an alternating current that passes through a coil producing an alternating magnetic field. If a piece of electrically conductive metal is close to the coil, eddy currents (also known as

**Figure 2.7:** *Minelab F3, recently most used MMD in Gryphon*

Focault currents) will be induced in the metal, and this produces an alternating magnetic field of its own. If another coil is used to measure the magnetic field (acting as a magnetometer), the change in the magnetic field due to the metallic object can be detected.

The small payload wrist mechanism of the manipulator can accommodate two different types of metal detectors, a MIL-D1 (CEIA, Arezzo, Italy) and a F3 (Minelab, Adelaide, Australia) [35]. Providing the option to rapidly switch back and forth between the two metal detectors can represent a real advantage in acquired data quality, since, depending on the soil type, one detector might perform better than the other. Despite this, some demining organizations prefer to use one sensor with respect to others, so the capability of mounting a diverse set of metal detectors can help the work on the field, by using the most suitable device.

The latest version of Gryphon uses mainly the F3 MMD (Fig. 2.7). The sensor's head has a diameter of 22 cm and is made up by a single circular coil. Two signed binary values, representing two different measurement variables, are read through serial communication [13]. Details of these two binary values will be deeply discussed in the following chapters.

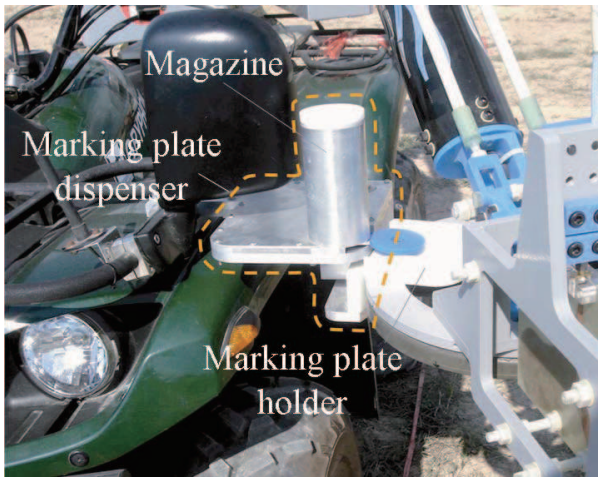


**Figure 2.8:** *Marking based on plate dispenser*

## 2.5 Marking System

Once a potential target is detected, it is necessary to mark the corresponding spot on the ground, for clearly showing the presence of a threat, and also for later removal procedures.

Two different marking systems have been developed: one based on a plate dispenser, and another based on painting. For the first one, a marker dispenser is installed on the vehicle (Buggy), and can contain colored markers (either blue or red at the moment). The robot is capable of reaching autonomously the marker dispenser and load a marker directly on a special plastic container installed on top of the metal detector. The marker itself is kept at its position by gravity, and after the MMD is placed on the correct position, a little leaning of the MMD is enough to drop the marker exactly at the desired location on the ground. The dispenser is not actuated, and the dropping must be performed with one marker at a time. The whole marker dropping sequence from the dispenser is actuated by pressing a lever which can be seen, among the other components of the system, in Fig. 2.8.

The secondary marking system with paint is based on a nozzle attached to the detector which is able to spray colored paint on the ground. There is a paint reservoir located near the manipulator base and a flexible pipe conducts the liquid to the nozzle, liquid which is moved by a pump located close to the paint

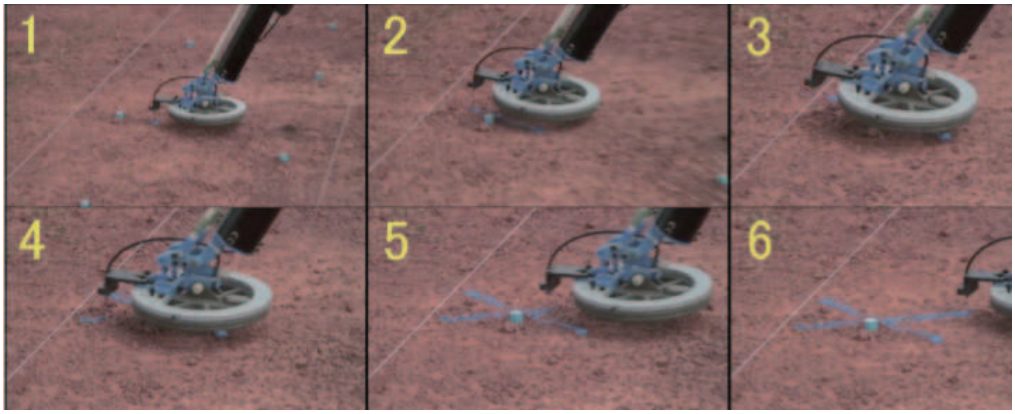


**Figure 2.9:** *Marking based on paint*

tank.

This last system proved to be extremely flexible, because it can effectively write useful data on the ground, like the perceived depth of the suspect object, or numbering and classification of the objects with respect to some of their basic characteristics.

## 2.6 GPS

In order to localize itself along the minefield, and to store the scanned information (where the targets were located, which areas were already scanned or not), Gryphon needs a robust positioning system. The chosen solution was a satellite navigation system based on the currently most accurate form of GPS: the so called Real Time Kinematics (RTK).

RTK systems use a single base station receiver and a number of mobile units (rovers). The base station re-broadcasts the phase of the carrier that it measured, and the mobile units compare their own phase measurements with the ones received from the base station. This allows the units to calculate their relative position to millimeters, although their absolute position is accurate only to the same accuracy as the position of the base station. In the Gryphon robot this allows to clearly distinguish the coordinates of the joint in which the unit is attached, thus solving the kinematic problem will lead to a precise measurement of point coordinates on the ground.

For the RTK, Gryphon uses as antennas the Trimble Zephyr Geodetic Model 2 as base (Fig. 2.10), and the Zephyr Model 2 as rover (Fig. 2.11) (Trimble Navigation Limited, CA, USA). The receivers used with each antenna are the Trimble SPS751 (Fig. 2.12). For data exchange between rover and base, a wireless modem model Yaesu YRM-211T (Fig. 2.13) was chosen.

Each additional Gryphon vehicle requires one additional receiver. Precision is typically 1 cm horizontally and 3 cm vertically. The GPS antenna is attached at the top of the first link of the manipulator. The GPS was not deeply used in this thesis, but is undoubtedly an important tool for demining operations with Gryphon, being explored in other related work [2].

## 2.7 Control Architecture

The remote control unit Fig. 2.14, is a rugged and weather-proof case which hosts every control mechanism for Gryphon.

Two joysticks are the core control mechanism for the manual driving of the manipulator and the vehicle itself, since the pressure of a button can commute the driving mode from the arm to the buggy direction and acceleration control. Its upper part contains a Tablet PC with an attached tablet pen, which allows keyboardless interaction with the main control software, named Visu. The operator is thus able to launch every automatic or semi-automatic procedure remotely, visualize the data, mark suspected locations and monitor the system state.

The communication is twofold because two independent wireless links are actually used: one for the control box related to the actuators, while the second one for autonomous interaction with the high level control software.

The whole architecture of the control system in Gryphon is shown in Fig. 2.15. The two applications running the control system are called BuggyPC and Visu, and are respectively installed inside the Buggy and the Remote Control Unit. The computer onboard the buggy runs an ad-hoc version of Windows XP Embedded. This choice minimizes unwanted memory waste and increases overall speed by utilizing almost all the resources for direct computation.

Windows XP Embedded, commonly abbreviated "XPe", is a componentized

**Figure 2.10:** *Base antenna (immobile)*

**Figure 2.11:** *Rover antenna (mobile)*

**Figure 2.12:** *Trimble receiver used with Gryphon*

**Figure 2.13:** *Yaesu modem for RTK*

**Figure 2.14:** *Remote control unit*

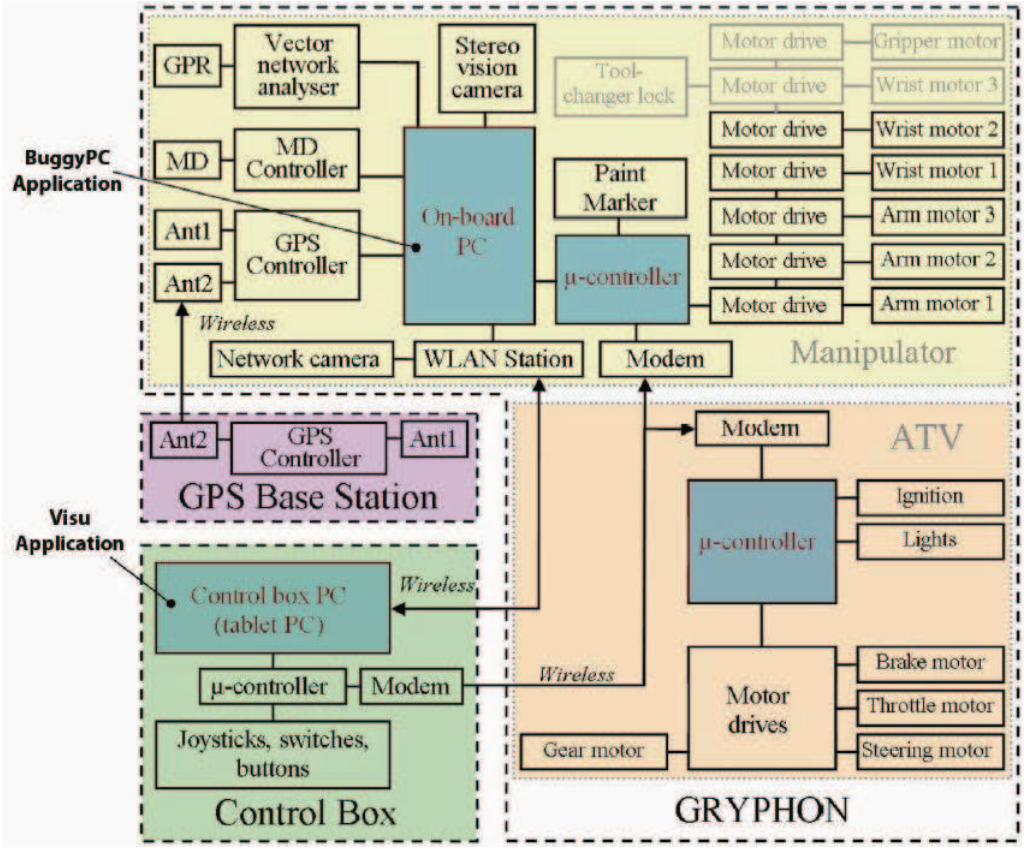


**Figure 2.15:** *Overview of the control architecture of Gryphon*

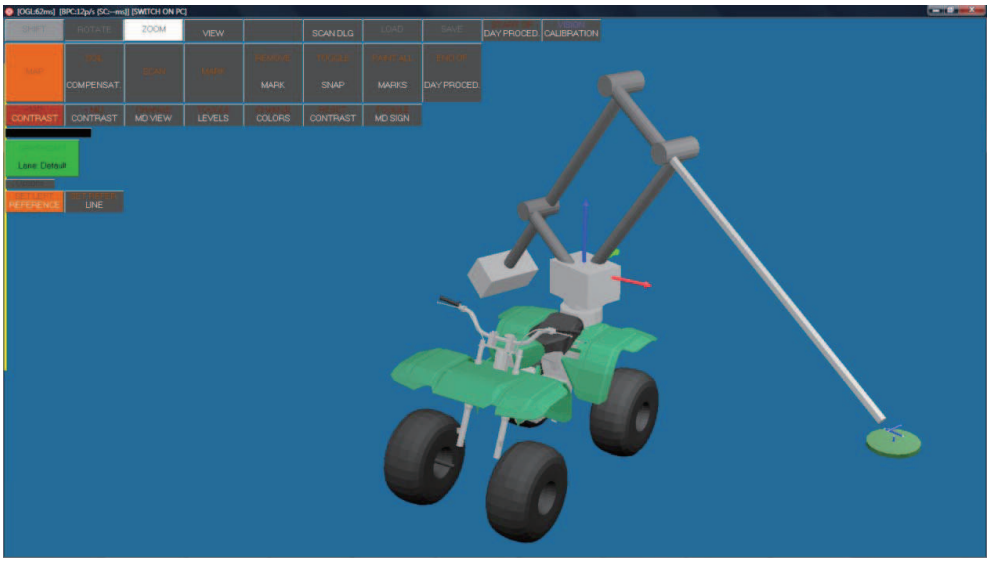


**Figure 2.16:** *Overview of the Gryphon user interface "Visu"*

version of the Professional edition of Windows XP. An original equipment manufacturer is free to choose only the components needed, thereby reducing operating system footprint and also reducing attack area as compared with XP Professional.

On the Remote Control Unit side, the Visu application runs within a Windows XP Tablet Edition environment. This edition is intended for specially designed notebook/laptop computers called tablet PCs. Windows XP Tablet PC Edition is compatible with a pen-sensitive screen, supporting handwritten notes and portrait-oriented screens.

This Visu application is where all improvements in this thesis were coded. The code was originally created by Doctor Marc Freese [13] and is composed of hundreds of classes and more than 52.000 lines of code [37]. The Graphical User Interface (GUI) is based on the OpenGL libraries, a standard specification defining a cross-language, cross-platform API for writing applications that produce 2D and 3D computer graphics. A screenshot of the GUI is shown in Fig. 2.16 where the 3D model of the robot is clearly visible: the model is automatically updated to reflect the instantaneous configuration of the real robot.

## 2.8 Analysis and Limitations of the System

According to [50], stress is recognized as a significant factor that affects performance in many areas such as aviation, military, sports and surgery. Time is a considerable stress causer. Problems involving waiting situations invariably assume that people dislike having to wait. [40] states that it is generally accepted that having to wait for a certain amount of time builds up anxiety and stress in a individual, due both to the sense of waste and the uncertainty involved in a waiting situation.

In the Gryphon system, it is not different. It still relies on a human operator who is prone to waiting times, workload and stress. For an efficient demining operation, these drawbacks must be reduced. Considering specially these points, this chapter will introduce a brief analysis of the Gryphon system and its limitations.

### 2.8.1 Demining Tasks with Gryphon

After introducing the main components and background of the demining robot Gryphon, its tasks can be easily understood. The tasks for a demining operation using Gryphon can be mainly divided in seven:

1. Start of day procedure: performs auto-checking of system components and move the manipulator from the resting position to the standby position, and brings the system to be ready to start the demining operation.

2. Mapping: the depth map, i.e, the 3D profile, of the area to be demined is mapped by use of a stereo vision camera.

3. Reference points/line setting: the area to be scanned is selected by a group of points/line.

4. Scanning: the manipulator is moved inside the scan area and the MMD signal is captured.

5. Marking: suspect mines locations are painted/marked.

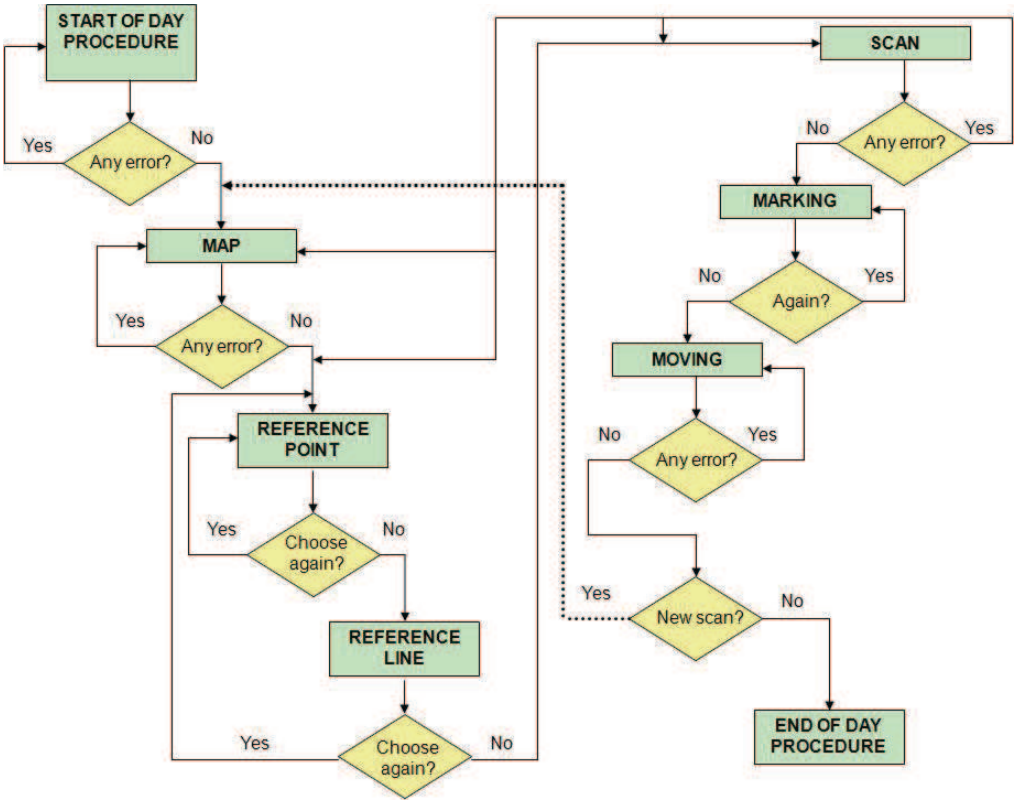


**Figure 2.17:** *Flowchart of the demining operation with Gryphon*

| MAP | REFERENCE LINE | SCAN | MARKING | MOVING |
|---|---|---|---|---|
| CPU | Human | CPU | Human | Human |
| Data Transfer | | Data Transfer | Data Transfer | GPS |
| Camera | | Arm Motion | CPU | |
| Arm Motion | | | Arm Motion | |

**Figure 2.18:** *Gryphon's main tasks and its sub categories*

6. Moving to the next area: the robot is moved to the next area to be scanned.

7. End of day procedure: the manipulator is brought to the resting position, and the robot is ready to return from the minefield.

Fig. 2.17 shows a simplified flowchart summarizing the above tasks and possible scenarios during the demining operation.

After the main seven steps are clearly identified, they can be divided into

TABLE 2.1: MAPPING TASK DETAILS

| Steps | Description | Category |
|---|---|---|
| a | Start of motion | CPU/Data Transfer |
| b | Motion from P1 to P2 | Arm Motion |
| c | Map displayed in the interface | Camera |
| d | Motion from P2 to P1 | Arm Motion |
| e | Map displayed in the interface | Camera |
| f | Motion from P1 to P3 | Arm Motion |
| g | Map displayed in the interface | Camera |
| h | Motion from P3 to P1 | Arm Motion |
| i | Map processing | CPU/Data Transfer |

categories, according to the main components responsible for the task as shown in Fig. 2.18. The tasks that contain "CPU" and "Data transfer" are tasks that require computer processing, and data transfer between the BuggyPC and the control box. "Arm motion" refers to tasks which contains arm movement, like the scanning and marking. "Human" means that the task strongly depends on the operator decision making. Finally, "GPS" and "Camera" are specific hardwares attached to the robot to perform very particular tasks.

Each of the seven tasks, by they turn, can be subdivided into steps, from the moment the operator inputs the command for starting until the moment the robot finishes the task and displays the results in the user interface.

### 2.8.2 Mapping Task

The mapping task can be mainly subdivided into nine steps ("a", "b", "c", "d", "e", "f", "g", "h" and "i"), as shown in Fig. 2.19 and Table 2.1.

After the operator presses the command button in the interface for starting the mapping task, after a certain time interval the robot starts the motion (step "a"). The robot grabs the environment information by capturing three consecutive images in three different positions (P1, P2 and P3 in steps "b", "d" and "f"), processing each image in each position (steps "c", "e" and "g"). Finally, the robot goes back to the initial position (step "h") and compute the final map, which is the combination of all three images (step "i").

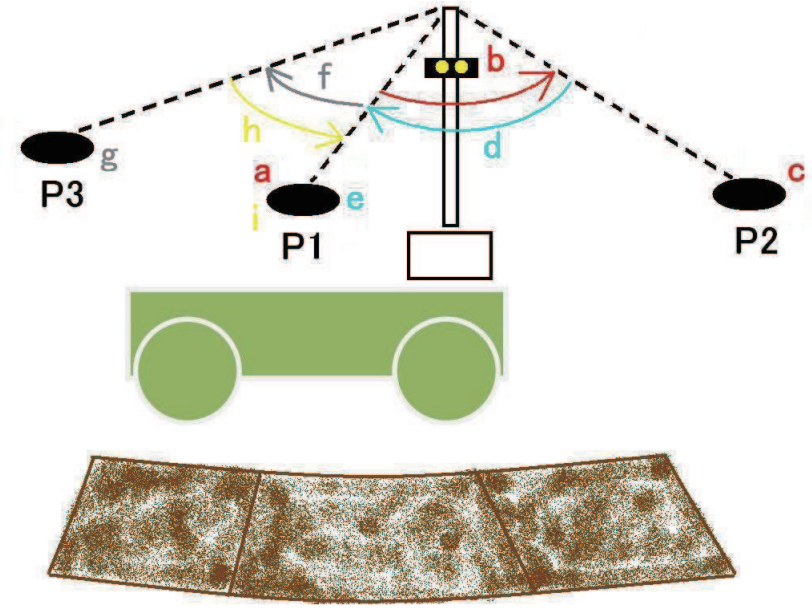


**Figure 2.19:** *Mapping task*

### 2.8.3 Scanning Task

The scanning task can be subdivided into six steps (“a”, “b”, “c”, “d”, “e” and “f”), displayed in Fig. 2.20 and Table 2.2.

In a similar way as in the mapping task, step “a” represents the interval for starting the motion, from the moment the command was input in the interface. The arm moves to a starting position (“b”), which is the closest point in relation to the Buggy that is going to be scanned. The robot then computes the trajectory (step “c”) according to the terrain information grabbed by the stereo vision camera. The robot moves the arm along the terrain (step “d”), scanning and storing the MMD information, which will be processed in step “e”. Finally, the arm goes back to the initial position (step “f”).

### 2.8.4 Marking Task

The marking task can be subdivided into four steps (“a”, “b”, “c” and “d”), illustrated in Fig. 2.21 and Table 2.3.

Step “a” represents the interval for starting the motion, from the moment

TABLE 2.2: SCANNING TASK DETAILS

| Steps | Description | Category |
|---|---|---|
| a | Start of motion | CPU/Data Transfer |
| b | Motion from P1 to P2 | Arm Motion |
| c | Trajectory generation | CPU/Data Transfer |
| d | Scanning | Arm Motion |
| e | MMD information processing | CPU/Data Transfer |
| f | Motion from P3 to P1 | Arm Motion |

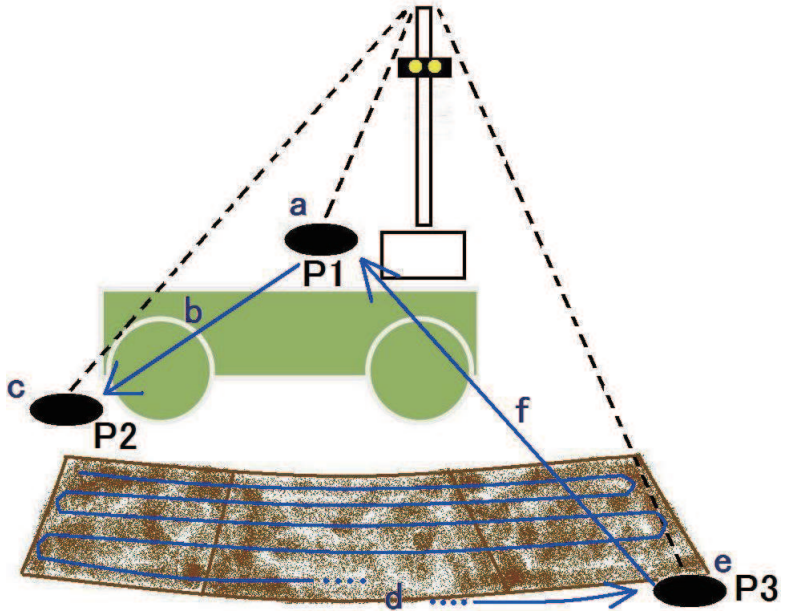


**Figure 2.20:** *Scanning task*

the command was input in the interface. The arm moves to spot on the terrain where the mark should be placed ("b"). The robot paints an "X" on the ground (step "c") and returns to the initial position (step "d").

### 2.8.5 System Analysis

The demining operation (except for "start" and "end of day procedure" tasks, which are done only once during a demining operation) were executed several times, and an average time for each step in each task was obtained. Moreover,

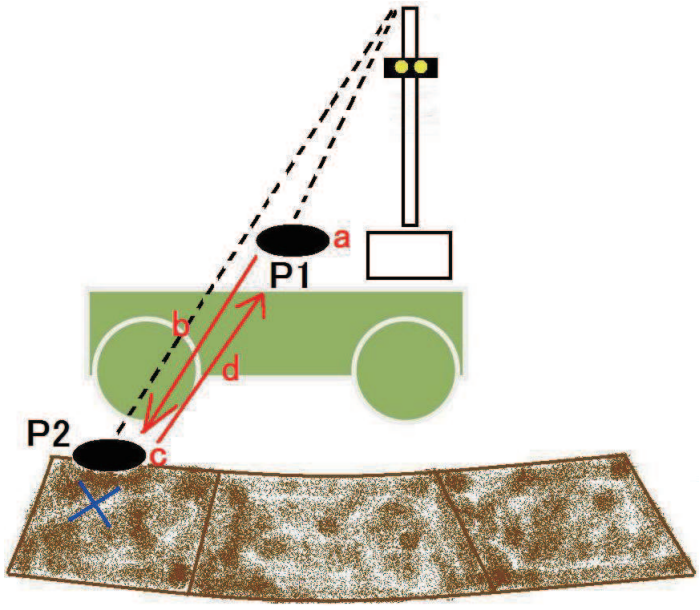


**Figure 2.21:** *Marking task*

TABLE 2.3: MARKING TASK DETAILS

| Steps | Description | Category |
|---|---|---|
| a | Start of motion | CPU/Data Transfer |
| b | Motion from P1 to P2 | Arm Motion |
| c | Mark painting | Arm Motion |
| d | Motion from P2 to P1 | Arm Motion |

the operation was repeated for three different methods, which different CPUs and RAM memories were used. The results are displayed in Fig. 2.22.

The category "Human" was not evaluated here due to the variety of performances from different operators. However, according to experiments in the real minefield [41], it could be verified that "human" is a fundamental factor to be considered for the total system efficiency, and it will be explored in details in the next chapters of this thesis.

Analyzing the results in Fig. 2.22, it is possible to observe that the categories which represent longest time duration refer to "Arm motion" and "CPU/Data Transfer", representing more than 90% of the total time duration of each task.

Gryphon deals with huge amount of data, so it is natural that CPU power is

required and data is constantly transferred between Visu and BuggyPC applications. According to the different methods applied in the analysis, as expected, hardware (mainly CPU) plays an important role in the system, and improvements can lead to great benefits (difference from a total of 388.53 s to 273.19 s).

In other words, besides the already mentioned drawback of human factors, arm motion should be also tackled in the system.

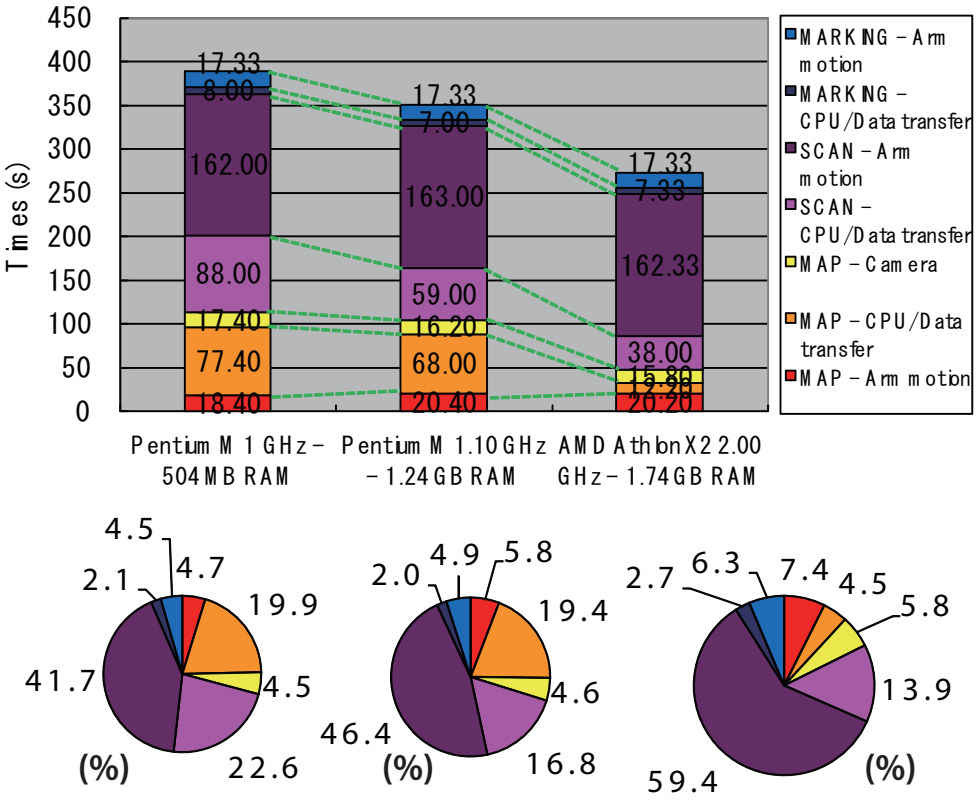


**Figure 2.22:** *Gryphon time duration analysis for three different methods*

# Chapter 3

# Targets Detection Method Enhancements

This chapter introduces some filtering techniques that permits clearer targets visualization, which are the basis for the remaining techniques presented in this thesis.

First, the so far used landmine detection procedures and its difficulties are described and then techniques for overcoming these problems are proposed. Most of the information in this and next Chapters are deeply detailed in [25].

## 3.1 Landmine Detection Difficulties

EMI based mine detectors sensors are prone to noises, generated by the soil or metal fragments nearby. Fig. 3.1 shows an example of one scan line (Fig. 3.2) of a Minelab F3 MMD [35] signal obtained in the test field. Even though there are no landmines around, it can be observed that the signal is oscillating around a positive value (offset) of 0.1 %. For compensating the offset, before starting the demining operation deminers perform the so called "ground compensation" [17], which the method varies from different makers.

However, from Fig. 3.1, it is possible to see that the noise suffers variable fluctuations, caused by changes in the soil patterns and also by subtle variations in the scan height. The "ground compensation" cannot balance this vari-

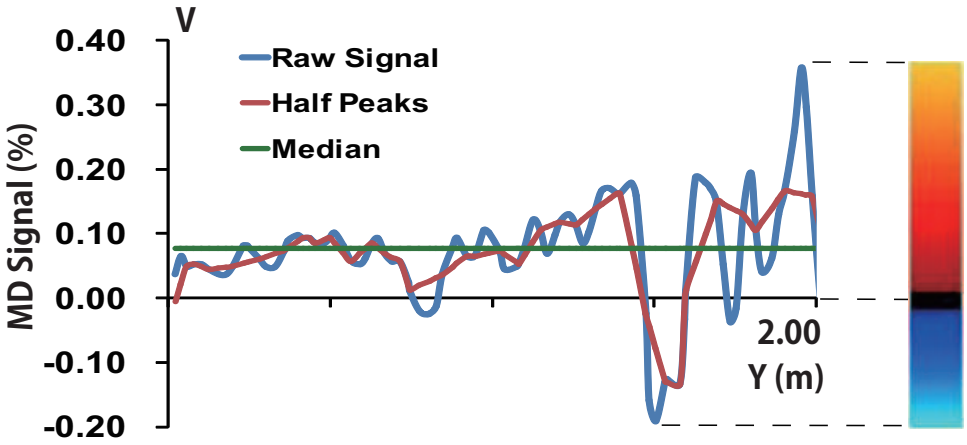


**Figure 3.1:** *Example of signal and different filters*

able offset, as shows the 2D signals image in Fig. 3.3. The MMD has two channels, "A" and "B", which the information is combined to form the final output "$V_{out}$", which in turn is stored in an array. Even though the normal method adopted with the MMD is as equation (3.1) [22], the basic method applied in Gryphon is as equation (3.2), where the output is function of $n$. It is a parametric variable which is related to an (x,y) position and is sorted according to the data acquisition sequence, the scanning path, as Fig. 3.2.

The 2D image is then composed by attributing colors (shown in the color bar in Fig. 3.1) to the MMD signals in the array, which 0 values are represented in black, the maximum value in yellow and the minimum in light blue.

$$V_{out} = max(|V_A|, |V_B|) \tag{3.1}$$

$$V_{out_a}(n) = \begin{cases} V_A(n), & if max(|V_A(n)|, |V_B(n)|) = |V_A(n)| \\ V_B(n), & if max(|V_A(n)|, |V_B(n)|) = |V_B(n)| \end{cases} \tag{3.2}$$

## 3.2 Sensor Pre-processing and Filtering

In the best result with Gryphon in [41], changes in the data processing led to better visualization of the targets compared to the normal method so far

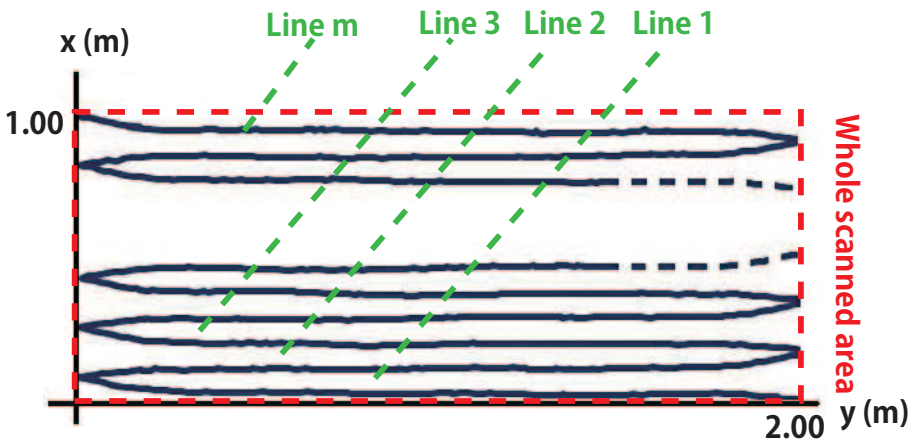


**Figure 3.2:** *Lines of a scanned area*

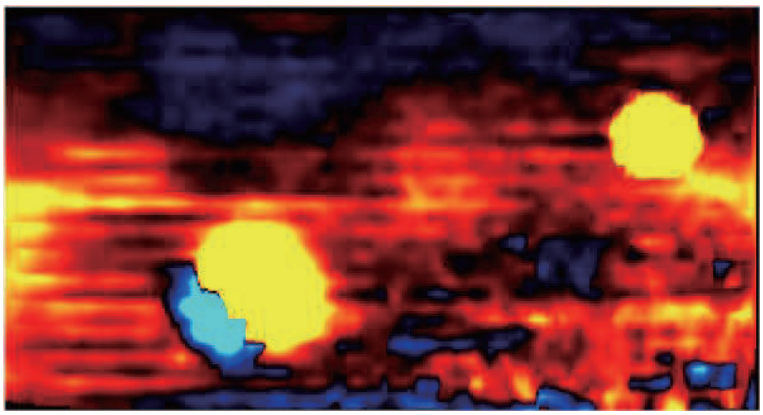

**Figure 3.3:** *Resulting images of the filtering methods: a) Basic*

adopted. The signal offset was compensated using the median value of the signals in each scanned area. However, the median value is not the most suitable since the noise along the area is variable as already shown in Fig. 3.1.

### 3.2.1 Proposed Filters

For compensating the variable offset, for instance, a filter through the signal in an average value (middle of two consecutive peaks), following the noise patterns could be applied. In this research, many other approaches for filtering the signals were implemented and analyzed. Figs. 3.3 to 3.7 shows the corresponding 2D representation in the user interface of a data filtered with different

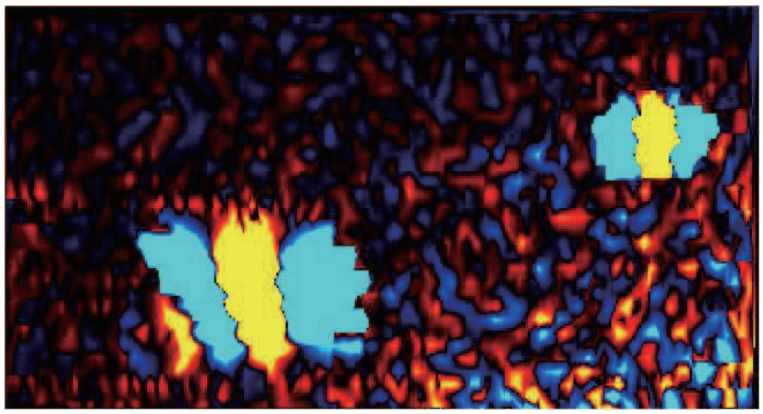

**Figure 3.4:** *b) Half-peaks offset*

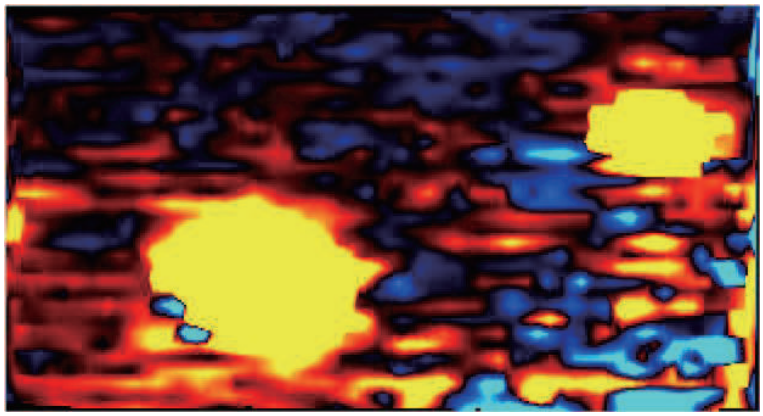

**Figure 3.5:** *c) Half-peaks*

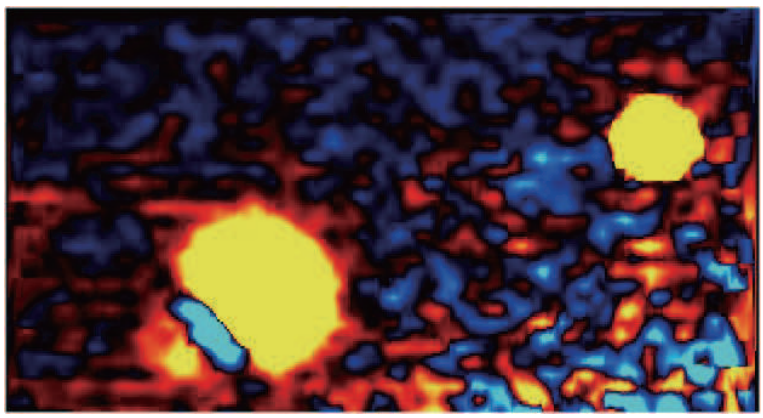

**Figure 3.6:** *d) Median of an entire area*

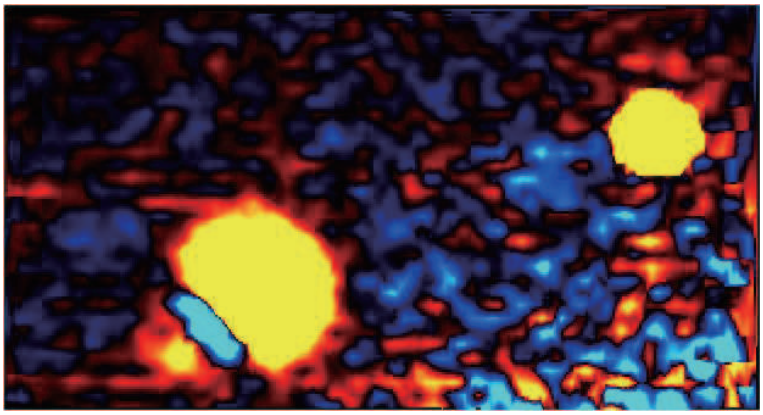

**Figure 3.7:** *e) Median of each line in an area*

methods: a) Basic Method, b) Half-peaks offset, c) Half-peaks as result d) Median of the whole area and e) Median of each line of the area.

Method “a” is the one used in the first evaluation during the test field in [41], described in section 1. Method “b” uses the offset ($V_{HP}$) calculated in the middle of two peaks (already shown in Fig. 3.1 and detailed in the flowchart in Fig. 3.8) and subtracts the values of the raw signal ($V_{RS}$):

$$V_{out_b}(n) = V_{RS}(n) - V_{HP}(n) \tag{3.3}$$

Method “c” displays the obtained offset line itself:

$$V_{out_c}(n) = V_{HP}(n) \tag{3.4}$$

Method “d” is the one applied during the second evaluation in [41], where the median of the whole scanned area (Fig. 3.2) is used as offset ($V_{MA}$), as shown in equation (3.5). The algorithm for computing the median is based on the Bubble sort [1] followed by a simple median calculation.

$$V_{out_d}(n) = V_{RS}(n) - V_{MA}(n) \tag{3.5}$$

Method “e” is a similar approach, but calculating the median for each line of the whole area ($V_{ML}$):

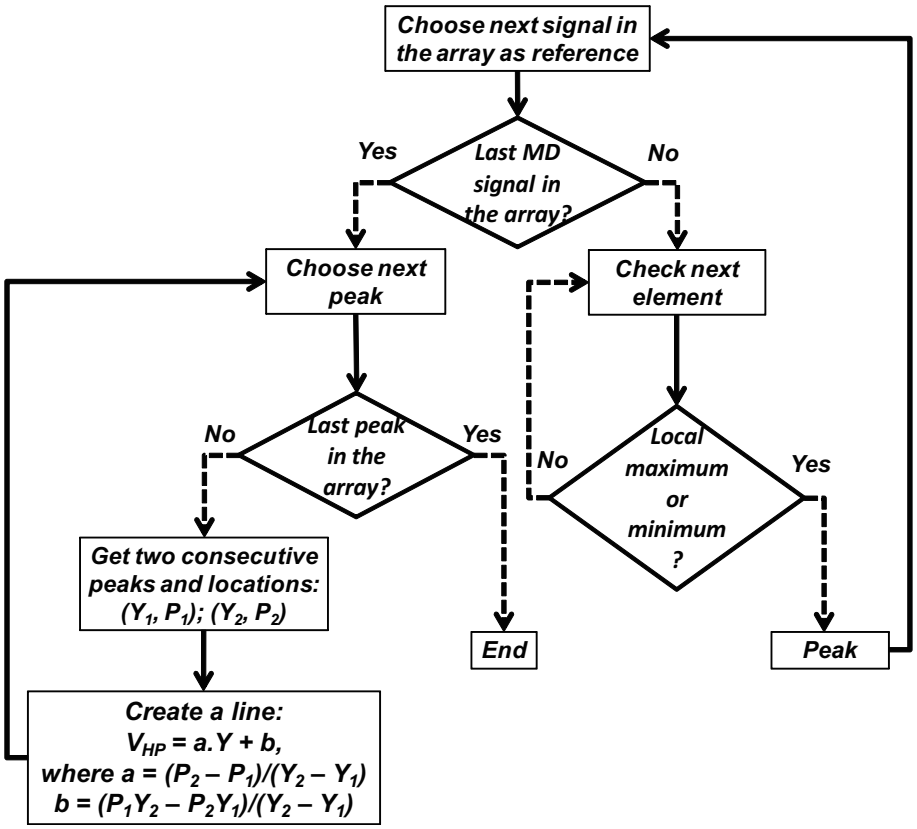


**Figure 3.8:** *Half-peaks algorithm*

$$V_{out_e}(n) = V_{RS}(n) - V_{ML}(n) \tag{3.6}$$

Table 3.1 shows a pre-evaluation of the methods. According to the database information (shown in Fig. 3.9), the area contains three targets ("T1", "T2" and "T3"). For a good identification, the first adopted requirement is that the targets have a minimum size so they can be clearly distinguished from noises. In the table, "T1", "T2" and "T3" indicate if the resulting sizes are above a determined threshold. Another adopted criteria is the number of zero crosses (variations from positive to negative signals), which a high number indicates that the filter is reaching a good offset level. After computing the sizes and zero crosses, it can be verified that all filters satisfy the size threshold for targets "T1" and "T3", but "T2" was not verified in filters "a" and "b". The half-peaks offset showed the best offset level with the highest number of zero crosses (as expected), but

TABLE 3.1: FILTERS COMPARATIVE TABLE

| Method | T1 | T2 | T3 | ZC |
|---|---|---|---|---|
| Basic (a) | O | X | O | 83.5 |
| Half-peaks offset (b) | O | X | O | 533.5 |
| Half-peaks (c) | O | O | O | 147 |
| Area median (d) | O | O | O | 222 |
| Each line median (e) | O | O | O | 222.5 |

T1, T2, T3: SIZE THRESHOLD CRITERIA OF TARGETS 1, 2 AND 3
ZC: NUMBER OF ZERO CROSSES (PER $m^2$)
SIZE CRITERIA: O = SATISFIED; X = NOT SATISFIED

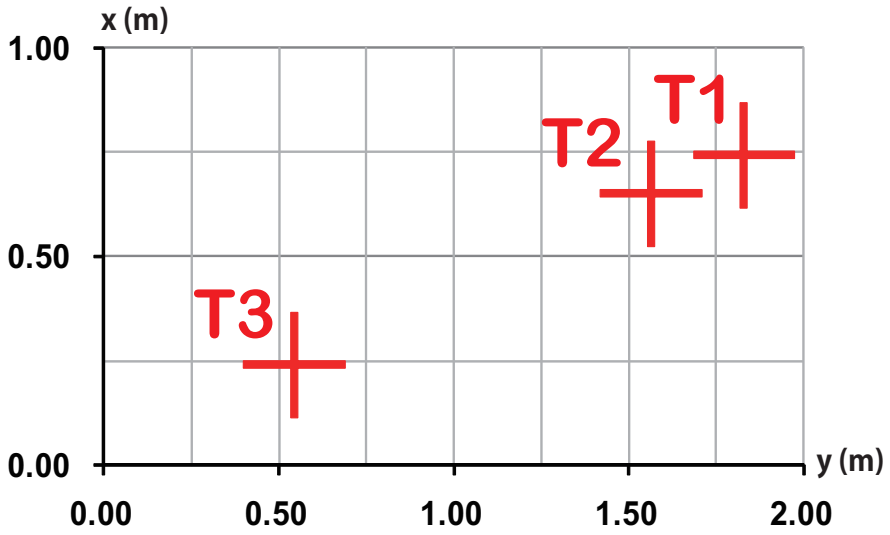


**Figure 3.9:** *Database information*

finally it distorted "T2" making the visualization unclear. Finally, among the filters that satisfy the threshold level for all targets, the one with highest number of zero crosses is “e”, detailed in the flowchart in Fig. 3.10. Even though targets can be equally seen as in “d”, it is possible to verify more meshes (red and blue alternations) in the upper regions, which helps avoiding false positives.

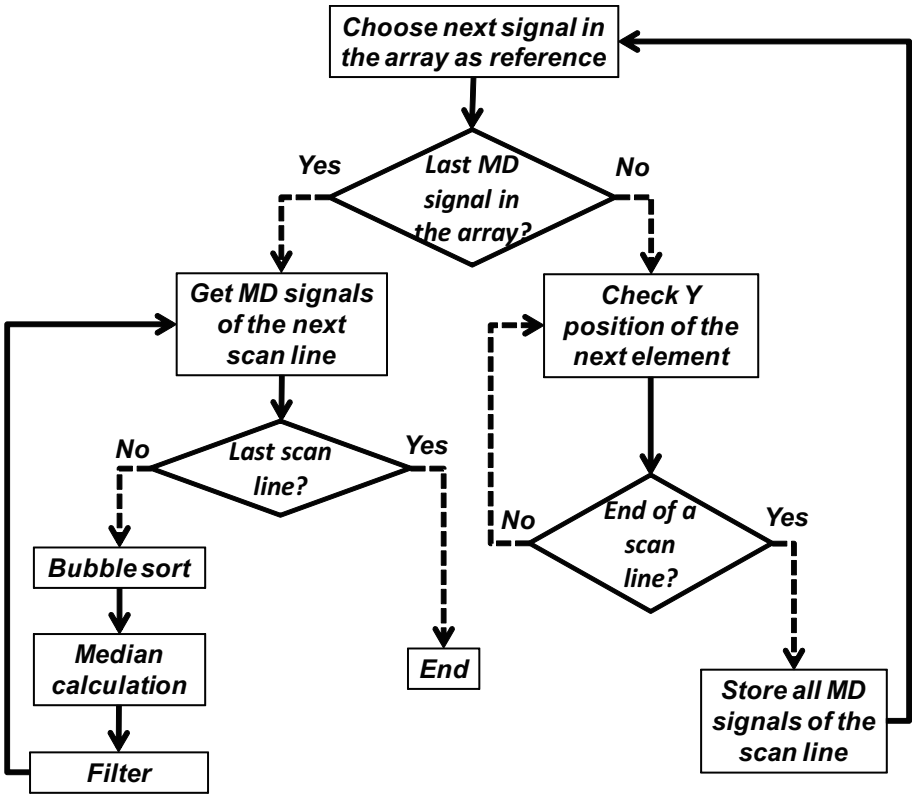


**Figure 3.10:** *Algorithm of the best pre-evaluated filter "e"*

### 3.2.2 Filter Evaluation

The proposed filter "e" was evaluated and compared with the basic method "a". Part of the test field data was chosen and 6 subjects non-familiar with demining marked the targets in laboratory. Before the experiment, the subjects received instructions on how to operate the user interface and how to look for potential targets. All subjects used the same computer (Windows 7, Intel Core i7, 4Gb RAM), equipped with a Gryphon user interface that was slightly modified for the experiment. Unnecessary command buttons were hidden and all experimental data was pre-loaded in the interface; the subjects had only to shift between one data and another by pressing implemented buttons. Each time these buttons were pressed, all marked targets were automatically stored in a data file. 3 subjects used method "a" and other 3 method "e". The experiment was repeated for each group in two different days. The time for completing the

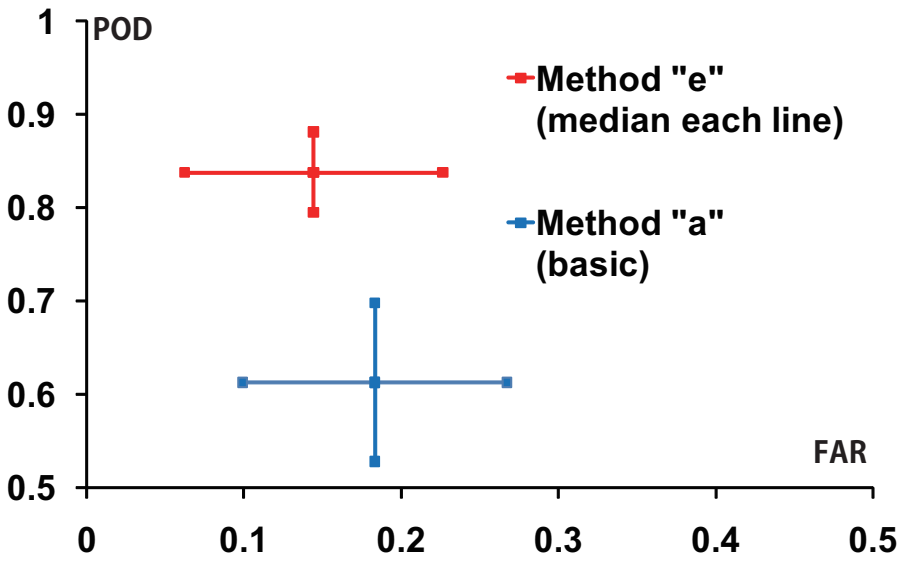


**Figure 3.11:** *Experiment results with 95% confidence limits*

whole data of each subject was recorded and an average time for completing each area ($m^2$) (excluding data loading time) was calculated. The results stored in data files were compared to the database, and POD and FAR of each subject were calculated.

Fig. 3.11 shows the average results for each method [14], which the increase in performance using method "e" can be clearly noticed. The only disadvantage occurring in method "e" is the time for decision making. Since it permits more information visualization, it requires longer decision time (19.95 s/m$^2$) from the subjects comparing to method "a" (9.6 s/m$^2$).

# Chapter 4

# Targets Marking Method Enhancements

The previous section introduced a method that enhances POD and FAR in case a human operator is performing the landmine detection task. Even though the performance was improved, the time duration continued being a drawback, and human factors still persist. This section introduces an automatic method for landmine detection and marking.

## 4.1 Targets Perimeter Searching Algorithm

The implemented algorithm is mainly based on the MMD signal amplitude and its represented size in the 2D graph. These two parameters were adopted by analyzing all data, and by previous knowledge of potential targets (Fig. 4.1). The algorithm is shown in the flowchart in Fig. 4.2.

The algorithm was tested with all data from the test field and with different filters. It was verified that using filter “c” followed by filter “d” the algorithm performed better, since this filter removes great part of the oscillations and smooths the data, compensating the offset afterward. For the tests, the right-lower part of the data was not used since it suffers a strong influence of a systematic error existing in the system (which has been removed in the latest system). The result can be seen in Fig. 4.4. According to the information in the

TABLE 4.1: AVERAGE TIME PER $m^2$

| Method "a" | Method "e" | Automatic Method |
|---|---|---|
| 9.6 s | 19.95 s | 3.33 s |

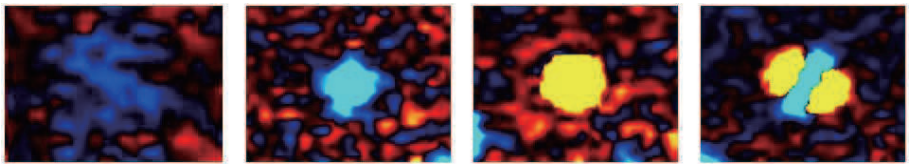

**Figure 4.1:** *Examples of target images*

database the noise was correctly erased, and the remaining signals represent in fact the perimeter of potential targets.

## 4.2 Targets Center Searching Algorithm

An algorithm for finding the theoretical location of a target, where the maximum MMD signal is located, is proposed.

The algorithm starts with the first element in the MMD signals array, used as reference, and looks inside a radius for the surrounding point which has the maximum value. The point with maximum value is the new reference, and the procedure is repeated until no bigger signal is located. These steps are repeated until all MMD signals are used as starting point (Fig. 4.3).

## 4.3 Proposed Method Experiment Results

According to the database information, the marks are correctly placed by the above algorithms (Fig. 4.4). Using the developed algorithms, the calculated POD and FAR for all data used is shown in Fig. 4.5. Comparing to the previous best result (Gryphon, E, 2007), there was a slight improvement in the POD (from 93% to 96%) and a negligible increase in FAR (from 0.12 to 0.14).

As indicated in Table 4.2, the average time for each square meter done with the automatic method is about 3.33 s, which greatly reduces the operation time

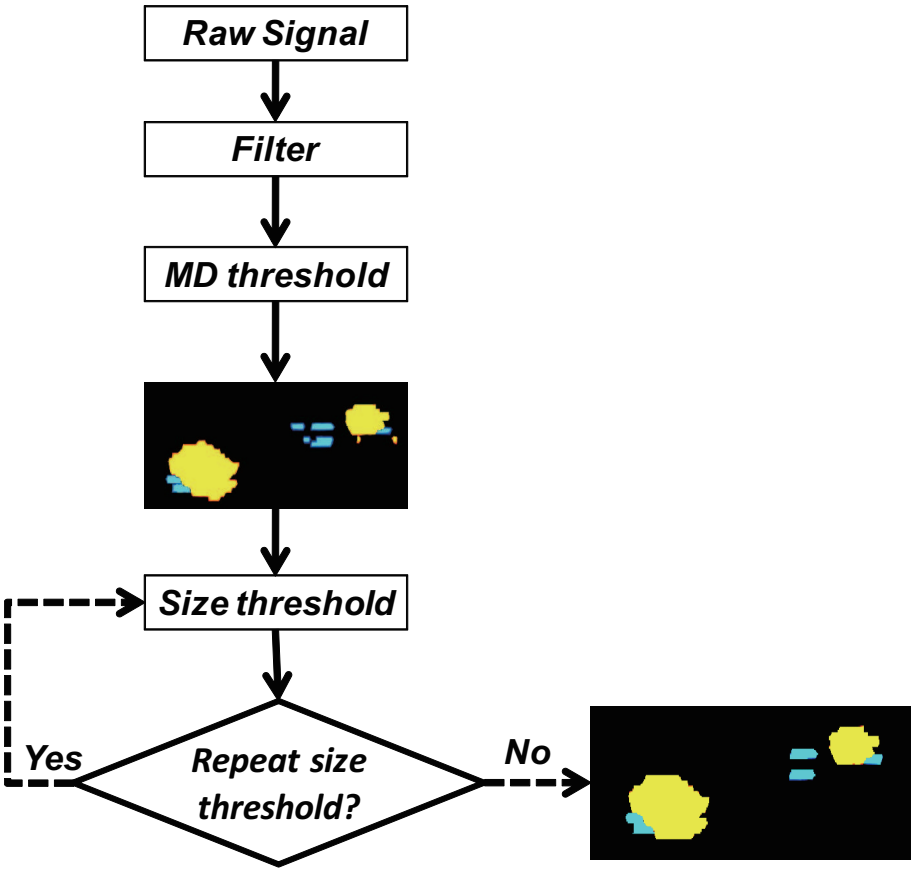


**Figure 4.2:** *Targets perimeter searching algorithm*

compared to the subjects in the experiment in section 3 (which time varied from 9.6 to 19.95 s for completing each $m^2$). Another major advantage is the correct location of the input marks, which done manually by the operator could generate errors.

## 4.4 Marking Trajectory and Sequence Enhancement

The demining operation with Gryphon is mostly composed by arm motion. It was testified that a part of this arm motion can be reduced during the marking task, as described below.

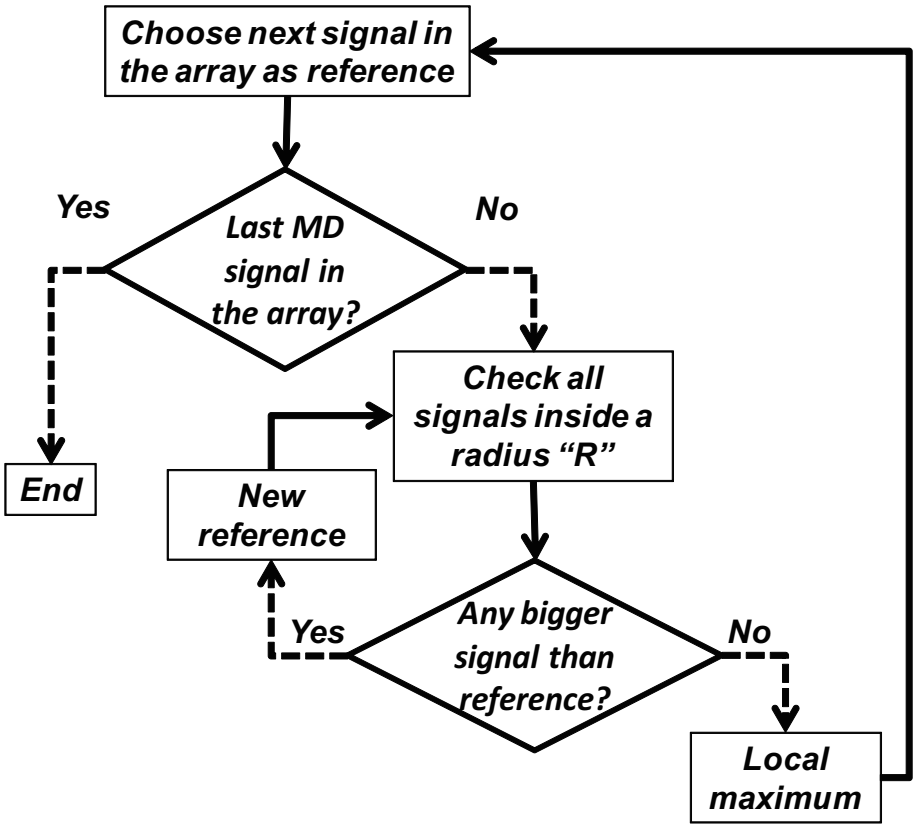


**Figure 4.3:** *Targets center searching algorithm*

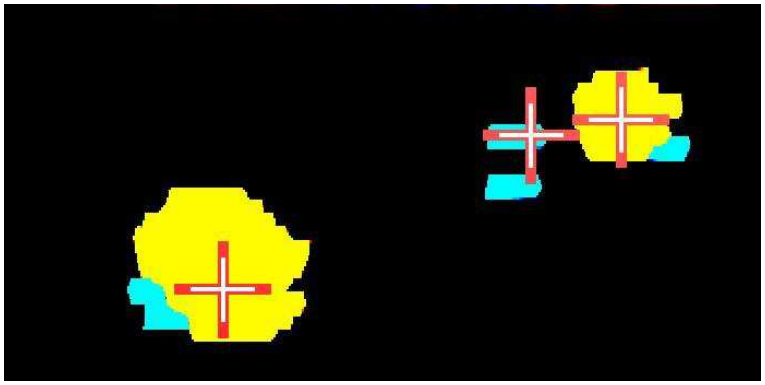

**Figure 4.4:** *Resulting image after applying the targets perimeter and targets center searching algorithms*

### Trajectory Enhancement

The steps for the marking task with Gryphon described in Chapter 2 are performed for each single mark. However, according to the situation, many

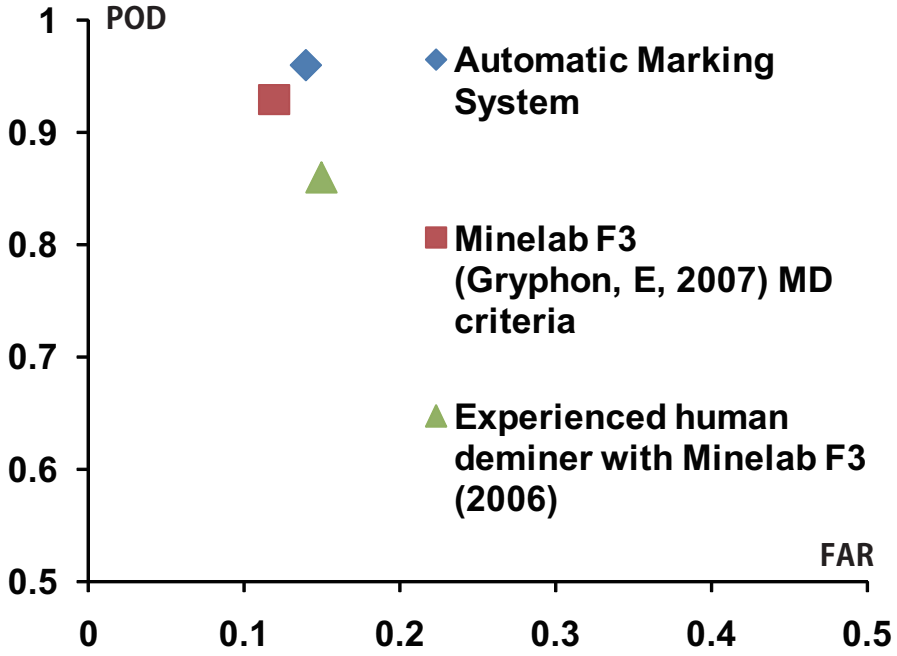


**Figure 4.5:** *Comparison between the best result in the test field and automatic marking system*

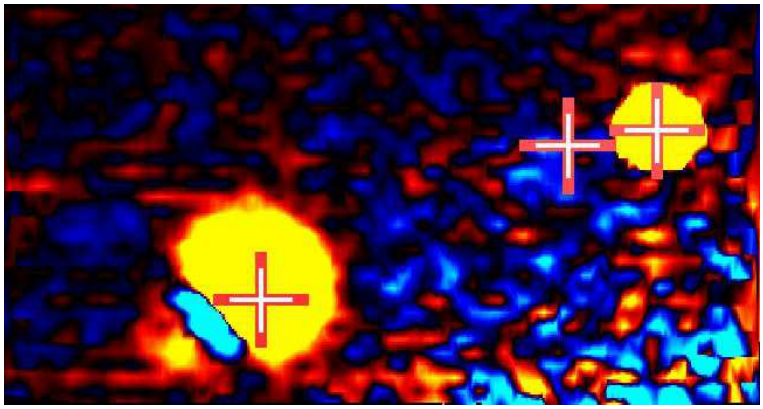

**Figure 4.6:** *Final output image displayed to the operator*

marks may be needed, and the described procedures will be repeated as many times as the input number of marks.

For example, consider Fig. 4.7. If the robot has to draw two marks, the arm will do the following trajectory: P1, P2, P1, P3 and P1. Regarding the shown

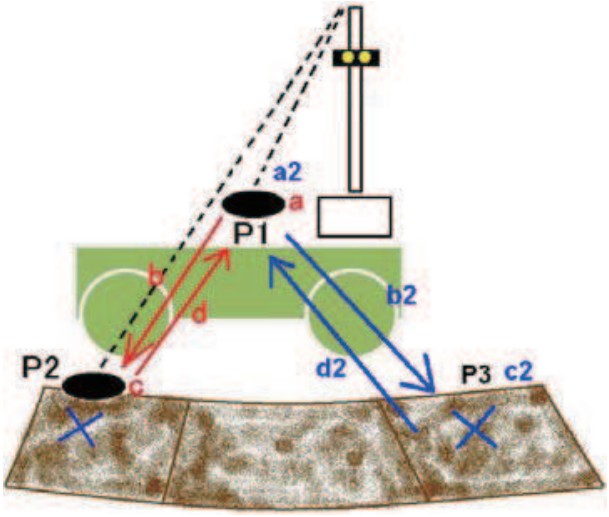


**Figure 4.7:** *Trajectory between two consecutive marks*

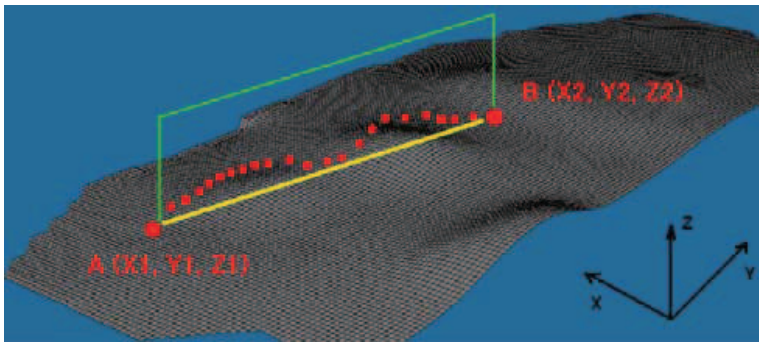


**Figure 4.8:** *Trajectory enhancement*

trajectory problem, the main requisite is that the arm does not go back to the initial position (P1) and goes directly to the next spot.

Once the terrain containing the two points is not flat, a simple straight line cannot be used as a trajectory, and a new path should be generated. The shortest path would be an ideal solution, and many previous works were performed to solve this problem [11] [43] [46] [4], but here as a first approach a simpler algorithm was applied.

Consider Fig. 4.8. The proposed method consists of building a line (shown in yellow) between the actual point (A) and the next point (B). Since we know the coordinates of A (X1, Y1) and B (X2, Y2), we can simply substitute them in

the line equation.

Once we have this line, we make a plane perpendicular to the ground (in green), that will guide the arm until the next point. Finally, the robot calculates the height fields (Z coordinates) in this plane, and moves the arm through these height variations in the terrain.

**Sequence Enhancement**

After proposing an enhanced solution for the marking trajectory, another became evident. Since the robot paints the marks in the inverse order they were input in the interface by the operator, in case we have three or more marks, an enhanced sequence should be considered.

Since asking the operator to input the marks in an optimized sequence would require too much workload and long decision making times, a more reasonable approach was adopted, which is the robot itself computes and chooses the shortest path to go through all the points. Here, many previous works such as [12] were done to find the shortest path passing through all the points. However, this algorithm is complex to be implemented and as a first enhancement proposal, an algorithm similar to [9] was implemented.

Suppose the operator has set the following sequence in the interface: “1”, “2”, “3”, “4”, “5”, “6”, “7” and “8”. With the proposed method, the robot gets the first mark of the sequence (“1”) and calculates the distance between this first mark and the remaining ones. For this first approach, only coordinates X and Y are being used, since variations in Z are usually relatively smaller. In the example shown in Fig. 4.9, the robot finds that the nearest mark from “1” is “3”, and uses this last mark as verified. The robot calculates the total distance obtained with this path having mark “1” as initial point and stores this value. Then, the robot sets the next mark (“2”) of the original sequence as initial point and repeats all the procedures. These procedures are repeated until the final mark (“8”) is set as initial point.

After calculating the total distance of all possible paths (eight in this example), the robot verifies which of them has the shortest path, and reorganizes the original sequence into this enhanced one (Fig. 4.10).

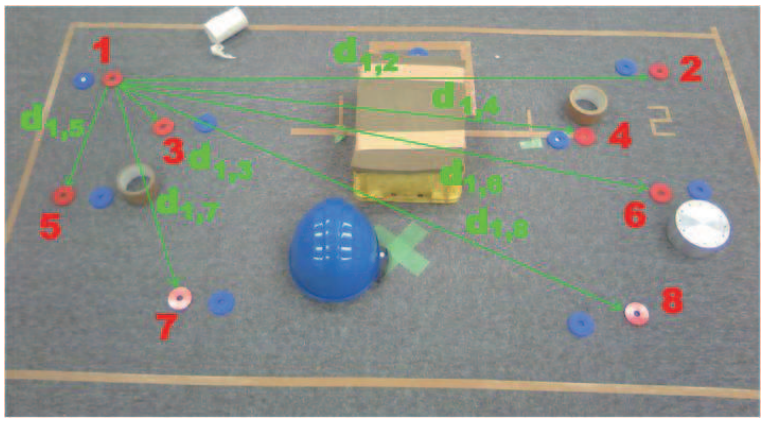


**Figure 4.9:** *Sequence enhancement*

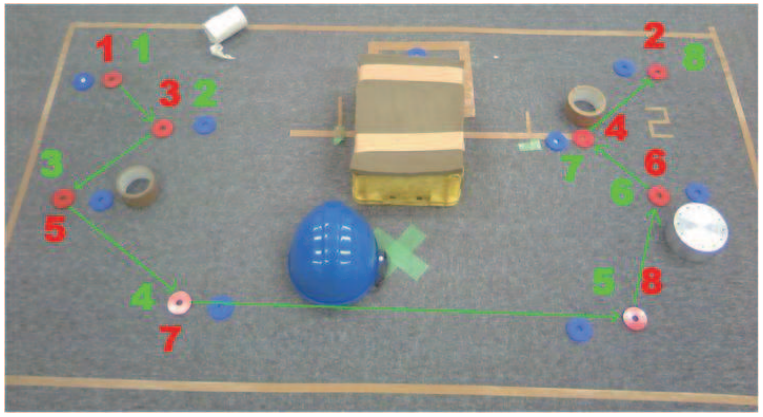


**Figure 4.10:** *Sequence enhancement test*

TABLE 4.2: MARKING TRAJECTORY AND SEQUENCE ENHANCEMENTS EVALUATION

| Nothing | Trajectory | Trajectory and Sequence |
|---|---|---|
| 235 s | 141 s | 120 s |

For quantifying the improvements, an experiment was conducted and shown in Table 4.2.

# Chapter 5

# Proposed Curve Characterization Method

A new, accurate and fast method for estimation of physical properties such as depth and material of metallic targets, as well as discrimination of landmines and metal fragments using MMDs is presented, which takes advantage of high precision scanning of the minefield using robotic manipulator as shown in Fig. 5.1. The very basic concepts are introduced in this chapter and further details can be found in [26] and [27].

## 5.1 Method Description

This Section guides the reader through the very basic concepts of MMD signals and their strong features according to direction. Then, the main parts of the method such as searching criterion and database building are deeply explained.

### 5.1.1 Spatially Represented Metal Mine Detector Signals

Humans usually scan areas swinging the MMD sideways advancing in steps, which the detected signals (called here as V(%)) are transformed into sound and the signals with corresponding positions must be memorized. Robots can also deal with the same task, but with higher precision and repeatability, easily

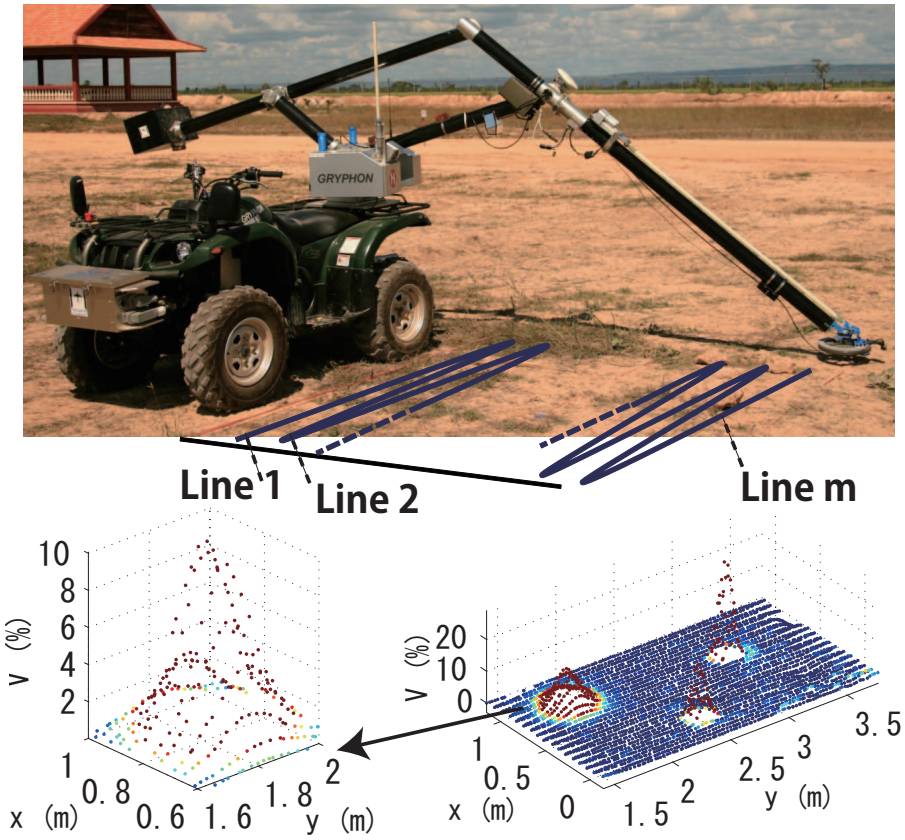


**Figure 5.1:** *Demining robot Gryphon and detected signals*

associating signals with the spatial position, processed in real-time and shown to the operator (Fig. 5.1).

In this research, we define "Spatially Represented Metal Mine Detector Signal" (SRMMDS) as a 3D representation of the MMD signal (Fig. 5.2). It can be observed the SRMMDS change according to the target's properties such as size, shape, buried depth and posture, composing very strong characteristics. This suggests that if a database with these characteristics is composed, discrimination could be done by comparing a SRMMDS obtained in the minefield to the closest match in the database.

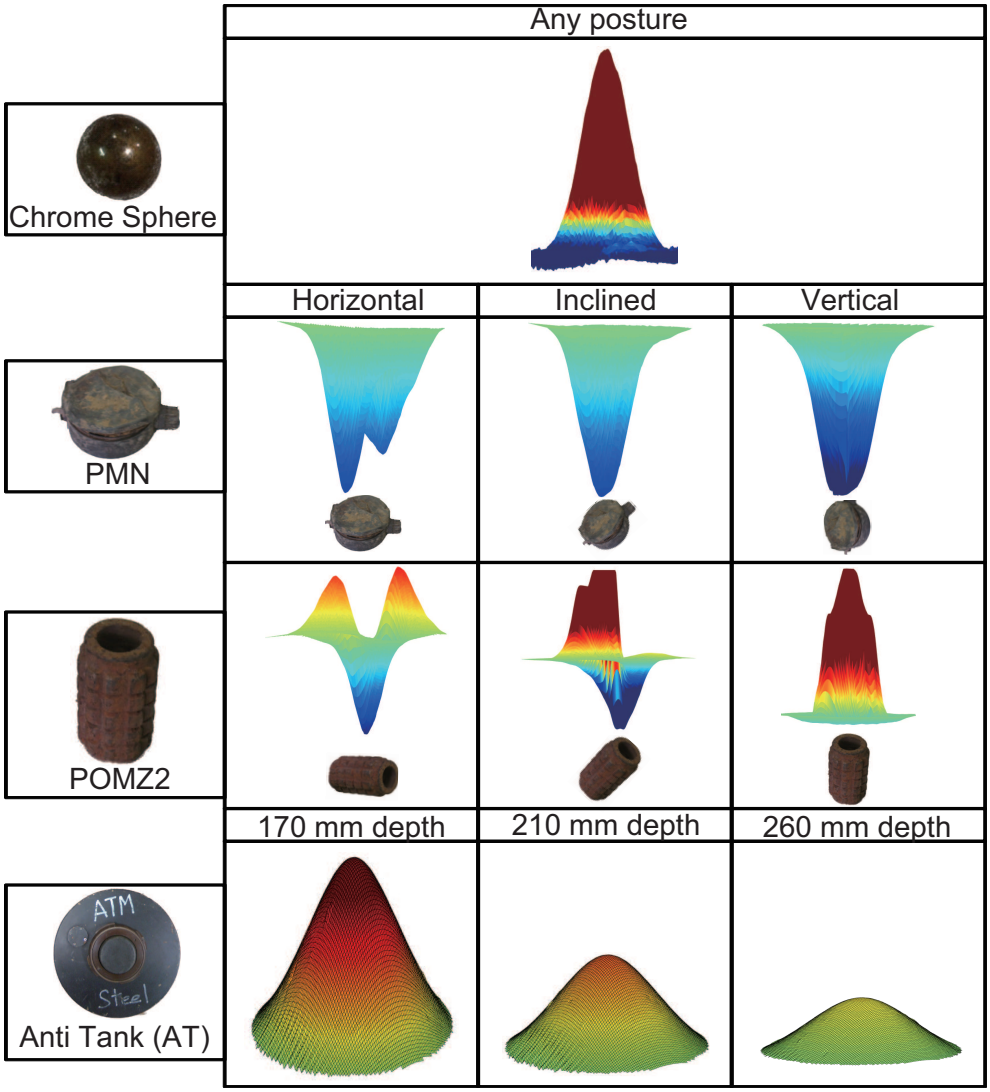


**Figure 5.2:** *SRMMDS for different targets and conditions*

### 5.1.2 Main Axis and Main Characteristic Curves Definition

For each detected SRMMDs, we set a local coordinate O-xyz', with an x'-y' plane parallel to the MMD scanning plane and z' axis passing through the maximum absolute value of the SRMMDS. Consider an orthogonal plane $P_\theta$ to the x'-y' plane that passes through the z' axis and with an angle $\theta$ relatively to the x' axis. The characteristic curve defined in this work $V(r(\theta))$ (Fig. 5.3) refers to the new generated curve, the contour of the intersection of the $P_\theta$ with the

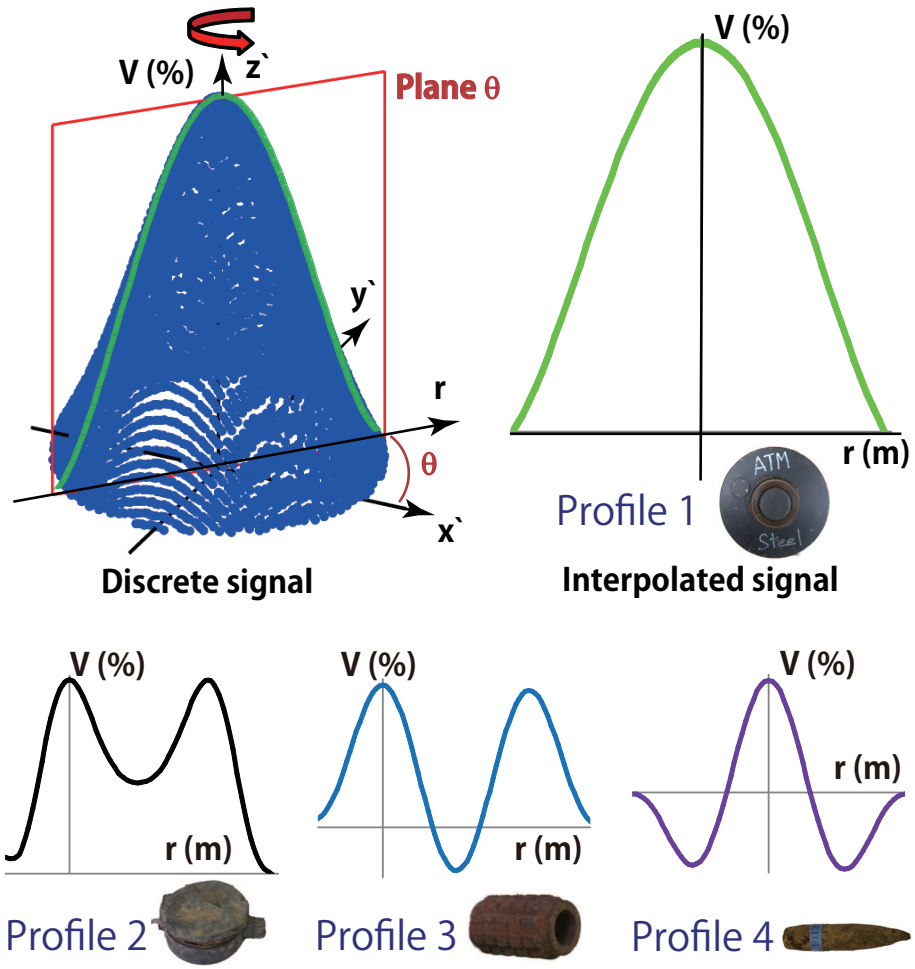


**Figure 5.3:** *Cutting plane and main characteristic curves*

SRMMDS, with new axis r(θ).

Fig. 5.3 shows that the characteristic curves for physically symmetric targets as an AT landmine don't change for all θ, and the SRMMDS can be simplified to one characteristic curve. For non-symmetric targets, characteristic curves change drastically according to θ, but simplifications to a set of minimum curves can be done. As shown in Fig. 5.2, for a particular θ which coincides with the target's longest direction, the curve has many inflections and peaks in relation to other angles. In this work, the characteristic curve with most inflexions and/or peaks is defined as "main characteristic curve" and its axis θ as "main axis", which four main Profiles are possible (Fig. 5.3).

Characteristics curves can be represented by several mathematical relations such as splines and polynomials, in the form of V = f(r(θ)). Since the number of inflections of the characteristic curves is limited, we propose the use of polynomials in the form of eq. 5.1, keeping the signal characteristic and filtering the noisy raw data. All signals are translated so that the maximum peak is located in r=0 and $a_0$ has the maximum absolute MMD value of the signal.

$$f(\mathrm{Y}) = a_0 r(\theta)^0 + a_1 r(\theta)^1 + a_2 r(\theta)^2 + ... + a_n r(\theta)^n \tag{5.1}$$

*where $a_0$, $a_1$, $a_2$,..., $a_n$ = polynomial coefficients*

### 5.1.3 Searching Criterion

A criterion of similarity (or error, *Err*) for the searching in the database between two curves is adopted here as the integral of the difference of their polynomials, as shown in eq. 5.2. A small *Err* indicates good similarity and high possibility of discrimination while higher ones suggest that the target is not part of the database.

$$\mathrm{E}rr(\%) = \int abs(f-g)/h * 100 \tag{5.2}$$

*where f and g = polynomials to be compared*
*h = max[∫ abs(f), ∫ abs(g)]*

### 5.1.4 Database Building

For verifying the validity of the proposed method we first built a database of characteristic curves in the polynomial form for many targets with different depths and postures using a robotic x-y manipulator. Data with weak (V(%)<1%) or saturated (V(%)=100%) signals were removed from the database this time. In total, the database is composed of 34 targets (Fig. 5.4) in several depths and postures resulting in a total of 340 main characteristic curves.

The targets consist of many shapes (sphere, tube, cylinder, cube) and materials (steel, brass, chrome, aluminum and stainless). The depths vary from 10 to 400 mm and several postures (horizontal, inclined $45^o$ and vertical) are


**Figure 5.4:** *Targets used for building the database*

included. All MFs and AT landmines were taken at linear speed of 50 mm/s, 10 mm line step between scan lines and data density of 0.2 points/mm. Some landmines are also included, with variable depths, postures and scan steps. Details are as in Table. 5.1.

### 5.1.5 Depth Interpolation for Characteristic Curves

In the former sections, we mentioned the possibility of having a pre-built database of different targets with physical properties (depth, posture, size, shape, etc) for permitting easy discrimination. On the other hand, it is not feasible to make a database with all possible combinations, but as shown in Fig. 5.5, characteristics curves for a given target keeps the number of concavity and mainly changes its amplitude according to variations in depth. This fact suggests that inputting an $a_0$ in one set of target with its different depths the corresponding curve as well as the depth with that $a_0$ can be generated. For example, in case the input $a_0$ is 80% the estimated depth is around 160 mm for the AT landmine and 80 mm for the MF21 (a type of MF, further detailed in the next sections),

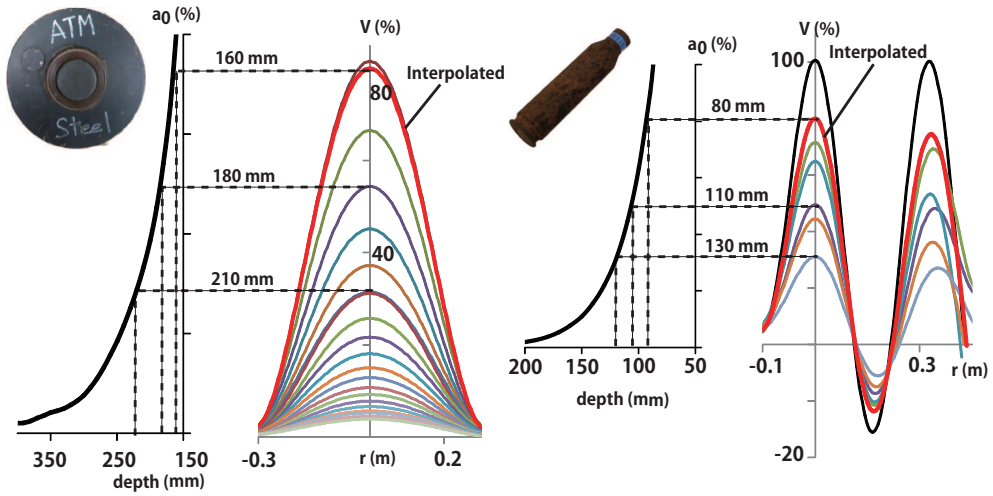


**Figure 5.5:** *Example of depth interpolation for an AT and MF21 target type characteristic curves*

and their corresponding curves are generated in red in Fig. 5.5. This shows that with a limited number of discrete data, reconstruction of characteristic curves for any depth for that particular target is possible, as shows eq. 5.3.

$$f^{t}_{interpolated}(\mathrm{Y}) = f(f(\mathrm{Y})^{t}_{1}, f(\mathrm{Y})^{t}_{2}, ..., f(\mathrm{Y})^{t}_{d}) \tag{5.3}$$

*where target = 1, 2, ..., t*

*depths per target = 1, 2, 3, ..., d*

### 5.1.6 Metal Mine Detector Signal Conditioning

The MinelabF3 Metal Mine Detector [35] was chosen for this experiment. This detector outputs signals in two independent channels ($Ch_A$ and $Ch_B$), which are combined [25] according to eq. 5.4, forming a signal with stronger intensity $Ch_C$ used for comparison in eq. 5.2. However, both $Ch_A$ and $Ch_B$ information are stored in the database and maximum MMD values and depths relation are used for refining and speeding up the search according to eq. 5.5, which only targets with depth error smaller than a threshold value are used for comparison, while others are directly discarded.

TABLE 5.1: DIMENSIONS OF THE TARGETS USED FOR BUILDING THE DATABASE

| **Data No** | **Target** | **Dimensions** | **Material** | **Posture** |
|---|---|---|---|---|
| 1-222 | Bullets and cartridges (MF01 to MF21) | 7-27 mm diameter, 27-114 mm height | Steel | Horizontal |
| 223-254 | Cube | 20 mm edge | Aluminum, stainless, brass | $45^o$ |
| 255-274 | Cylinder | 11 mm diameter, 12.5 mm height | Aluminum, stainless, brass | Horizontal |
| 275-291 | Tube | 11 mm external diameter, 0.5 mm thickness, 12.5 mm height | Aluminum, brass | Horizontal |
| 292-301 | Sphere | 25.4 mm diameter | Chrome | Horizontal |
| 302-305 | ITOP ($I_0$) | 4.8 mm outer diameter, 0.38 mm wall thickness, 12.5 mm height | Aluminum | Horizontal |
| 306-330 | AT | 300 mm diameter | Steel | Horizontal |
| 331-335 | PMN | 112 mm diameter, 56 mm height | Steel | Horizontal |
| 336-340 | PMN2 | 125 mm diameter, 65 mm height | Steel | Horizontal |
| 341-362 | Other Landmines (P-40, PSM1, MD82B, etc) | Many variations | Steel | Many variations (horizontal, verticaal, $45^o$ in xz |

$$Ch_C = Ch_B - Ch_A - median(Ch_B - Ch_A) \tag{5.4}$$

$$Depth\ error(cm) = abs(depth_B(a_{0B}) - depth_A(a_{0A})) \tag{5.5}$$

### 5.1.7 Database Integrity

The database was built with a limited number of data taken for discrete depth so that an interpolation method as explained previously should be valid to reconstruct a corresponding curve at a given arbitrary depth. Notice that many minefield conditions exist, but in this work we focus on limited conditions such as targets located in flat grounds, no other metals nearby, scans can be done without obstacles and different types of soils are not accounted. All these parameters are important to be analyzed and will be considered in future work.

In order to verify the "depth interpolation method" and also check the integrity of the database, the following test was conducted: depth interpolation, with search of the closest target. For each data number N, consider it is a detected signal and exclude it from the database, performing the searching in all the remaining data, looking for the target with smallest error. Since no extrapolations are done, data of the shallowest and the deepest depths of each target were used only for interpolation, without being input for estimation. The depth error threshold adopted for eq.5.5 is 17 mm, which means, if a target has depth error higher than this value than it is discarded as potential candidate for estimation, guaranteeing good estimation levels. The method, then, outputs (estimates) two main information: a) depth (interpolated) and b) target (which is a combination of material, size, shape, etc). Since the method estimates a target, theoretically we could directly discriminate the unknown signal as this target, i.e, as landmine or MF. However, safety issues must be considered for reliable discrimination and for this reason we will deeply discuss this possibility in the next sections. For now, we only analyze the estimated depth and material.

The estimation results are shown in Table. 5.2 and Fig. 5.6. Comparing to the existing method [38], more targets were analyzed and a greater mate-

TABLE 5.2: DATABASE INTEGRITY: MATERIAL AND DEPTH ESTIMATION RESULTS

| | **Existing Method [38]** | **Proposed Method** |
|---|---|---|
| **Average depth error (mm)** | - | 4 |
| **Maximum depth error (mm)** | - | 39 |
| **Estimation time per target (s)** | >96 | 1 |
| **Steel (%)** | 22/23 = 95 | 197/197 = 100 |
| **Aluminum (%)** | 11/23 = 48 | 14/19 = 74 |
| **Stainless (%)** | - | 11/17 = 65 |
| **Brass (%)** | 5/23 = 21 | 15/19 = 79 |
| **Chrome (%)** | - | 7/8 = 88 |

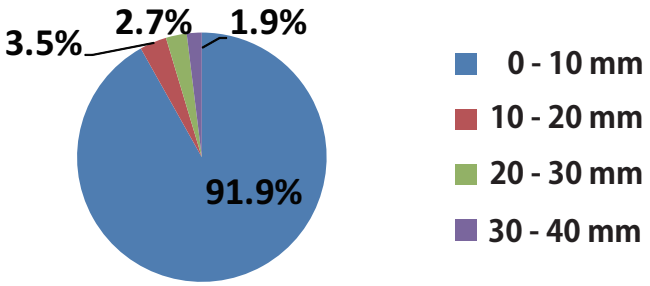


**Figure 5.6:** *Database data depth error distribution*

rial estimation rate can be observed: 95 to 100% for Steel, 48 to 74 % for Aluminum and 21 to 79 % for Brass. Even though [38] doesn't mention Stainless and Chrome, in this work the estimation was high, with 65 and 88 %, respectively. The depth estimation resulted in a average error of 4 mm and maximum 39 mm. More than 91% of the data resulted in depth error below 10 mm, while only 1.9% has the error between 30 and 40 mm, what can be considered satisfactory for supporting the use of GPRs. Even though [38] doesn't mention the average depth errors, the time for estimation is hugely smaller for the proposed method (1 s per target) against many minutes per target for the existing method.

All cases which depth error are between 30-40 mm are detailed in Fig. 5.7. The signals in the database are interpolated very accurately generating curves

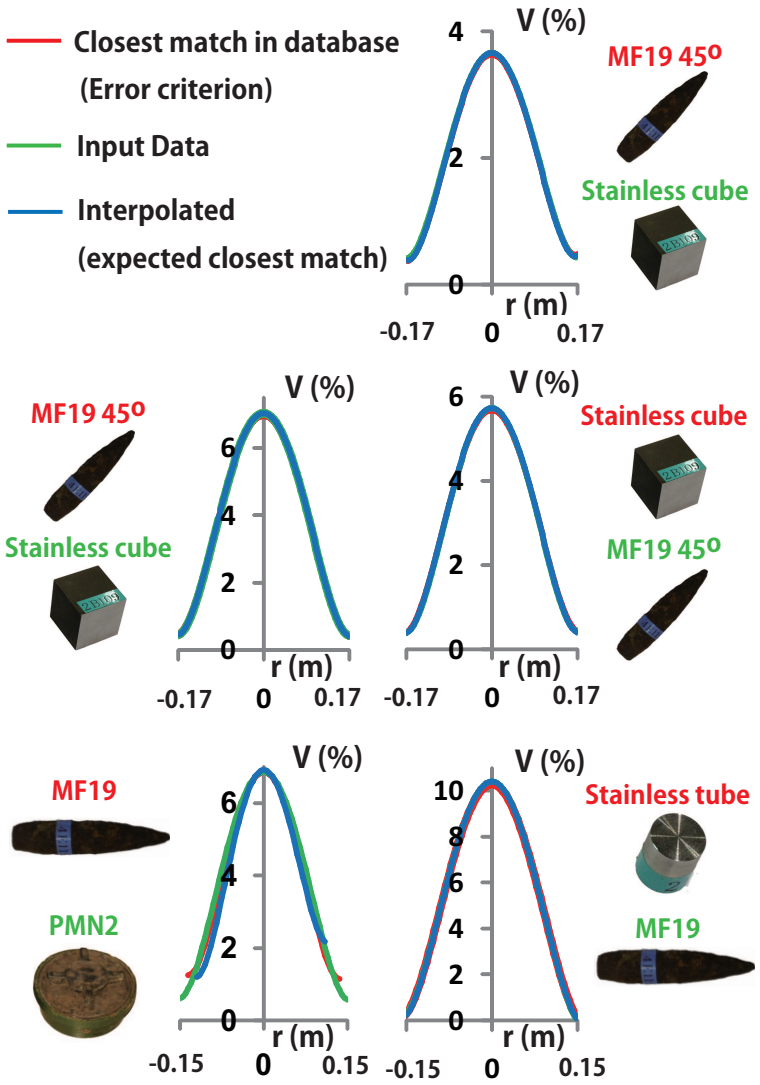


**Figure 5.7:** *Targets with 30-40 mm depth errors*

as expected. However, one can observe that for all cases, the target's profiles are like Profile 1 (Fig. 5.3), which is the profile with least inflections and peaks, being very poor in characteristics. One can notice that the MF19 is responsible for many high depth error cases. Even though the MF19 is not physically symmetric, its detected SRMMDS for both horizontal and $45^o$ under certain conditions are nearly symmetric, generating profiles similar to targets such as cubes, tubes and the PMN2 landmined, becoming difficult to estimate.

## 5.2 Landmine/Metal Fragment Discrimination Method

As mentioned in the previous subsection, the presented method outputs a target and a depth. Discrimination could be simply done by discriminating the detected signal as the output, but since errors are unacceptable, further safe parameters (margins) are introduced.

### 5.2.1 Basic Discrimination Scheme

The basic scheme for discrimination of sensed signals can be implemented as follows:

Step 1: Calculate the measure of error *Err* (eq. 5.2) for the characteristic curve of the sensed signal against all data in the pre-built database.

Step 2: Select the data with minimum *Err*, as candidate for discrimination. This scheme can result in 4 possible cases, namely R1, R2, R3 and R4, as shown in Table 5.3 (a) and illustrated in Fig. 5.8. Cases R1 and R4 result in right discrimination and R2 although resulting in FALSE POSITIVE thus increasing FAR is still acceptable. However, case R3 find a MF data as the closest match for a landmine obtained signal causing FALSE NEGATIVE (mine judged as MF) result, which is by no means allowed in this research, as well as in any other demining research for that matter.

In this research, the FALSE NEGATIVE situation is overcome by flagging as "potential mines" all MF data that can cause the case R3, resulting in a new case R3', shown in Table 5.3 (b). The identification of R3 and flagging to R3' is done during the "database building and conditioning process" explained later.

### 5.2.2 Practical Discrimination Process

Safety margins are set for increasing safety and reliability of the method. Considering these parameters, the whole discrimination process is described.

a) Measure of Difference of Errors (*dE*)

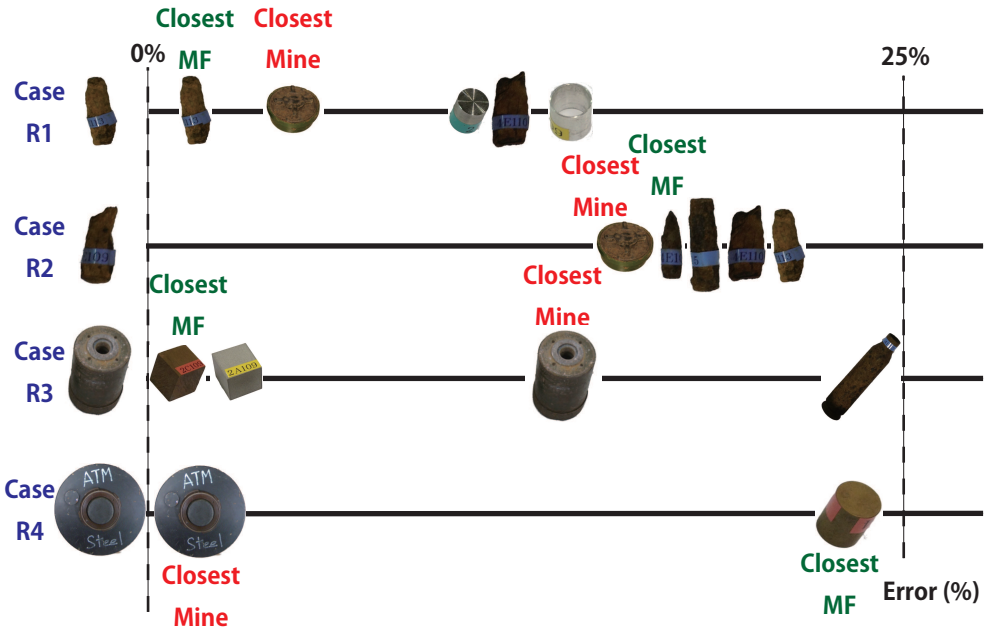


**Figure 5.8:** *Basic discrimination cases (R1, R2, R3 and R4) and target distances according to Err*

In Fig. 5.8, one can notice that the measure of error *Err* of some MF is very close to Mines, as in the example in case R1. In a real situation as we don't know what the test subject is, in order to prevent any misjudgment we will introduce a measure of difference of errors (*dE*), calculated by eq.5.6, as the difference between the measure of error *Err* of the closest MF ($Err_{closestMF}$) and the *Err* of the closest landmine ($Err_{closestlandmine}$).

$$dE = Err_{closestMF} - Err_{closestlandmine} \tag{5.6}$$

A threshold for *dE*, $dE_{threshold}$, is also defined for flagging as "potential mines" all MFs which $|dE| < dE_{threshold}$, reducing chances that landmines are discriminated as MFs.

b) Measure of Confidence ($E_{threshold}$)

Another case that can be observed in Fig.5.8, is that the measure of error Err of the closest target (called $E_{closest}$ hereafter) can be sometimes too high, indicating there is no good match in the database. This can mean the data contains

TABLE 5.3: DISCRIMINATION CASES ACCORDING TO *Err (%)*.

| (a) Basic discrimination cases according to *Err (%)* | | | | |
|---|---|---|---|---|
| **Case** | **Test Subject** | **Closest Match** | **Discrimination Result** | |
| R1 | MF | MF | TRUE NEGATIVE | Good: decrease FAR |
| R2 | MF | Mine | FALSE POSITIVE | Still acceptable: increase FAR |
| R3 | Mine | MF | FALSE NEGATIVE | Not acceptable: a missed mine |
| R4 | Mine | Mine | TRUE POSITIVE | Good: increase POD |
| (b) After database condition process, case R3 become R3' | | | | |
| R3' | Mine | "Potential mine" | TRUE POSITIVE | Good: increase POD |

too much noise or the target is degraded, being a potential risk. In this research, we adopt a safety criterion that considers the test subject as "potential mines" when $E_{closest}$ is greater than a given threshold $E_{threshold}$ (to be determined by experiments). Fig. 5.9 shows some examples of MFs very similar to landmines.

c) Discrimination Steps

The final scheme for discrimination of sensed signals, taking in account the above measures is implemented as:

Step 1: Calculate the measure of error *Err* of the obtained signal (sensed signal) against all available data in the database

Step 2: Select the data with minimum *Err*, i.e., $E_{closest}$. If $E_{closest} \geq E_{threshold}$, consider the sensed signals as "potential mine" and end discrimination.

Step 3: Calculate the Measure of Difference of Errors ($dE$), and make final decision.

-If $dE > 0$: consider the sensed signal as mine.

-If $dE < 0$ and $|dE| > dE_{threshold}$ : consider the sensed signal as MF.

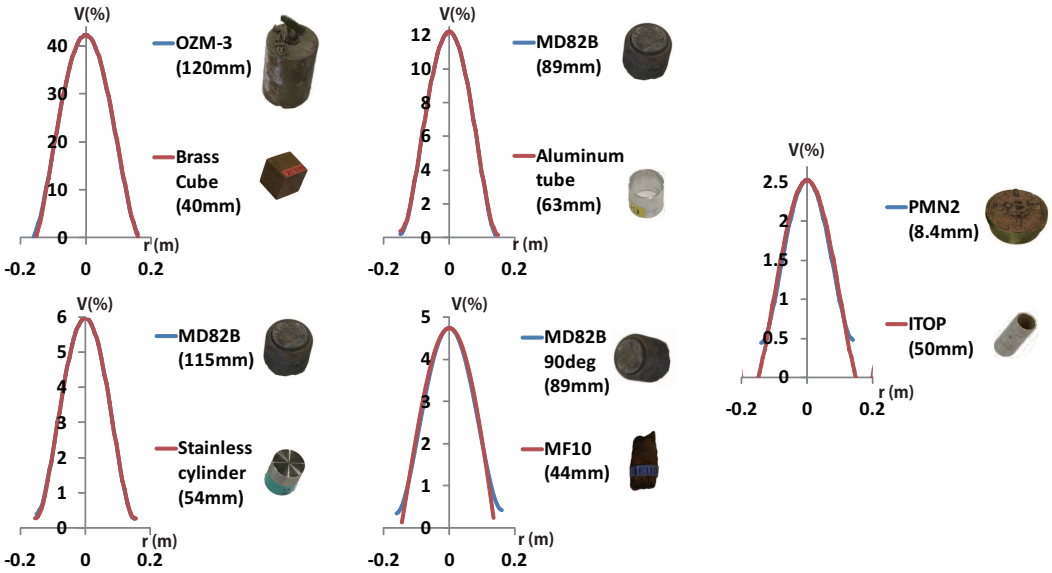


**Figure 5.9:** *Examples of MFs considered as "potential mines" by the $E_{closest}$ and $|dE| < dE_{threshold}$ criteria. Targets and corresponding depths are shown in parenthesis. Note that the ITOP conceived for an International Test Operations Procedure project as the metal content of larger stimulant mines shows a SRMMDS very similar to some landmines (such as PMN2) and it is also classified as "potential mine" by this criteria*

-Otherwise: consider the sensed signal as "potential mine".

d) Measure of Confidence Setting

For each given data N in the database (Table 5.1, N=1 to 362), consider it as a test subject and calculate the measure of error *Err* against all other data in the database. The Cases (R1, R2, R3 and R4) described in former sections are analyzed and shown in Fig. 5.10, which Data is sorted for easier visualization.

For determining $E_{threshold}$, several values from 0 to 100% were set and corresponding values of False Positives and True Positives were observed. As shown in Fig. 5.11, $E_{threshold}$=10% is the value which maximizes the difference between True Positives and False Positives.

Repeating the analysis with the interpolation method, smaller values of *Err* are obtained and a new threshold $E_{threshold}$ =15% (Fig. 5.12) and R1, R2, R3 and R4 Cases are set. Since no extrapolation is done in the interpolation, part

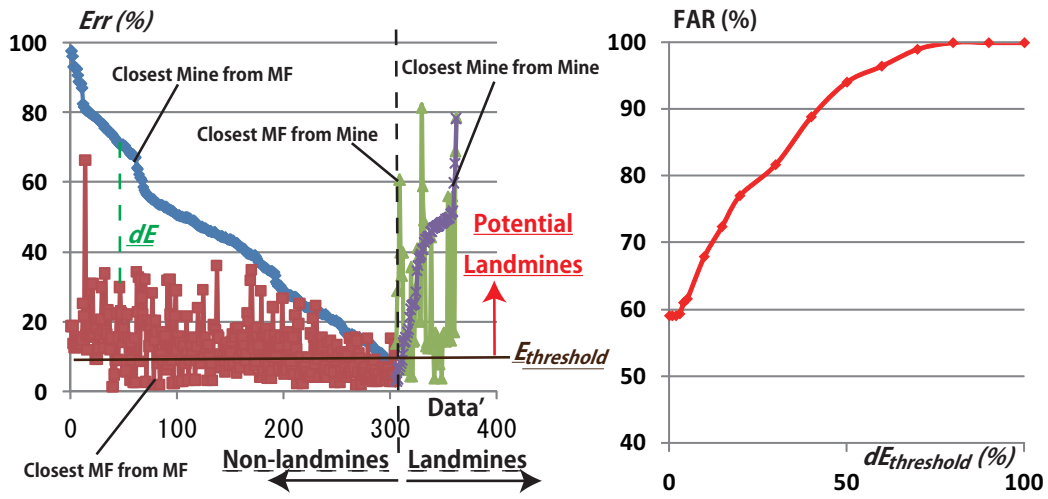


**Figure 5.10:** *Resulting Errors of closest MFs and mines from each data. According to the adopted safety margins $dE_{threshold}$ and $E_{threshold}$ different FAR can be observed*

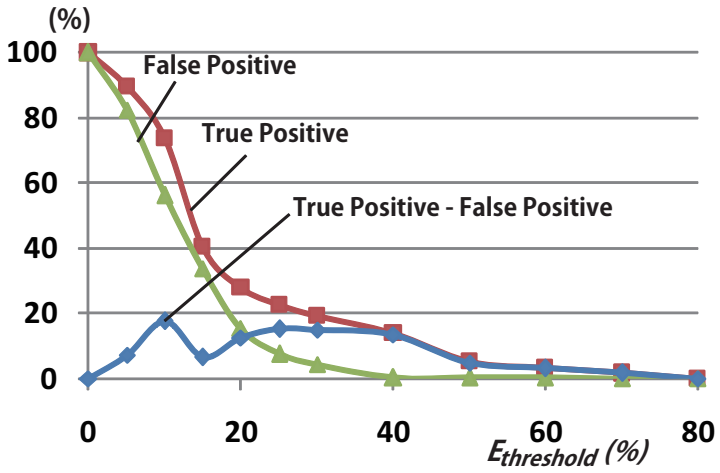


**Figure 5.11:** *Variation of False Positive and True Positive values according to $E_{threshold}$ (discrete case)*

of the data (deepest and shallowest data of each target) is not used. Since depth estimation errors are possible, depth error margins are also considered and the trade-off analyzed as shown in Fig. 5.13. For interpolated cases, one can observe much lower FAR levels comparing to "Discrete Data 10 mm" case. Fig.12 shows a FAR analysis done with the data in the database in laboratory. Since

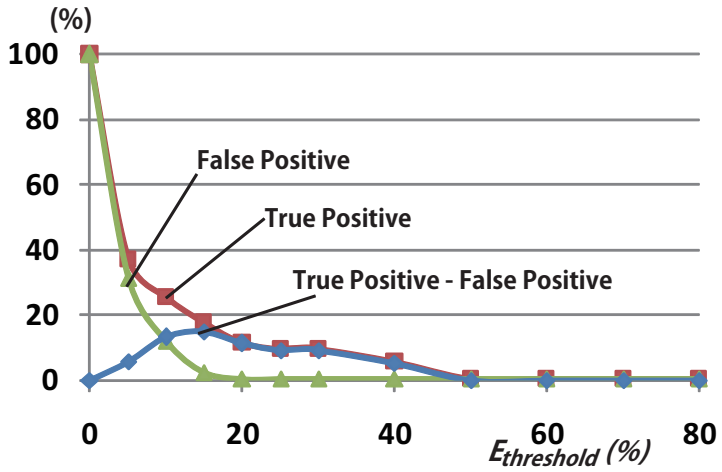


**Figure 5.12:** *Variation of False Positive and True Positive values according to* $E_{threshold}$ *(interpolated case)*

"potential mines" were flagged with criteria shown before, Fig.5.10 shows all cases in which False Negatives never occur, even if $dE_{threshold}$ = 0. However, in real demining operations using $dE_{threshold}$ = 0 is unacceptable and a convenient safety margin must be set. In former sections, we showed that ITOP is very similar to landmines and it is considered as "potential mine" in discrete case by $|dE| < dE_{threshold}$ criterion when $dE_{threshold}$ 10%, so we adopt $dE_{threshold}$ = 10%. For interpolated case, ITOP is "potential mine" by direct search with eq.5.2 and $dE_{threshold}$ = 0 would be enough, but we adopt a minimum of $dE_{threshold}$ = 5%. Moreover, since the maximum depth estimation error of this method is 40 mm (as will be shown in the next section), this Depth margin is adopted in real operations.

## 5.3 Experiments

This subsection presents experiments with the proposed method. First, the subproducts of the method (material and depth) are tested, and later, discrimination with test field data are done.

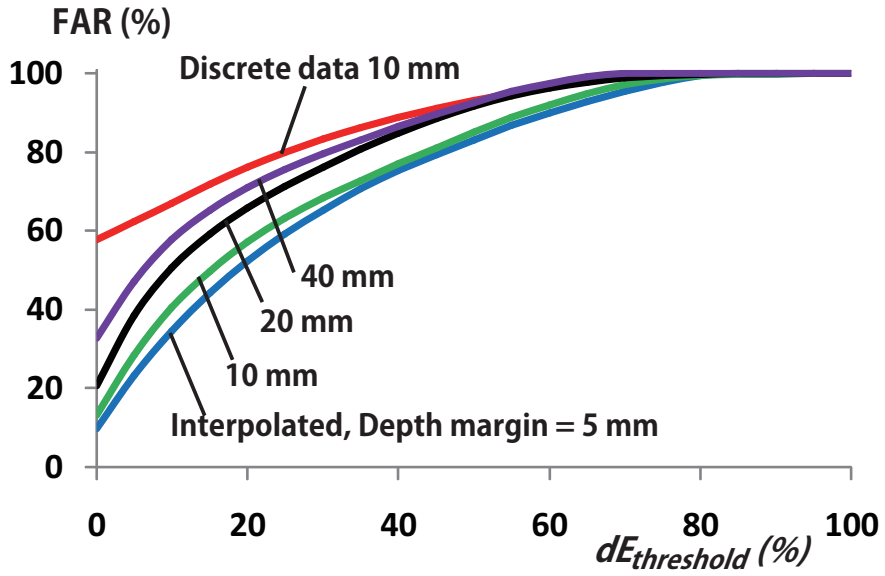


**Figure 5.13:** *Trade-off of adopted safety margins and FAR. For all cases, FAR is generated with no occurrence of False Negatives due to the discrimination criteria and safety margins adopted*

### 5.3.1 Material and Depth Estimation

The database was built using a x-y manipulator for taking precise data in laboratory. We need to test the behavior and robustness of the method when data taken with Gryphon is input. Some MFs, AT landmine and ITOP simulate target (a standard target for tests, simply called "ITOP" hereon) data were obtained with Gryphon at different depths and compared to the database for depth and material estimation. ITOP is a very important target to be tested for it has the metal content typical of a class of mine [6]. It is also known as "ITOP inserts" since it is conceived for an International Test Operations Procedure project as the metal content of larger simulant mines. There are several types of ITOPs [6], with different levels of detectability and metal contents. In this work, the used ITOP is the type $I_0$ with a "hard to detect" level of detectability with dimensions as Tab. 5.1.

In total, 37 data (16 MF data at 70 mm depth, 10 ITOP data at 60 mm depth and 11 AT data from 150 to 250 mm depth) were obtained with Gryphon's default scanning settings: linear speed of 50 mm/s, 40 mm line step between scan

lines and data density of 0.2 points/mm. All data is filtered, main axis extracted, translated to the origin according $a_0$, smoothed and interpolated. The results are shown in Table 5.4. According to the results, the average depth estimation error increased considerably, when comparing to the database integrity experiment (from 4 to 19 mm). However, 19 mm average error can be still considered a low error and satisfactory for real demining operations, for permitting quick use of GPRs and landmines removal by human operators. The maximum error kept almost unchanged (from 39 to 41 mm). Fig. 5.14 shows the maximum depth error case, which a MF6 had as closest target a MF1 in $45^o$. In this case, the calculated *Error* between the curves is 37%, what is too high, what suggests great chances of wrong estimation, making the *Error* value an important parameter for reliability and must be explored in future work. Deep (AT landmines) and shallow targets (MFs) were correctly estimated according to depth with small errors, i.e, no MF was estimated as deep and no AT was estimated as shallow. It is important to notice that some experienced deminers are able to estimate depths only with a metal detector alone, however, this task is accomplished after several extra swings, slowing down operations in some seconds or minutes. In this work, the same task can be accomplished with less than 1 s for searching and one single scan.

The metal estimation results show that steel could be estimated in 100% of the cases (MFs and AT landmines), while aluminum couldn't be estimated. However, in this experiment, ITOP was used as example of aluminum, but its shape and size are projected to be very difficult to be detected and also to simulate some landmines signals [6] (made of steel), what was a very particular case of aluminum-made target. In fact, 2 of the ITOP data had as closest target the PMN2 landmine, proving the similarity in signal of ITOP and landmines. Some examples of wrongly estimated ITOP are shown in Fig. 5.14.

The experiments show that steel can be estimated with high accuracy. Even though ITOP is difficult to be directly estimated, the depth for this and other targets can be estimated with small errors. Moreover, observing all cases with highest *Errors*, it is emphasized that targets with characteristic curve as Profile 1 have poor characteristics, leading to difficult estimation.

TABLE 5.4: DEPTH AND MATERIAL ESTIMATION EXPERIMENT RESULTS

| | |
|---|---|
| **Average depth error (mm)** | 19 |
| **Max depth error (mm)** | 41 |
| **Steel (%)** | 27/27 = 100% |

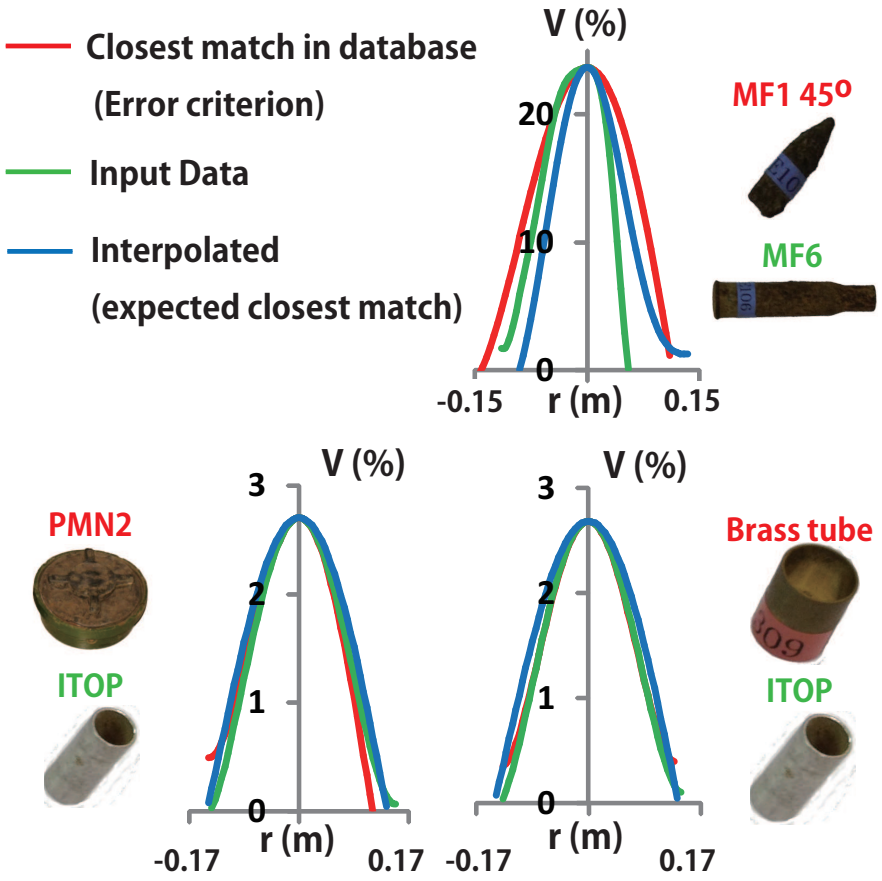


**Figure 5.14:** *Target with maximum depth error (top) and examples of ITOP wrong estimation (bottom)*

### 5.3.2 Landmine and Metal Fragment Discrimination

In this section, we will use data taken in a test field [41] carried out at Croatia in 2007. This test was conduct by the actual robot Gryphon, and consisted of scanning uneven lanes of different soil properties, where several metal clutters and ITOP containing landmines surrogates were buried in random positions at

depths between 1 and 14.5 cm. Among the total 38 targets per lane X 6 lanes (180 data in total, which 120 were ITOP), 14 ITOP containing landmines surrogates and 14 MF (bullets, rockets, etc) data is chosen to be applied as input to the proposed discrimination method. The data was chosen so that no other metal fragments were nearby and it was totally located inside a standard scan area ($2m^2$), avoiding cut data. The safety margins and results are shown in Table 5.5.

The adopted safety margins guarantee correct detection of all ITOP data as "potential mines". In laboratory, all ITOP data (discrete and interpolated cases) has as closest target the MF10 (cartridge, shown in Fig. 5.4 and Table 5.1). In this experiment with ITOP data from the test field, 6/14 for discrete and 12/14 data for interpolated case pointed the MF10 as closest target by direct search with eq.5.2, showing consistency with laboratory environment. The ITOP in the vertical posture in the database (in air) and the safety margin criteria are valid for correct discrimination of data obtained with Gryphon in soil.

Great part of the MFs were discriminated as "potential mines" in the discrete case due to $E_{threshold}$ criterion (what indicates that there are no similar targets in the database), which detects 8/14 in the discrete and 5/14 MFs in the interpolated case. This result is expected by the safety adopted in the method and adding similar targets information in the database would result in more accurate discrimination.

2/14 MFs were considered as "potential mines" by $dE_{threshold}$ criterion, being too close to some landmines. Since no size, shape and material information of the MFs in the test field is available we will call them as "MFX" and "MFY". In the interpolated case, MFX was considered a "potential mine" for being too similar to a MF13 (cartridge, shown in Fig. 5.4 and Table 5.1), also considered a "potential mine" for being too similar to a PMN2 landmine (Fig. 5.15). MFY was discriminated as landmine by direct search, which a PMN2 was pointed as closest data (Fig. 5.15).

One can observe the better performance of the interpolated method, which generates lower FAR levels. Finally, time for discrimination is also a great advantage of this method, which takes 1 s per target, what is very fast comparing to method [38], which takes more than 96 s per target.

TABLE 5.5: PARAMETERS ADOPTED AND RESULTS OF THE PROPOSED METHOD

| | **Discrete** | **Interpolated** |
|---|---|---|
| **$E_{threshold}$(%)** | 10 | 15 |
| **$dE_{threshold}$(%)** | 10 | 5 |
| **Depth margin (mm)** | 40 | 40 |
| MFs discriminated as "potential mines" according to $E_{threshold}$ criterion | 8/14 | 5/14 |
| MFs discriminated as "potential mines" according to $dE_{threshold}$ criterion | 5/14 | 1/14 |
| MFs discriminated as landmine by closest data in database | 0/14 | 1/14 |
| **FAR(%)** | **13/14 = 92%** | **7/14 = 50%** |
| ITOP discriminated as "potential mines" according to $E_{threshold}$ criterion | 3/14 | 0/14 |
| ITOP discriminated as "potential mines" according to $dE_{threshold}$ criterion | 9/14 | 13/14 |
| ITOP discriminated as ITOP itself in vertical posture by closest data in database | 1/14 | 1/14 |
| Discriminated as landmine by closest data in database | 1/14 | 0/14 |
| **False Negatives** | **0/14** | **0/14** |
| Time for discrimination/target (s) | $< 1$ | $< 1$ |

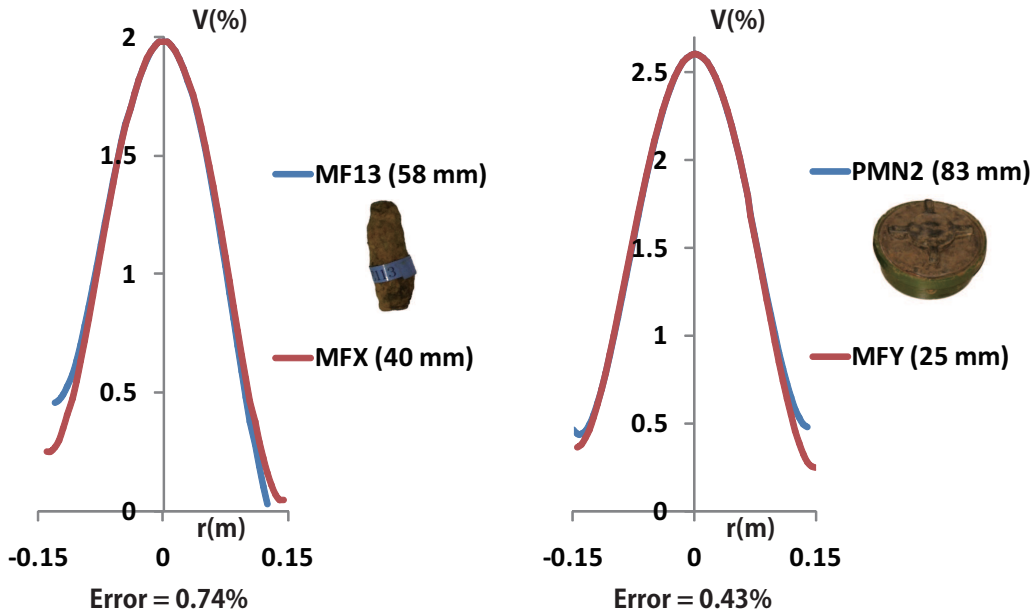


**Figure 5.15:** *Examples of FAR: fragment discriminated as "potential mine" (left) and fragment discriminated as "landmine" (right). Depth of each target is shown in parenthesis. MFX and MFY are two MFs from the test field, which size, shape and material are not available*

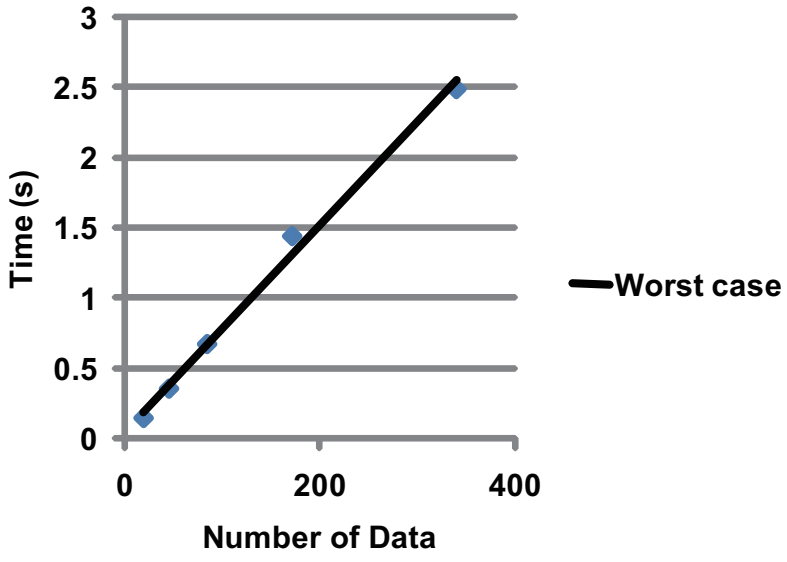


**Figure 5.16:** *Average time for estimation of a target per number of data in the database (worst case)*

### 5.3.3 Database Searching Time

In the previous sections it was shown that the searching for each target could be performed in an average of 1 s/target, considering 362 data in the database and the speeding criteria adopted in this work. However, if we analyze the slowest time (worst case) for different database sizes, we can observe a linear relation, as shown in Fig. 5.16. This proves that even if very large amount of data is available, the searching in the database can be done parallelly, further decreasing the searching time.

## 5.4 Experiments With Noisy Data

The previous experiments showed satisfactory results, but more extreme conditions are possible in real operations. Detected signals can contain some noise and discrimination affected. This section analyze how much discrimination change by inputting noisy signals.

### 5.4.1 Types of Noise

There are many sources of noises that affect MMDs: soil, electrical, mechanical. Noises caused by soil (Fig. 5.17) are the most common and investigated in demining operations. According to the type of soil, its humidity and materials in its composition, noise levels can vary drastically. These levels can be filtered using the "ground balance" (or "ground compensation") [17] and the "median" techniques [25]. Noises shift signals but without changing their features, what is fundamental to the proposed method. Some MMDs are equipped with a ground balancing function, which compensates these noises and permits correct detection of targets. Gryphon can also use this function, but instead, the median value of the whole scanned area is used for compensating these noises [25], correctly balancing signals in a similar way as the ground balancing function, as shown in Fig. 5.18, with a target scanned in soil using a) no compensation, b) ground balancing and c) median. To illustrate that the median can correctly compensate signals and allow correct discrimination in soil, two MFs, MF18 and MF21, were tested (Fig. 5.19), buried at depths of 8 and 10

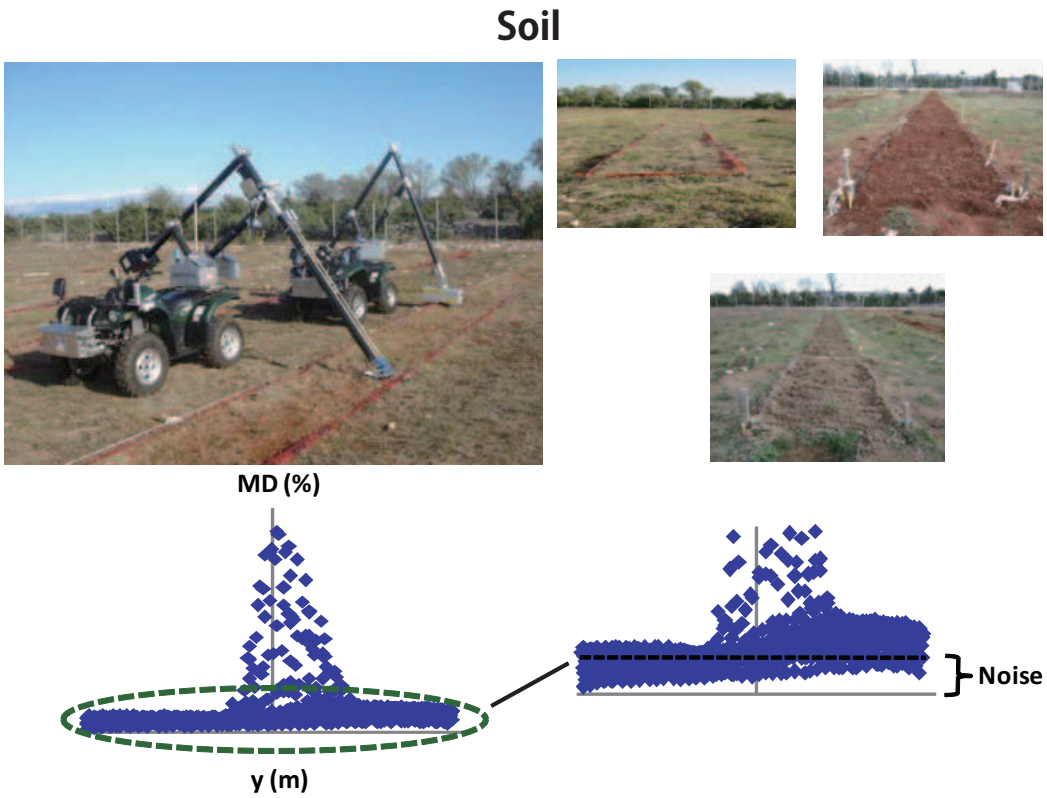


**Figure 5.17:** *Noise caused by soil*

cm, totalizing 19 data. For all cases, the proposed method could discriminate (resulting from the closest data by search in database) the targets in soil as exactly the ones in the database. This experiment shows that signal features, if correctly filtered, are kept in presence of noise from soil, showing the potential of the method in soil.

Noises caused by electrical systems are also possible and in fact a serious one affects Gryphon, causing systematic noises. When the ATV engine in on, the current circulating in the coil causes high interferences in the detected signals by the MMD, as shown in Fig. 5.20. The Figure shows a case which an area with no metals was scanned and regions close to the engine end up resulting in high signals. The calculated gradient vectors point to the noise source, and the engine is clearly pointed. This error is unacceptable since it masks mines in these noisy regions and must be eliminated by shielding the engine coil. Other

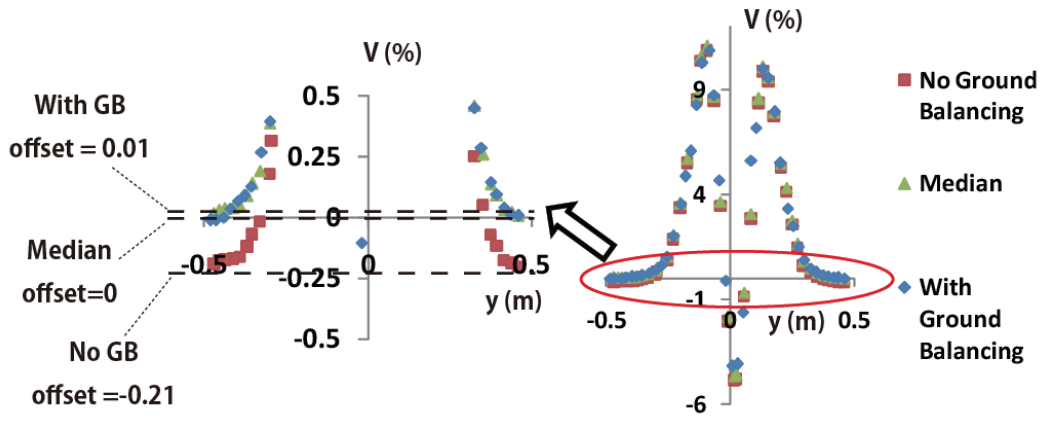


**Figure 5.18:** *Noise balanced without changing its features. "Median" and "Ground Balancing" result in very similar offsets. In relation to the median method, the GB has an offset difference of 0.01%, while for no GB, -0.21%*

**Figure 5.19:** *Experiments with targets in soil*

possible interferences are caused by nearby electrical devices such as antennas, radios, power lines, etc. These devices also cause high noise levels and some techniques such as the "noise cancelling" [36] exist. In the case of Gryphon, this noise is not common since the robot operates in mine fields where no such devices exist.

The last and also occurring type of noise in Gryphon is the one caused by mechanical vibrations, such as wind, causing more random noises and which are more difficult to predict and filter. This kind of noise is analyzed and dis-

cussed in this section.

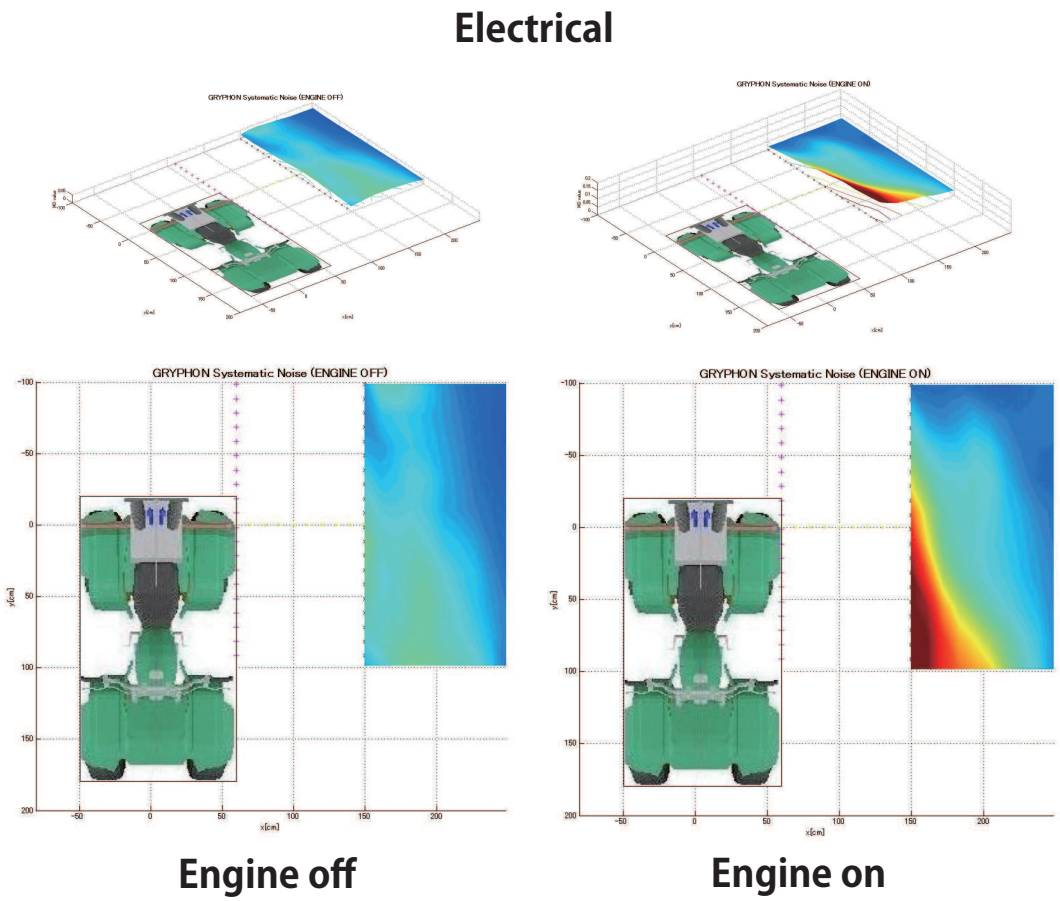


**Figure 5.20:** *Noise caused by electrical devices*

### 5.4.2 Experiment Methodology

The discrimination method presented searches the closest data in the database outputting a target and depth information. 362 data is used, which each one is removed from the database and used as input so that the original data is not in the database during the search. We input some different levels of White Gaussian Noises to simulate mechanical noises, keeping all other conditions unchanged. When inputing this noise (Fig. 5.21), the control parameter is the standard deviation "ST", which higher values represent high errors and ST =

0.01, 0.05 and 0.1 were adopted in this paper.

### 5.4.3 Depth Estimation Analysis

According to the closest data obtained from database search, we have depth estimation as shown in Table 5.6. From the results, one can observe that the depth estimation errors tend to increase as the ST values increase. The average depth error for all tested ST values kept small (sufficiently accurate for supporting demining operations), but the maximum error greatly increased (Fig. 5.22). For ST=0.01 and ST=0.05, the maximum error are 84 and 45 mm, respectively. For ST=0.01 it happened for only 1/261 valid data while others kept bellow 45 mm (maximum error with no noise case). For ST=0.05, the maximum error kept similar to no noise case, happening for 3/261 targets, but can be still considered good level for demining operations. For ST=0.1, 8 data has errors higher than 45 mm, with maximum depth error 142 mm, what can be considered high caused by changes in the maximum signal amplitude, main parameter for depth estimation. However, the resulting Error is also high (19.3%) so that adopted safety margins consider the target a potential mine, i.e., there are no risks for the demining operation.

### 5.4.4 FAR Analysis

The variation in FAR according to ST is shown in Fig. 5.23. There is very little increase in FAR according to ST since the search in the database is slightly affected. For ST = 0, there are no False Negatives (FN), but 2 FN happen for ST = 0.05 and 1 for ST = 0.1 when $dE_{threshold} \leq 10\%$. In short, adopting a correct safety margin of $dE_{threshold} \geq 10\%$ is enough for generating no False Negatives.

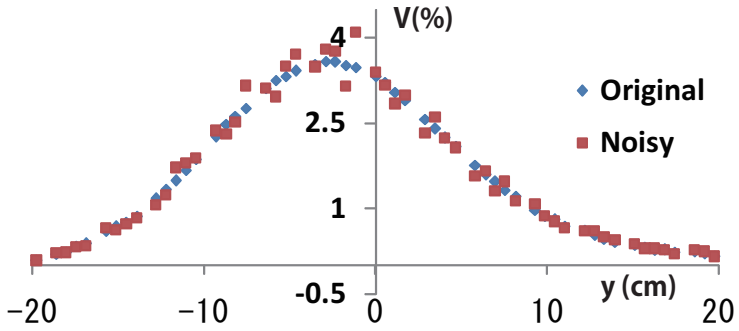


**Figure 5.21:** *Data with White Gaussian Noise*

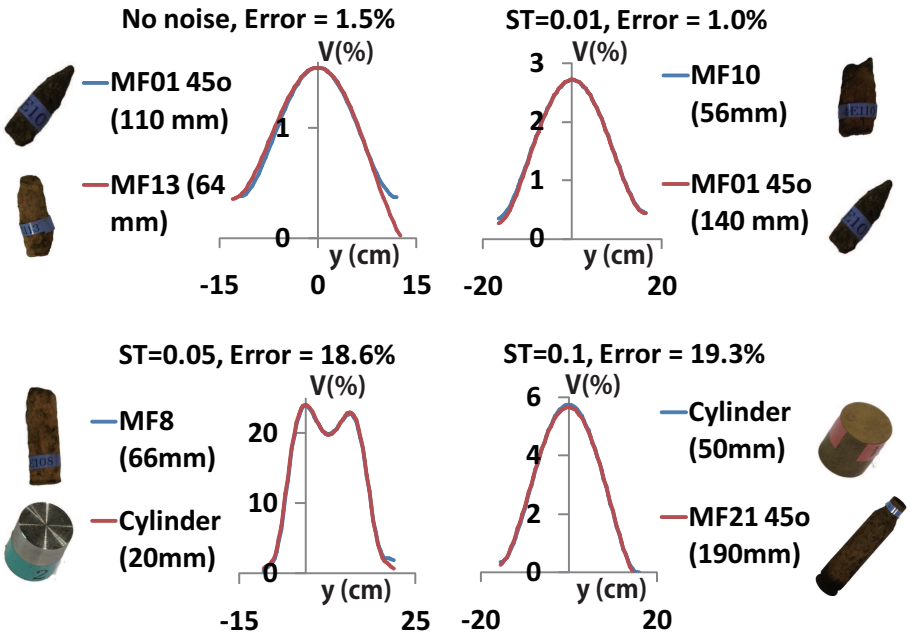


**Figure 5.22:** *Maximum depth error cases for each target, with depths in parenthesis. Blue lines represent noisy inputs and red ones the closest data in the database*

TABLE 5.6: DEPTH ESTIMATION RESULTS

| | Average Depth Error (mm) | Maximum Depth Error (mm) |
|---|---|---|
| **No noise** | **4** | **45** |
| **ST = 0.01** | **5.1** | **84** |
| **ST = 0.05** | **5.4** | **45** |
| **ST = 0.1** | **10.4** | **142** |

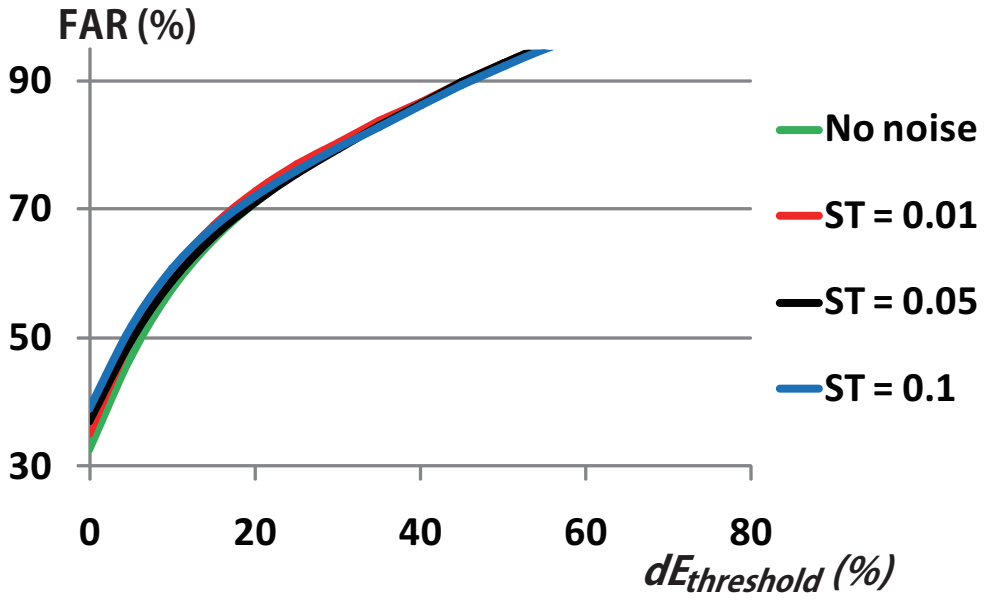


**Figure 5.23:** *Resulting FAR according to adopted ST levels*

# Chapter 6

# Further Enhancements in the System

This Chapter describes some other achievements, such as trajectory and scan line step optimization, new vision techniques and GPS data integration interface.

## 6.1 Trajectory and Scan Line Step Optimization

The current scanning procedure using Gryphon consists in swinging its arm sideways in lines (1,2,..., m), advancing in steps between one line and another. Small steps provide more information necessary for better discrimination, but it increases the total scanning time. Nevertheless, for actual manual demining operations, a scan line step of half of the MMD head's diameter is considered enough to assure no target is missed [35]. This chapter describes the minimum line step for the robot scan to not miss any target, and also analyze about the required line step to assure effective discrimination. Finally, a new scanning trajectory is proposed, which take in consideration both line step criteria and greatly decrease the total operation time. Further details of this chapter can be found in [28].

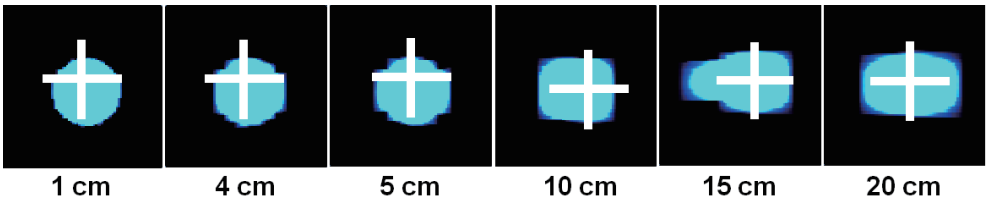


**Figure 6.1:** *Auto-detection results of ITOP target at 3 cm depth, for various line steps*

### 6.1.1 Scan Line Analysis for Targets Detection

As mentioned above, for manual scanning, lines step of 1/2 (10 cm) of the MMD head's diameter would assure no missed targets. For the robotic scanning, a line step of 1/5 (4 cm) has being adopted so far considering a large safe margin. Other reason for adopting small steps is explained by the necessity of good imaging of the target signal, because the displayed image was to be used by a human operator who would identify and mark the target on a PC screen. However, the authors have been developing new automatic target detection methods [25], which consider peaks and sizes/area of the signals, and for this case good imaging is not essential.

Fig. 6.1 illustrates one of targets imaging and auto-detection results for different scan line steps. The cross marks indicate a detected target and its maximum MMD signal location. The results show that although images get distorted for higher line steps they still can be recognized as targets, and the signal peak maximum variation was about 6.4% between 20 cm and 1cm line step. For line steps higher than 15 cm, the error increases abruptly (Fig. 6.2), which confirms that a line step of 1/2 of the MMD head's diameter is also the limit for robot manipulation as well. Other targets with similar size of the ITOP but made of different materials (aluminum, brass and stainless) and variable depths were used for a similar analysis. Fig. 6.2 shows the average maximum decay for each step of 36 analyzed data, divided by $MMD_{max}$ ranges. The smaller the $MMD_{max}$, the higher the $MMD_{max}$ decay, but for all cases there is an abrupt increase after 15 cm. Even though 15cm is the resulting limit in this analysis, for safety 10cm will be adopted as suggested by the MMD maker.

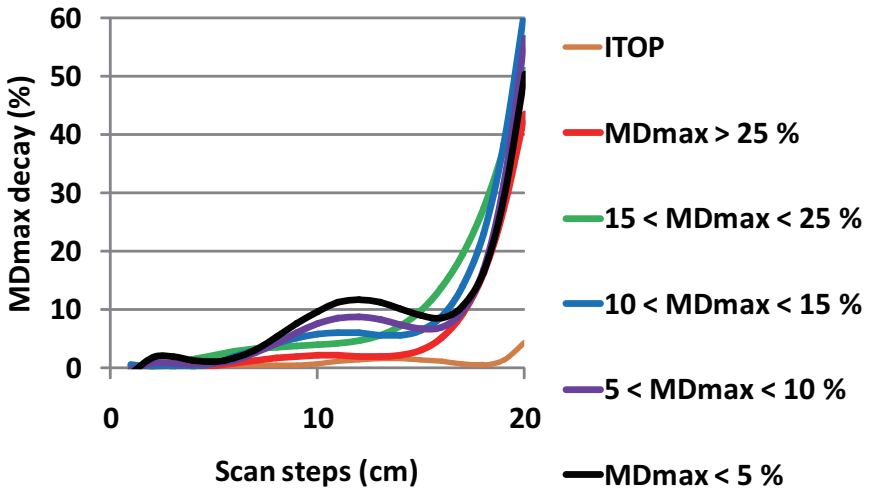


**Figure 6.2:** *Average maximum signal decay in function of scan steps for different* $MMD_{max}$ *ranges*

### 6.1.2 Scan Line Step Analysis for Discrimination

In this section, the minimum line step that ensures discrimination using the Curve Characterization Method is analyzed. For this analysis, several targets were scanned at two positions: 1) targets longest length on the same direction as the scan lines ($0^o$) having the maximum amount of information and 2) targets longest length at $135^o$ in relation to the scan lines. The analysis consists in comparing the extracted signals on both directions and for different scan line steps.

For the scan line step, one target of each type of signal profile (Fig. 6.3) was scanned several times on both $0^o$ and $135^o$ positions, which main axis information was extracted and transformed in polynomial. The procedure was repeated for different scan steps and the average values of each case were calculated. The comparison criterion, the error between two curves, is adopted as eq. 5.2 in Chapter 5.

The obtained errors with corresponding line step are shown in Fig. 6.3. For Profiles 2 and 3 the errors are small until step=4cm, rapidly increasing after this value. For Profiles 1 and 4 the errors start increasing from step = 3cm, what

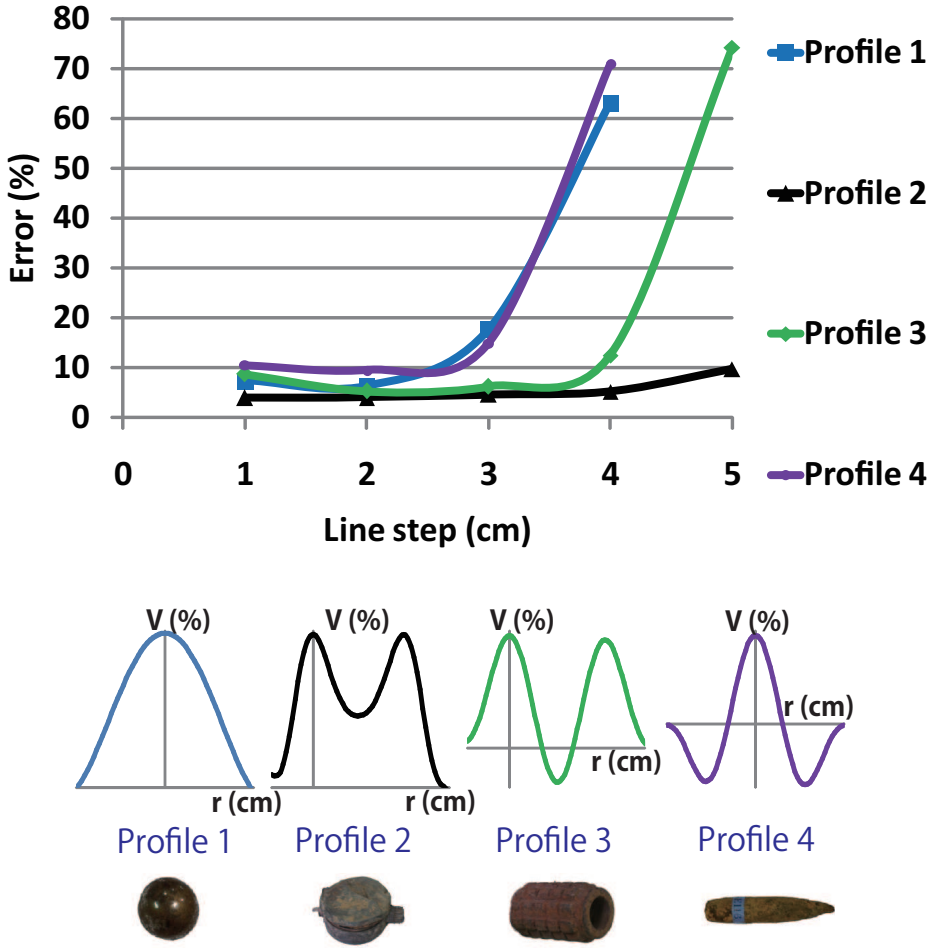


**Figure 6.3:** *Obtained errors between targets in 0 and* $135^o$ *for each scan line step*

shows the maximum scan line step is 3 cm. Even though the error in Profile 4 is relatively high (around 10%), it can still be matched with the information in the database, for its Profile being less frequent than the other ones.

### 6.1.3 New Scan Procedure

Taking in consideration results from previous sections, we introduce a new scan procedure, which consists in:

- Step 1) Scan the area of interest at 10 cm line step
- Step 2) Detect the target center position [25]

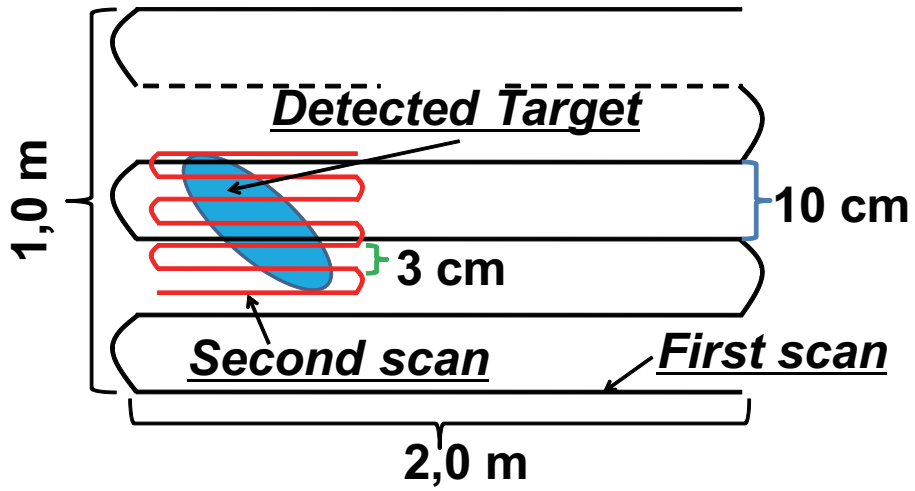


**Figure 6.4:** *Proposed new scan procedure*

- Step 3) Scan around the detect targets at 3 cm line step
- Step 4) Make discrimination of mine/metal fragment

An example of scanning trajectory is shown in Fig. 6.4. With the new proposed scan, the demining process with Gryphon (map, scan, marking) is reduced, as in Fig. 6.5.

## 6.2 New Artificial Vision Technique for Extreme Sunlight Conditions

Gryphon scans areas moving mine sensors at constant distances from the ground. Since the ground is almost never flat, nor no assumption can be made a priori of it's shape, it becomes necessary to use a system able to perform 3D acquisition of the scene, like stereoscopic vision. Considering robustness to various field conditions, cost, precision and processing speed, a BumbleBee stereo vision camera [42] (model no. BB-HICOL-60, Point Grey Research, Vancouver, Canada) was selected. However, when Gryphon operated in the field, an unforeseen problem arose: for some lighting conditions, the stereo correspondence algorithm was unable to find enough features to perform a depth analy-

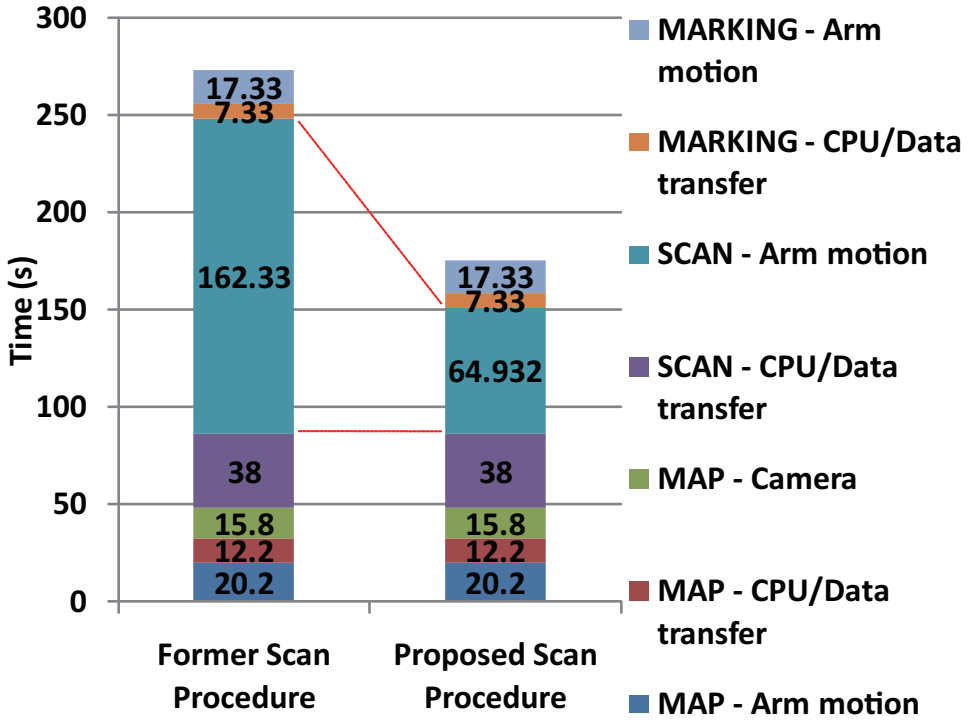


**Figure 6.5:** *Demining time with Gryphon with the former and new scan procedures*

sis. The problem was caused by the strong sunlight, which saturated some areas of the image while keeping other areas of the image visibly under-exposed. Even the shadow of the robot itself was enough to create such depth map calculation problems. Once the map can't be correctly grabbed, the robot tries to fill these empty regions with simple planes. In other words, if variations in the terrain are not correctly detected, the manipulator will be wrongly moved and even with the possibility of hitting an obstacle.

The problem was reproduced in laboratory (Fig. 6.6) and reported in [29]. The extreme lighting environment simulated direct sunlight by using a high power directional light. One can observe that the depth map calculated from the stereo pair lacks so many features that only a small fraction of it was actually used to compute the 3D information. The limitation comes from the camera's

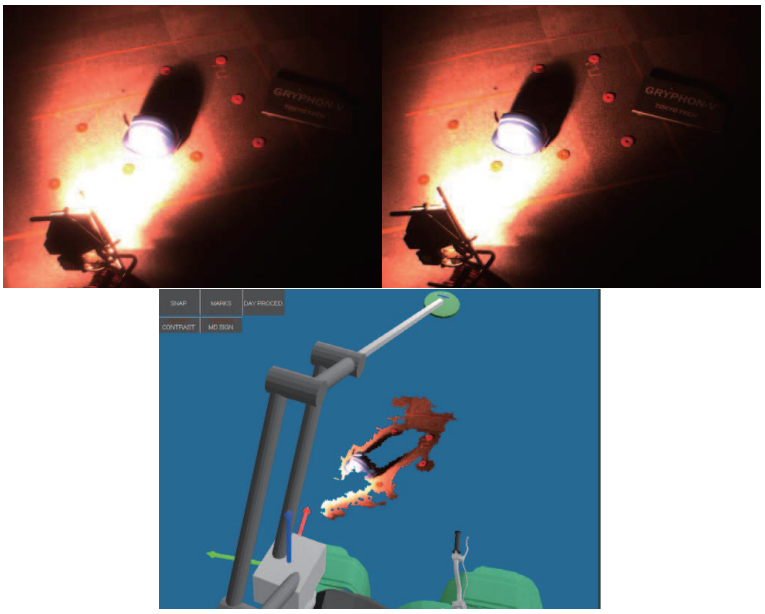

**Figure 6.6:** *Effect of strong light contrast in the map calculation. Camera's left-right pair images (top) and calculated depth map displayed in the user interface (bottom)*

dynamic range, which represents the limits of luminance range that a given device can capture. In the case of Gryphon, automatic exposure control is used for capturing images but Fig. 6.6 shows that automatic exposure control can not be perfect, since it doesn't know which part of the scene to illuminate properly and simple average adjustments are used. The proposed solution for this problem, experiments done in four different conditions, the results and analysis are presented in the next sections

### 6.2.1 Implemented Solution

Several different techniques exist to solve this kind of iluminance problem, which two will be specially detailed here: the High Dynamic Range (HDR) and the Exposure Fusion (EF). In many vision-related fields, the HDR imaging is intended to allow a greater dynamic range of luminance levels between light and dark areas of a scene. It tries to accurately represent the ultra wide range of intensity levels found in the real world, ranging from direct sunlight to shadows.

The nowadays accepted as standard is the one presented by [8], presented to the computer graphics community and widely used. The presented technique basically consists in photographing the same scene many times with a wide range of exposure settings and then combining those separate exposures into the final image. Despite its overall good results, it is mathematically complex and difficult to be implemented in real time.

EF, by its turn, requires less computing power than HDR. It was originally presented in [33] and its purpose is to skip the step of computing a high dynamic range image in order to immediately fuse the images. It computes a perceptual quality measure for each pixel in the multi-exposure sequence, encoding desirable qualities like saturation and contrast, selecting "good pixels" in each image to form the final one. As HDR, this technique also requires perfect alignment of the input images.

Even though most of the works based on HDR and EF are for photography purposes, there are some works which these techniques were applied to enhance real time stereo vision [23] [24] [19]. For photography, the goal is to obtain a scene with superior quality than the one with single exposure, while for stereo vision, the goal is to obtain features for the stereo correspondence algorithms also caring about optimizations for real time operations. The final implemented algorithm aims at reaching a good level of fusion, taking HDR as the maximum possible qualitative achievement, but following an approach more similar to EF. The algorithm computes the average image by considering weights when balancing the pictures, but instead of basing in continuous values, the weights in this project works in discrete logical fashion. It bases on the average brightness levels of each image and in case it exceeds a certain threshold, the weights are set to zero. The threshold level was statically set with a training procedure, taking several pictures with the stereo camera in many possible light conditions, with the aim of setting a 1.0 weight only for images which presented at least 20% of valid information with respect to the total data. Differently to the classical EF, in this implemented algorithm only luminance information is used, speeding-up the overall computation time. The number of exposures was chosen according to the range of the camera, which ranges from EV 0 to EV 1000. EV 200 step is enough for avoiding information loss, resulting

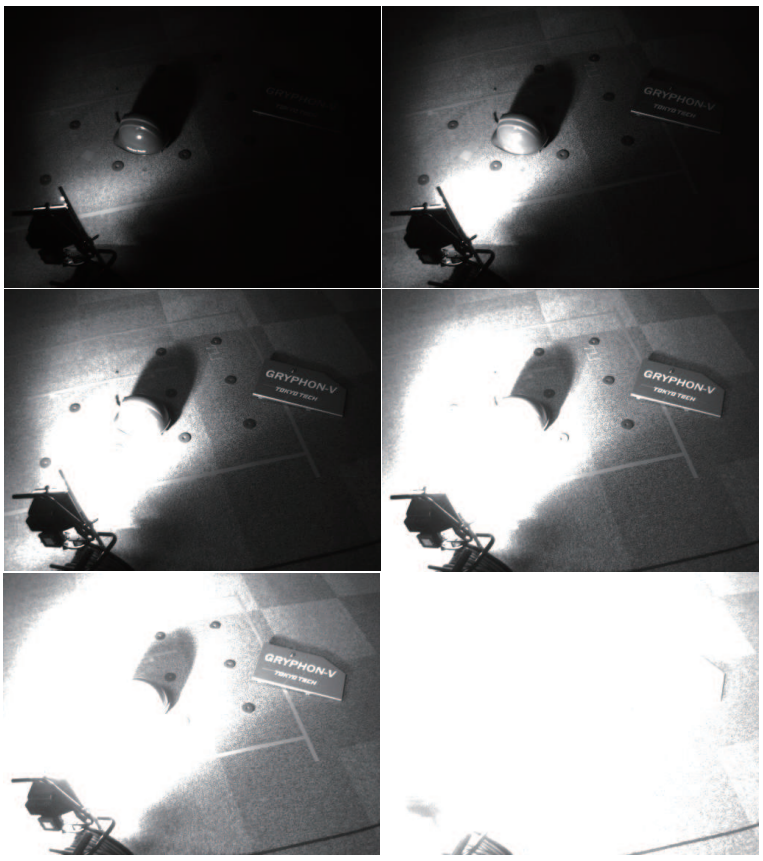


**Figure 6.7:** *Six images with different exposures used for computing the compensated image in laboratory environment*

in a total of six steps (six images with different exposures, Fig. 6.7), which EV 0 is also taken into account. In mathematical terms, the output image is computed in a similar way to EF, as shown in eq. 6.1:

$$O_{ij} = \sum_{k=1}^{v} W_k I_{ij,k} \tag{6.1}$$

*where O = output image*

*I = input image*

*W = dynamic weight*

*ij = pixels*

*k = input image index*

The algorithm was implemented and tested in similar conditions as showed in Fig. 6.6. As can be seen in Fig. 6.8, the improvements in the compensated image are evident. The overall contrast is on average lower if compared to the HDR and EF techniques, but this lack in vivacity of colors is not important for stereo vision, since the quality and total number of visible features are improved. From Fig. 6.8, one can also observe more features in the calculated depth map displayed in the user interface, thus meaning more information for a precise terrain scan by landmine sensors. To quantify the increase in features between the two images, Speed Up Robust Features (SURF) [3] was applied and results are shown in Tab. 6.1. Further detailed information of the proposed algorithm can be found in [29].

### 6.2.2 Natural Sunlight Experiments

The proposed solution showed improvements in laboratory environment, but conditions more similar to real operations, i. e., natural sunlight, must be analyzed and are presented in this section. All the presented data in this paper was obtained during summer season in Tokyo area of Japan, between 11h and 14h, with average temperatures from 30 to $35^o$ C on no cloudy days. The experiments consist in repeating the procedures done in laboratory, i. e., measuring with SURF the number of features of both non-compensated and compensated images of many different conditions (called here as “Condition A”, “Condition B”, “Condition C” and “Condition D”) with lighting contrasts and also analyzing the final computed depth map. For quantifying the enhancement of the depth maps, valid pixels for each compensated and non-compensated case were computed, which more pixels represent more information captured by the camera and more area for reliable landmine detection. All depth maps were normalized for comparison and all pixels containing the color white (representing holes in the depth map) were considered “non-valid” pixels. The Conditions were selected with the intention of testing the algorithm in as many different conditions as possible, even if they don’t directly represent most of the real minefield conditions. However, Condition D was chosen for being the closest condition to real minefields, so it will be particularly deeply explained. For

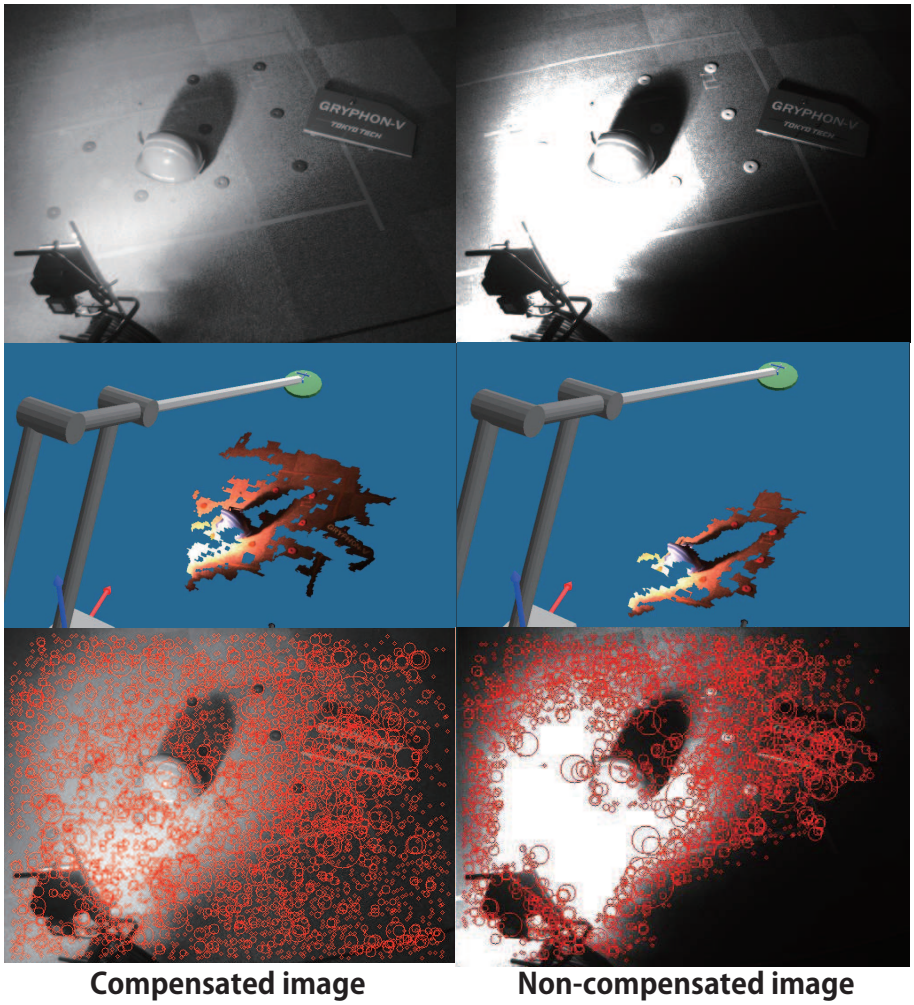


**Figure 6.8:** *Implemented algorithm: compensated (left top) and non-compensated image (right top). Resulting depth map displayed in the user interface (middle images) and detected features for each case (bottom)*

deeper analysis, a digital illuminance meter (maker Tondaj, model LX-1010B, 5.0% accuracy and 1 lux resolution) was used and corresponding values will be presented. All data obtained in the experiments are summarized in Tab. 6.2, detailed in the next sections.

a) Condition A: Asphalt

Fig. 6.9 shows the first location for experiments: an asphalt. The top images taken with the camera are the cases which no contrast exist. Even though

there is some shadow from the robot's arm, a wider shadow area was artificially created in this work, resulting in contrast areas (bottom images) for analysis. As shown in Tab. 6.2, the average measured illuminances are 97500 lux for the bright and 9900 lux for dark (shadow) regions. When there is no contrast, the measured number of features for the non-compensated image is 5795 and 5606 for the compensated one, resulting in an decrease of 3.3%. This decrease is small and observing the detected features with SURF and the calculated depth map, no significant loss of information occurred. In fact, analysing the number of valid pixels in each depth map, there is an increase of 0.4% from the depth map calculated from the non-compensated image to the compensated one. Here, one can observe that even though SURF can correctly detect features in images, it does not directly represent the quality of the resulting depth maps. This is due to the particular algorithm used for stereo correspondence by the camera drivers. Even if the algorithm is using some kind of feature detection method similar to SURF, it can be stated that the feature extraction implemented by the camera maker behaves differently from SURF correspondences.

When a contrast is created, the number of features are 4372 and 4772 for the non-compensated and compensated cases respectively, generating 9.1% increase in features. It can be said a good improvement specially considering the computed depth map, which great part of the missing areas in the non-compensated image was correctly computed in the compensated case, resulting in 29% pixels increase.

b) Condition B: Low Vegetation

Condition B is composed of some low vegetation and leaves (Fig. 6.10). The average illuminance is 93000 lux for bright regions and 12900 for dark ones. The number of features in the no contrast case is 5395 for the non-compensated and 5311 for the compensated case, this results in a decrease of 1.6%. Even when shadows are introduced, the average increase keeps in 0.0% (from 5193 to 5191). Observing the calculated depth map for both non-compensated and compensated images for both contrast and no contrast cases, one can observe a similar overall result in the depth map (increases of 3.6% and -3.1%, meaning no significant change in their quality). Even though in this example SURF was

able to detect features even in contrast regions (due to the diversity of shape and colors in the image) the depth map presented poor information in similar areas (mostly in dark ones), due to the stereo correspondence by the camera drivers already explained in Subsection A.

c) Condition C: Middle Size Vegetation

Fig. 6.11 shows Condition C, formed of middle sized vegetations. The measured brightness is 82000 lux in bright regions and 10300 in dark ones. The results in Tab. 6.2 shows similar behavior as the previous Conditions, with a decrease in the number of features in the no contrast case (from 5460 to 5274, decrease of 3.4%). The calculated depth maps were also similar in terms of pixels (subtle increase of 1.9%), with a good detection with SURF but which ended generating some empty regions. On the other hand, a 7.7% increase in features (from 4908 to 5284) is achieved in the case with contrast. The depth map computed with the non-compensated image has many features missing (almost one quarter of the whole image) and this lack of information is reduced using the compensated image, resulting in 5.6% more pixels.

d) Condition D: Soil

Condition D is the last and most important observed place, for it has similar conditions as minefields: soil (Fig. 6.12). The average measured brightness is 79500 lux in bright and 9500 in dark areas. When no contrast exist, there is a 5.6% decrease in the number of features for the non-compensated image, from 4425 to 4177 features. From the images, one can observe that soils can contain very featureless regions, difficult to be detect, as shown with SURF in the bottom parts of the images, and that result in empty areas in the depth maps. The calculate deph maps for each compensated and non-compensated case remains unchanged, with as small decrease of 4.5% in pixels.

When contrast is introduced, there is a small 4.7% increae in features, from 3307 (non-compensated image) to 3463 (compensated image). However, a great improvement in the depth map can be observed. While for the non-compensated image almost no information was correctly captured in the dark regions, in the compensated image it was almost completely detected, representing 18.5% in-

crease in the whole image. For further analysis, the six images used for computing the compensated one are displayed in Fig. 6.13. One can observe that different features arise as the exposure is changed. For some minor parts of the images there is still little or no significant information, what shows that even though the algorithm is performing correctly limitations in the camera still exist. Finally, after the depth map is correctly calculated by the proposed algorithm, the scan can be performed. Fig. 6.14 shows that when the non-compensated image is used, there is not enough information in the dark areas for computing the manipulator trajectory. Even though the robot tries to compensated featureless regions with planes as mentioned in the former sections, the depth map is still too poor for permiting a scan and an error message is displayed. On the other hand, with the compensated image the scan on the same area can be done normally, which MMD trajectory and detected signals [25] can be visualized (top-rigth image in Fig. 6.14).

e) Results Discussion

The increases in features and depth map quality for all Conditions are summarized in Fig. 6.15. For all cases, when there was contrast, there was an increase of ±5% in both number of features detected by SURF algorithm and depth map quality when comparing non-compensated and compensated images, representing insignificant loss or gain of information. In other words, when there is no contrast, the algorithm doesn‘t influence on the results, as also observed in laboratory environment.

However, greater improvements can be observed in contrast cases. While for Conditions B and C no significant changes were obtained, Conditions A and D resulted in high increase of features and depth map quality. Condition A presented the highest increase (29%) in pixels and features (9.1%) from non-compensated to compensated cases. Condition D, the closest to minefield conditions, also presented high increase (19.5%) in the depth map quality, permitting correct positioning of landmine sensors (Fig. 6.14).

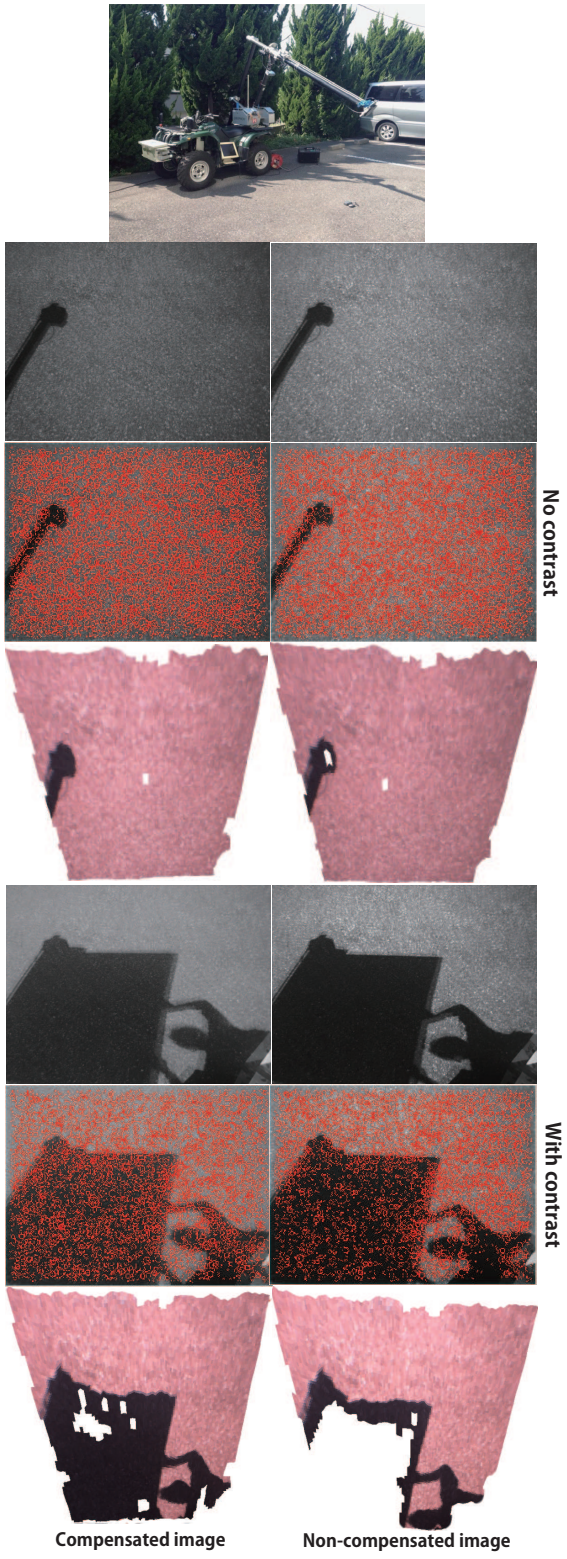


**Figure 6.9:** *Condition A: asphalt. No contrast images, detected features with SURF and calculated depth map (top); contrast images, detected features with SURF and calculated depth map shown in the user interface (bottom)*

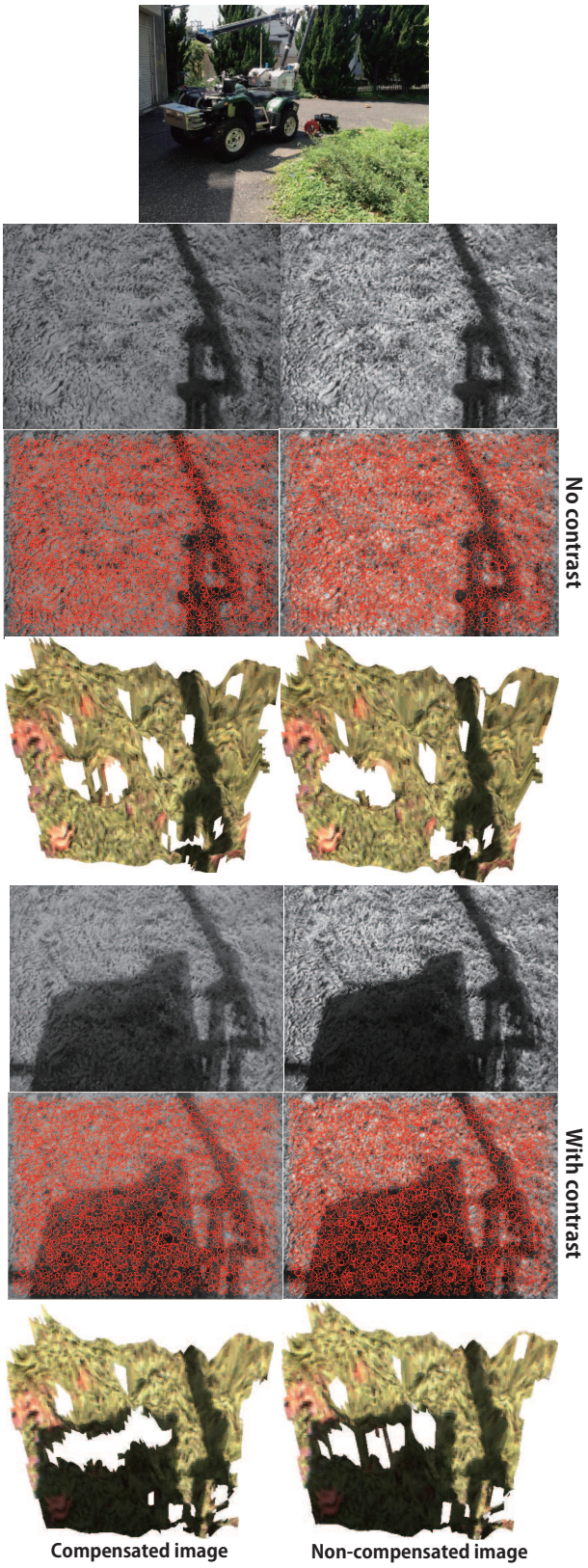


**Figure 6.10:** *Condition B: low vegetation with fallen leaves. No contrast images, detected features with SURF and calculated depth map (top); contrast images, detected features with SURF and calculated depth map shown in the user interface (bottom)*

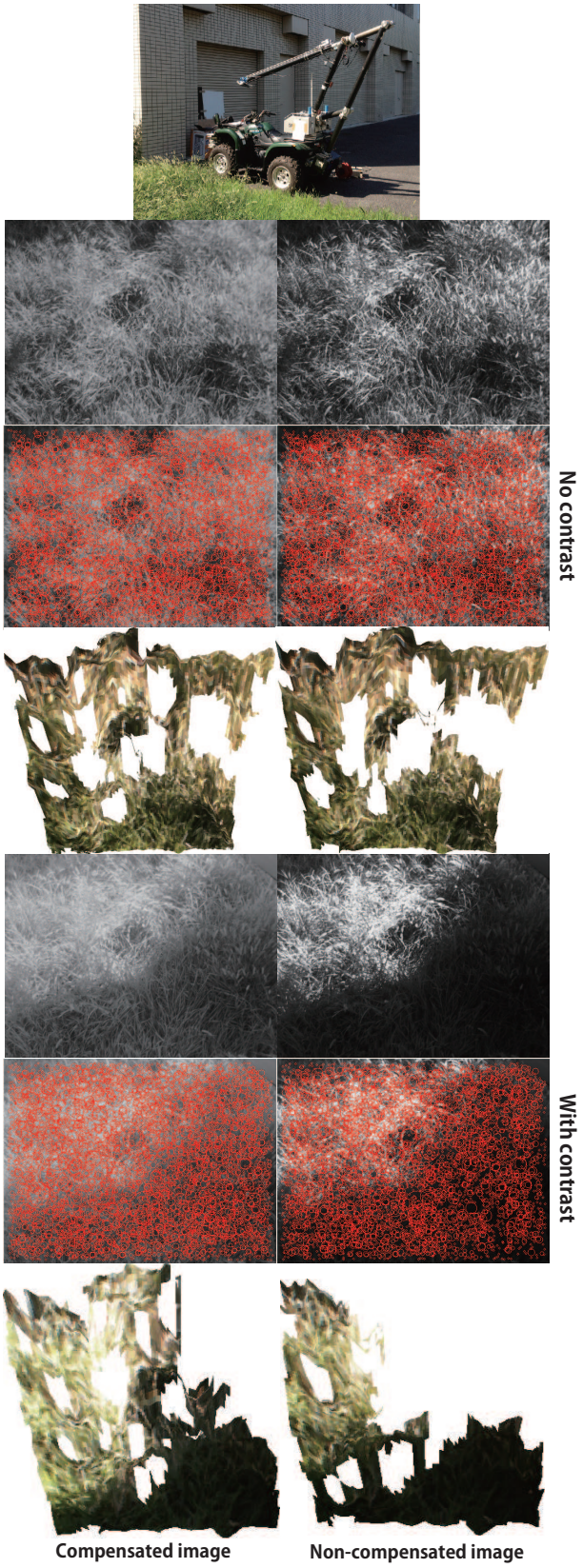


**Figure 6.11:** *Condition C: middle size vegetation. No contrast images, detected features with SURF and calculated depth map (top); contrast images, detected features with SURF and calculated depth map shown in the user interface (bottom)*

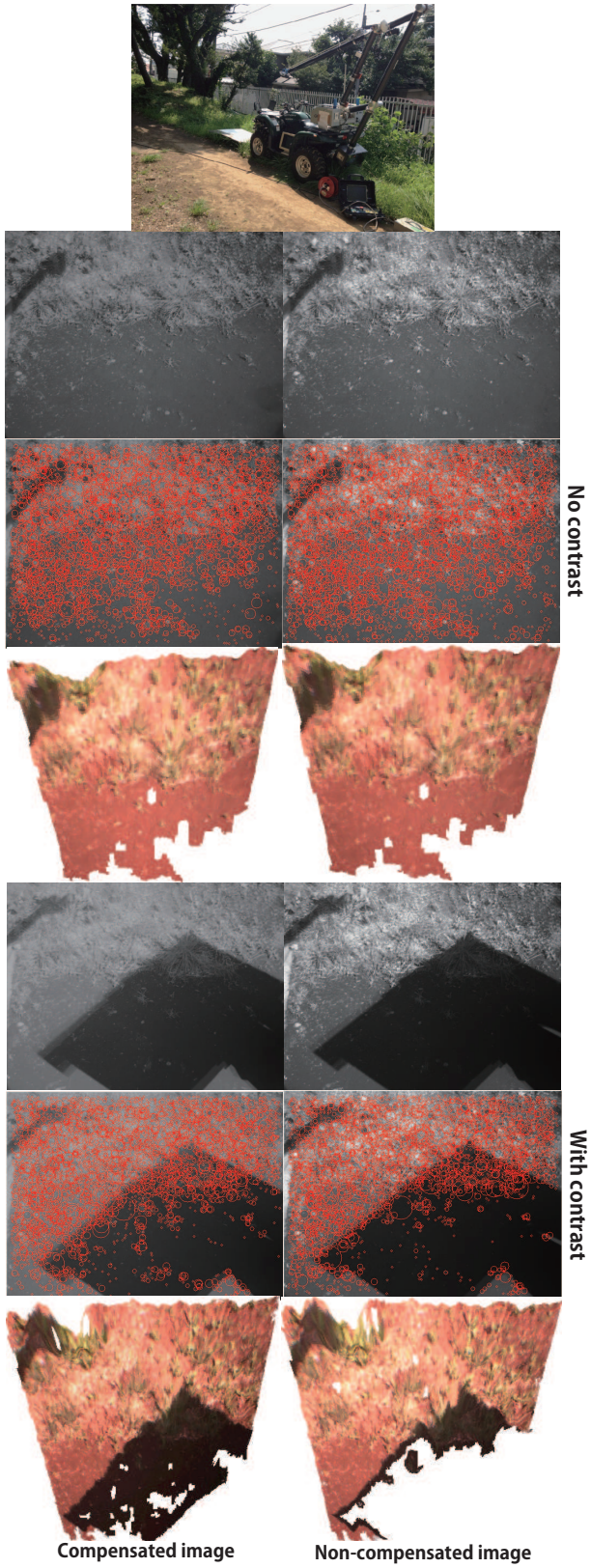


**Figure 6.12:** *Condition D: soil. No contrast images, detected features with SURF and calculated depth map (top); contrast images, detected features with SURF and calculated depth map shown in the user interface (bottom)*

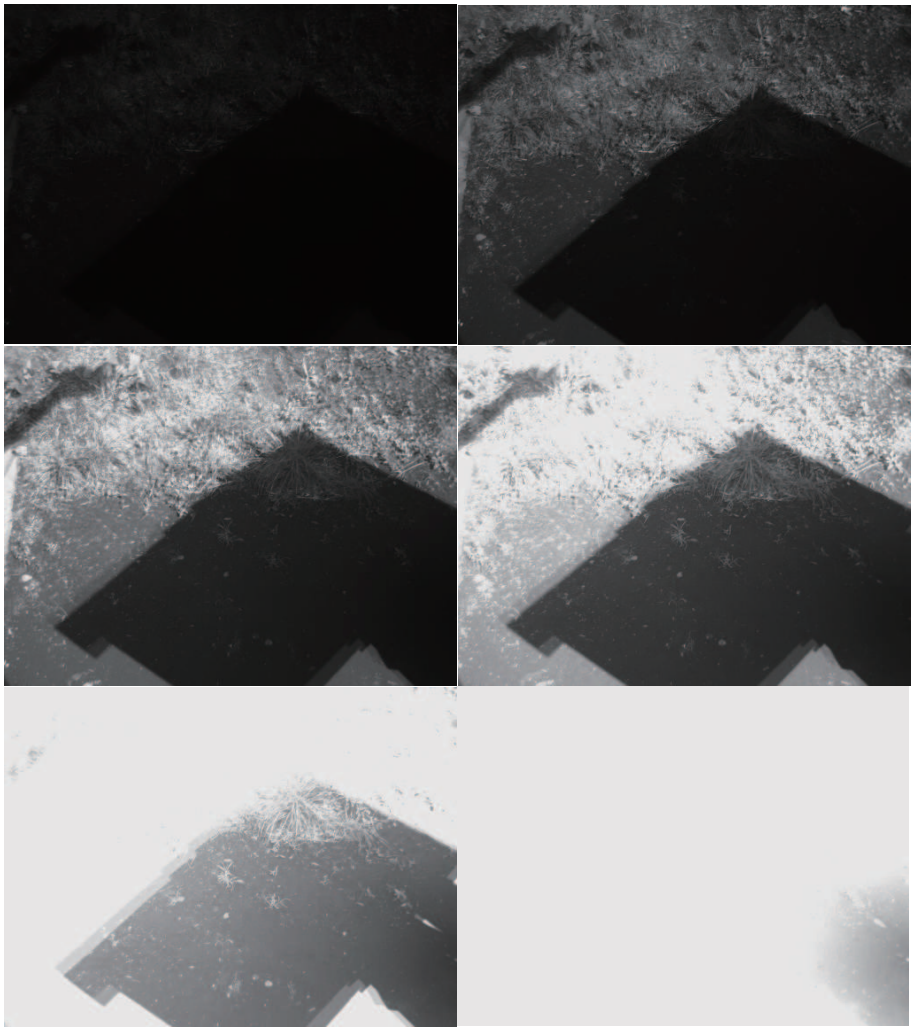

**Figure 6.13:** *Six images with different exposures used for computing the compensated image in outdoors experiments*

TABLE 6.1: NUMBER OF FEATURES DETECTED IN LABORATORY EXPERIMENTS

| Non-compensated | Compensated | Increase |
|---|---|---|
| **2493** | **3920** | **57.24%** |

## 6.3 GPS Signal and Map Integration

Chapter 2 introduced the GPS used with Gryphon for marking and navigating along the minefield. These information were only saved and stored into

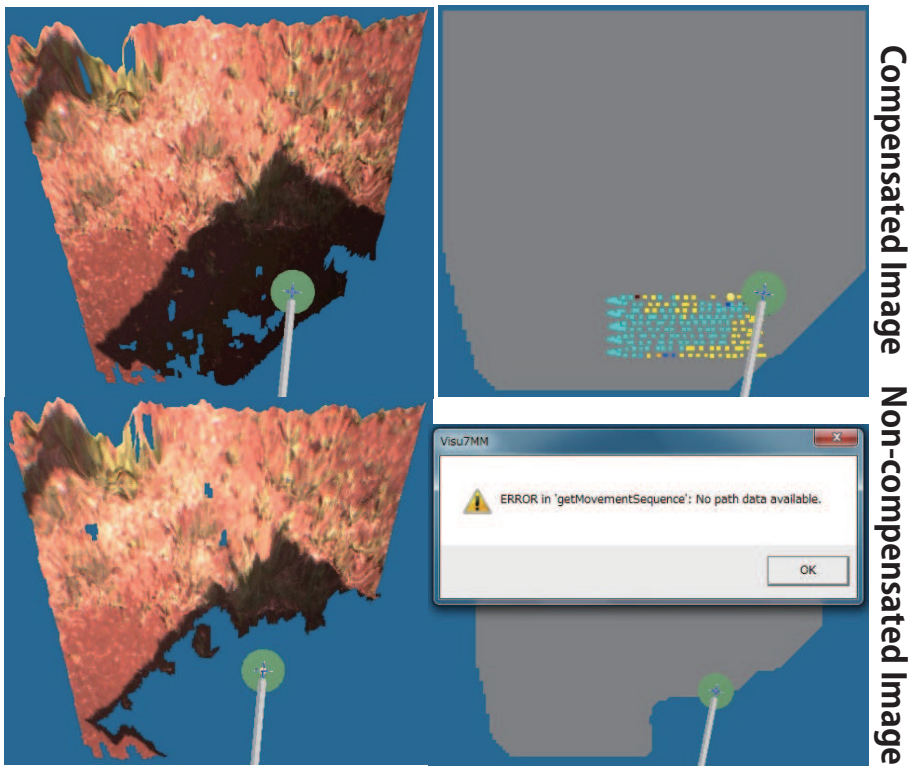


**Figure 6.14:** *Scan failure with the non-compensated image (bottom) and scan successfully performed with the compensated image (top). Depth maps detected by the camera (left) and calculated (including artificially filled empty spots) possible scan regions (right)*

files locally, but integration with external databases permit easy share of information. An interface (Fig. 6.16) for capturing. displaying and storing GPS data was developed in previous work.

The interface has two operation modes: read and record. In “record” mode, the interface basically receives the GPS output in National Marine Electronics Association (NMEA) protocol and displays basic information like position (in geographic coordinates), altitude, time, etc. During this mode, the operator can view in real time the grabbed points (clicking the button “Current Point”). In “read” mode, the interface recovers and displays the recorded data.

Using the mouse, the operator can at any time shift, rotate and zoom into the virtual map. For avoiding the operator to miss the starting point, the X, Y and Z coordinates are always displayed, as well as the angles in which the map

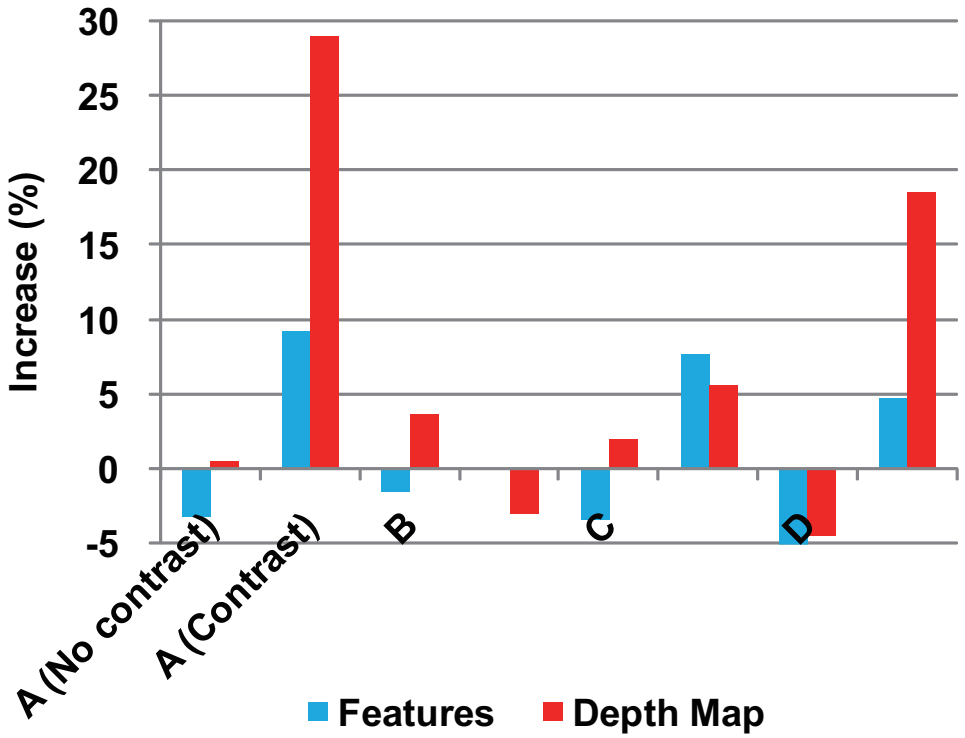


**Figure 6.15:** *Experiments results. Increases in features (blue) and depth maps (red) from non-compensated to compensated images*

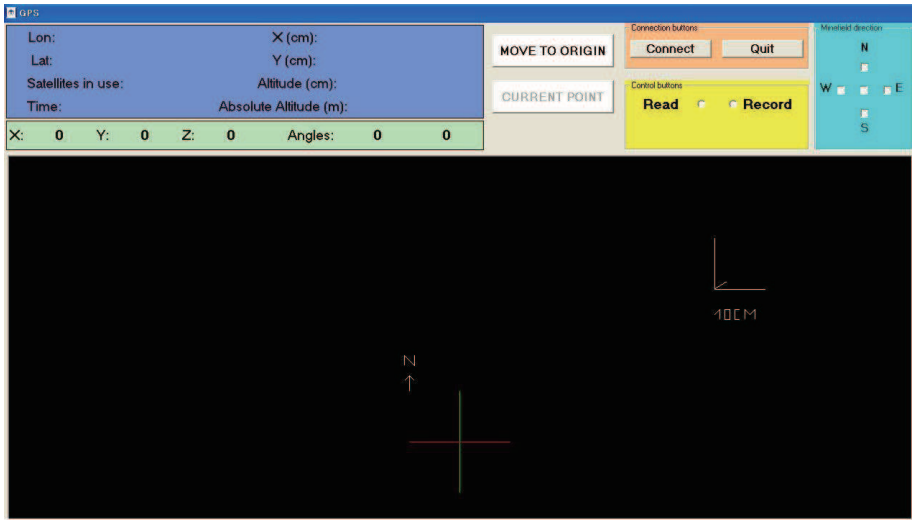


**Figure 6.16:** *Implemented interface for capturing GPS signals*

TABLE 6.2: NATURAL SUNLIGHT EXPERIMENTS AVERAGE RESULTS

| Condition | A | | B | | C | | D | |
|---|---|---|---|---|---|---|---|---|
| Description | Asphalt | | Low vegetation | | Middle size vegetation | | Soil | |
| Illuminance bright regions (x100) | 975 | | 930 | | 820 | | 795 | |
| Illuminance dark regions (x100) | 99 | | 129 | | 103 | | 95 | |
| Contrast | No | Yes | No | Yes | No | Yes | No | Yes |
| No. features non-comp. image | 5795 | 4372 | 5395 | 5193 | 5460 | 4908 | 4425 | 3307 |
| No. features Comp. image | 5606 | 4772 | 5311 | 5191 | 5274 | 5284 | 4177 | 3463 |
| No. features increase (%) | -3.3 | 9.1 | -1.6 | 0.0 | -3.4 | 7.7 | -5.6 | 4.7 |
| No. pixels non-comp. depth map | 67570 | 49992 | 59469 | 60178 | 53137 | 48258 | 66779 | 54737 |
| No. pixels comp. depth map | 67873 | 64473 | 61638 | 58339 | 54170 | 50963 | 63774 | 68846 |
| No. depth map pixels increase (%) | 0.4 | 29.0 | 3.6 | -3.1 | 1.9 | 5.6 | -4.5 | 18.5 |

was rotated. Moreover, the button “Move to Origin” rapidly shifts to the origin of the coordinate system.

The interface also shows a “N” symbol, which represents Earth’s geographic north, just for reference. Another features are the “N”, “E”, “S” and “W” symbols inside the “Mine direction” frame, which represents the direction in relation to the boundaries in which the minefield is located. After grabbing the boundaries, the operator checks the checkboxes and the interface stores this information for later use.


**Figure 6.17:** *GPS signal plotted in database by KML file*

### 6.3.1 Interface Enhancements for Database Integration

There are many different databases for inputting saved GPS data. Google Earth is a virtual globe, which provides map information by superimposition of satellite images and aerial photography. An interesting feature is the possibility of importing kml files, which structure is based on tags and XML pattern, for displaying GPS data in the globe for easy visualization (Fig. 6.17).

### 6.3.2 Interface Experiment

An experiment was done for verifying the validity of this approach. A GPS antenna was moved within a path in Izu Oshima, Japan. Fig. 6.18 shows the obtained GPS data and resulting plotted image in Google Earth. One can observe that the map was successfully plotted within the expected region. For further comparisons, a Hand GPS (model Edge 305, maker Garmin [16]), Fig. 6.19, was used for getting data in the same path and moment, which results are represented in Fig. 6.20. The signals obtained with both GPS proves the system is correctly capturing GPS signals and integrating with databases. Moreover, even though accuracy tests could not be done, results show that signals obtained with the Hand GPS are less robust, being more vulnerable to noises,

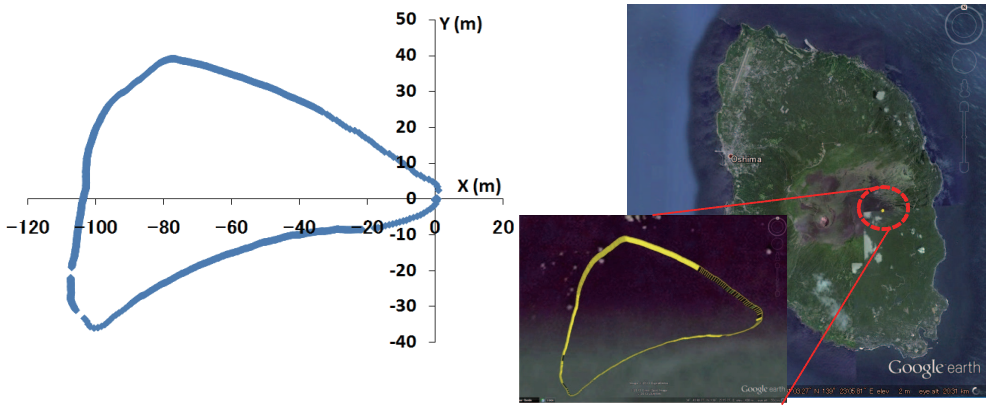


**Figure 6.18:** *GPS experiments done nearby Mt. Mihara in Izu Oshima, Japan*


**Figure 6.19:** *Hand GPS used for comparisons*

while signals from Gryphon's GPS are more stable. Fig. 6.20 displays errors in the order of 5 m, which are expected since Gryphon's GPS has cm order accuracy while the Hand GPS ±5.79 m (19 ft) [16].


**Figure 6.20:** *GPS data comparison between Gryphon' GPS (yellow) and a hand GPS (blue)*

# Chapter 7

# Conclusions

The primary objective of this thesis of proposing new methods for the landmine detection and marking system of the robot Gryphon was achieved.

First, filters for improving the landmine detection were proposed and experiments showed that both POD and FAR are improved by setting median as offset per each line of scan.

Second, the automated marking system for finding targets proved to be very effective in terms of time duration and reducing human factors, which guarantees constant results under any circumstances, differently from a human operator. Besides reaching POD and FAR rates as good as an experienced human deminer operator, the correct location of marks can be fast (3.3 $s/m^2$ seconds per data) and accurately found by the implemented algorithms.

The objective of proposing a new method for landmine discrimination was also achieved. The method can estimate depth of metallic targets using commercially available Mine Metal Detectors, based on a basic principle of comparison of signals/curves of targets obtained in the mine-field to a pre-build library containing data of many targets at different postures and depths. The concept of "Spatially Represented Metal Mine Detector Signal" was introduced to the signal obtained from robot manipulator scanning of the mine field and also proposed simplification of the SRMMDS to "Characteristic Curves" represented by polynomials, which lead to effective signal characterization and estimation in real-time. With this method, large amount of data can be eas-

ily stored in a database and quickly interpolated for representing continuous depth data from limited depth samples with high accuracy. The method can estimate depth with average error of 19 mm and maximum 41 mm, which are accurate enough for supporting the use of GPRs and also for permitting fast discrimination of AT landmines (usually located in deep areas) and anti-personnel landmines (usually buried in shallow areas). As a sub product, the method could also estimate targets materials; steel in 100% of the cases and other types of metals (aluminum, stainless, brass and chrome) in higher rates than the existing method. This is very important for quickly removing objects not made of steel, which is the material usually used in landmines.

Furthermore, as fundamental point in any discrimination method, possibility of FALSE NEGATIVE occurrences were overcome by labeling metal fragments that show similar characteristic curves of known landmines, as "potential mines". Moreover, we introduced a methodology to interpolate discrete data in the database according to its depth, making possible the evaluation of sensed data in arbitrary depths, which are originally not available in the prebuilt database. The occurrences of FALSE POSITIVEs, which increase FAR, depend on the adopted error margin criteria. After extensive laboratory tests, thresholds of $E_{threshold}$ (%) = 15% and $dE_{threshold}$ (%) = 5% were selected, which reduces the FAR to about 50%. Results from analysis of data obtained in a test field at Croatia in 2007, showed the robustness, validity and the potential of the proposed method for practical applications. A major advantage of this method is the time duration for estimation per target. While existing methods take minutes for estimating each target, the proposed method in this work takes only 1 s per target, which can be further reduced if parallel processing is applied.

This work also applied White Gaussian Noise for analyzing the influence of random noises in discrimination by SRMMDS. The method is robust for depth estimation for the analyzed standard deviations 0.01 and 0.05, while it presented higher errors for ST = 0.1, but can be compensated by safety margins. FAR increase as ST increases but adopting $dE_{threshold} \geq 10\%$ False Negatives are avoided.

Further enhancements in the scan trajectory, sequence and trajectory during the marking task (almost 50% time reduction, in the presented experiments)

and in the robot's vision were achieved. The analysis done showed that adopting 10 cm scan line step is enough for detecting targets with no misses, and a detailed scan with 3 cm step ensures reliable amount of information necessary for discrimination methods. With this results, a new trajectory was proposed, which can greatly reduce the total scan time for the Gryphon system.

From the experiments in natural sunlight in the four Conditions, the depth map calculated with the compensated image obtained with the proposed algorithm showed similar or better results comparing to the map with non-compensated image. When contrasts existed, the algorithm enhanced features of the images in until 9%, generating depth maps with until 29.0% more information and permitting correct landmine detection with mine sensors. It is possible to infeer that an efficient and on-site solution for the contrast problem caused by extreme sunlight exposure has been achieved and Gryphon has greater potential for real demining operations.

## 7.1 Future Work

Gryphon has proved to be a promising demining solution, but many improvements can still be done. First, the proposed discrimination method can be expanded for being able to discriminated multiple targets instead of a single one. One challenge is to predict how two signals of two targets sum to each other (Fig.7.1). The database implemented is able to successfully interpolate signals according to depth but interpolation according to other physical properties such as posture is also desired. As another sub product of the discrimination by curve characterization, a chart is obtained by plotting the norm of each polynomial (signal) in function to the maximum MMD signal (Fig.7.2). One target is connected to itself in different depths by lines. One can observe that each target and signal is located in different regions of the chart and similar signals stay very close to each other, what suggests that one can visually predict which target at which condition is close to another. This chart has great potential for quick discrimination and must be explored in details.

Another important improvement is currently being investigated and is related to a self-calibration method for Gryphon [31]. In order to compensate

errors caused by wrong initial position the stereo vision camera system is employed to make a self-calibration. The robot kinematic model parameters are being automatically self-calibrated on-site with a sequence of rotation of each robot link, which internal and/or external points in relation to the robot are detected by the camera and an external sensor (optotrak) and compared to the estimated position by kinematics by the robot (general scheme shown in Fig.7.3).

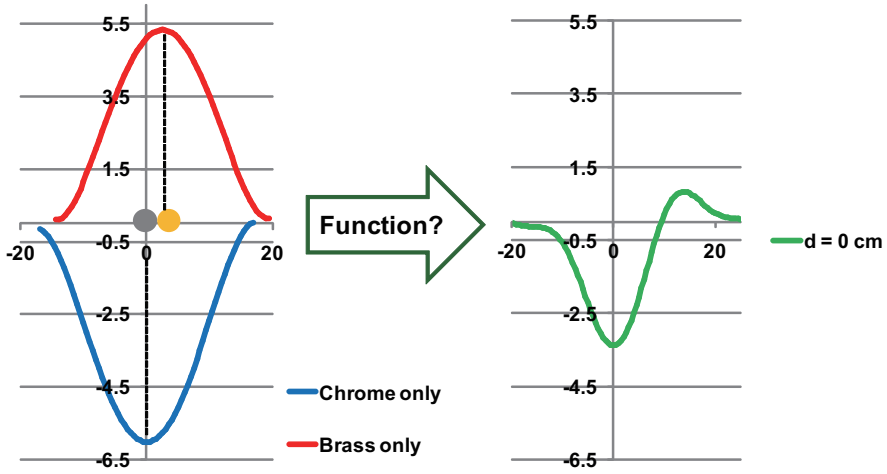


**Figure 7.1:** *Generated signal from two different targets scanned close to each other*

The control of the ATV should be improved to make the navigation through the minefield more autonomous. The ATV can perform basic movements such as accelerating, braking and steering, but advanced control such as path planning for efficient positioning is required. Moreover, also using the GPS, targets located in the overlapped regions between two areas must be joined accurately for correct discrimination with the here proposed method. A major problem that must be eliminated is the mentioned systematic noise caused by the engine's coil in Chap 5. This noise can mask any landmine in these areas being potential threats.

Finally, experiments in test fields in Angola are planned and more different targets must be input in the database for search. More real conditions such as soil and climate are very important factors to be included in the analysis.

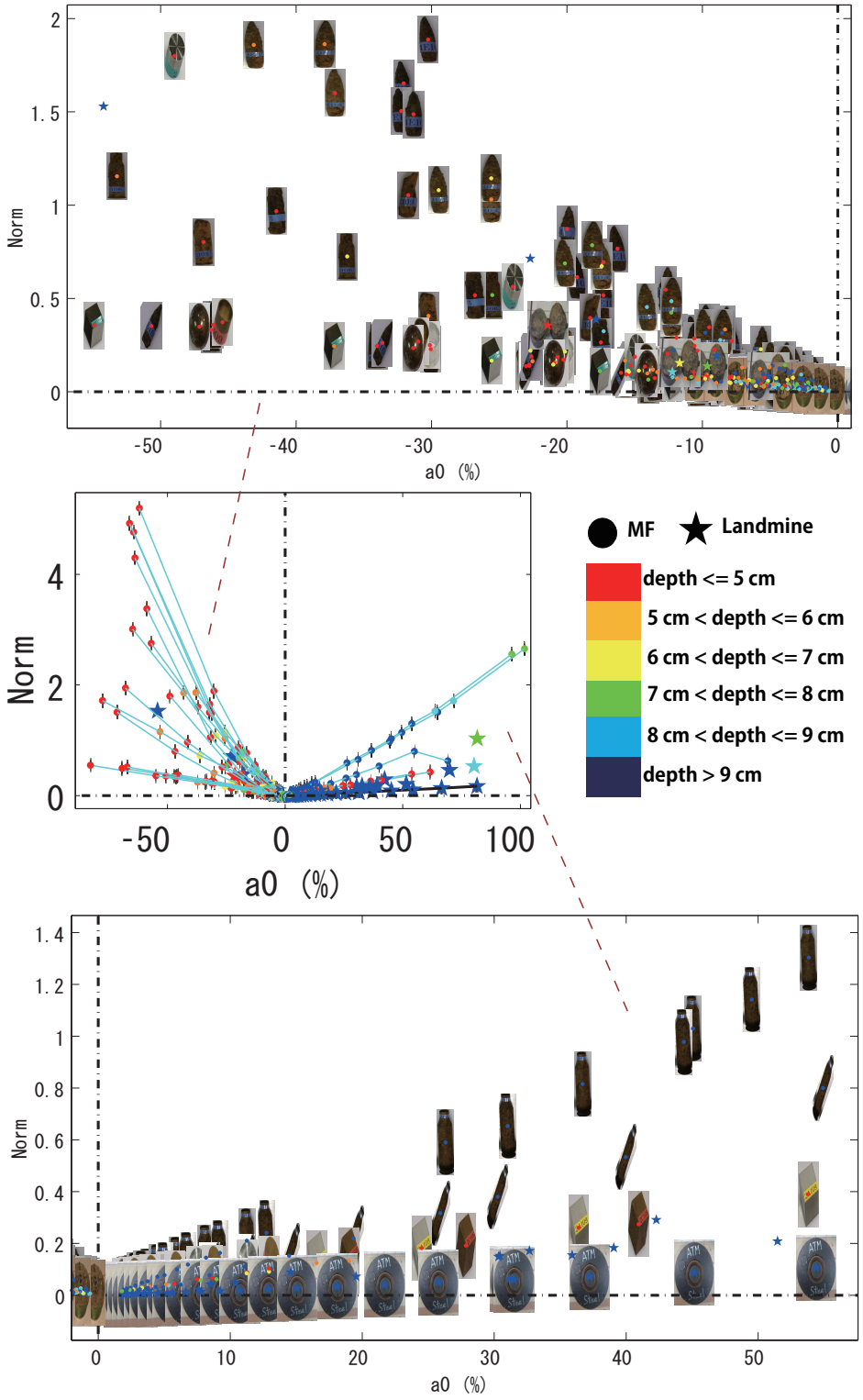


**Figure 7.2:** *Chart obtained as sub product of the proposed discrimination method showing all metal fragments and landmines regions. Signals with similar features (similar polynomials) are located very closely*

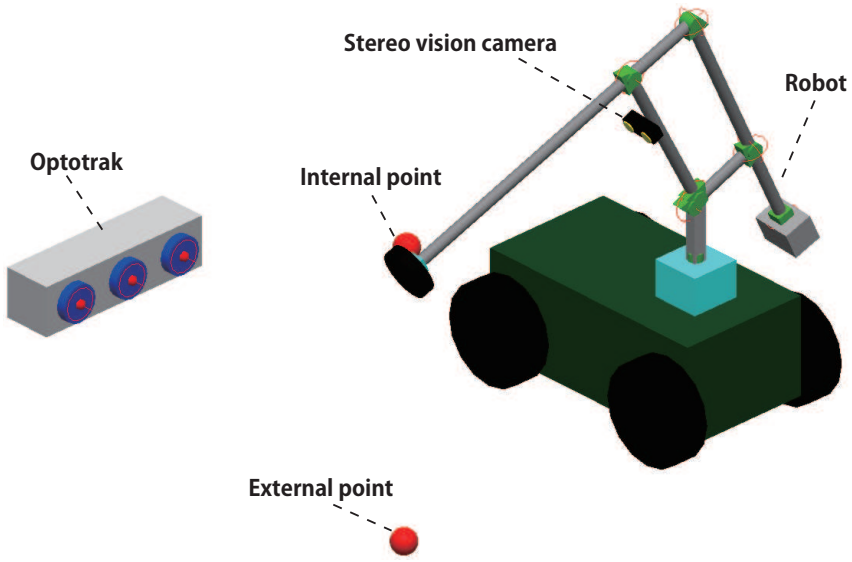


**Figure 7.3:** *General scheme of the self-calibration method: Internal/external points and sensors*

# Bibliography


[1] Astrachan O. Bubble sort: An Archeological Algorithmic Analysis. http://www.cs.duke.edu/ ola/bubble/bubble.pdf. 2003.

[2] Baumann J., Kaneko A. M. and Fukushima E. F. “GPS Based Autonomous Point-to-point Driving of All Terrain Vehicle Gryphon”. In proceedings of The 29th Annual Conference of the Robotics Society of Japan, Tokyo, Japan, September 2011.

[3] Bay H., Ess A., Tuytelaars T., Gool L. V. “SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU)”, Vol. 110, No. 3, pp. 346–359, 2008.

[4] Bicchi A., Casalino G., and Santilli C. Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles. J. Intell. Robot. Syst., vol. 16, pp. 387-405, 1996.

[5] Borgwardt C. "ODIS - Ordnance Detection and Identification System", WAPM’95 Workshop on Antipersonnel Mine Detection and Removal, Lausanne, June 1995, pp 37-43.

[6] CEN Workshop Agreement. “Humanitarian Mine Action - Test and Evaluation - Metal Detectors”, CWA14747, 2003. Available: http://www.itep.ws/pdf/CWA_metal_detectors.pdf.

[7] Collins L., Gao P., Schofield D., Moulton J. P., Makowsky L. C., Reidy D. M., Weaver R. C. “A Statistical Approach to Landmine Detection Using Broad-Band Electromagnetic Induction Data”, IEEETrans. Geosci. Remote Sensing, vol.40, No.4, pp.950-962, April 2002.

[8] Debevec P. E., Malik J. “Recovering High Dynamic Range Radiance Maps from Photographs”. International Conference on Computer Graphics and Interactive Techniques archive, Proceedings of the 24th annual conference on Computer graphics and interactive techniques table of contents, pp 369 - 378, 1997.

[9] Dijkstra E. W., A note on two problems in connection with graphs, Numerishe Mathematik, 1959, pp. 269-271.

[10] Doheny, R. C. “Handheld Standoff Mine Detection System (HSTAMIDS) Field Evaluation in Thailand”.

[11] Dubins L. E., On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. Amer. J. Math., vol. 79, pp. 497-516, 1957.

[12] Floyd, R. W., Algorithm 97: Shortest Path, Communications of the ACM, 1962.

[13] Freese M. A. “Improved Landmine Discrimination With an Off-the-Shelf Metal Detector”, Journal of Mine Action, Vol 2,No 1, pp 93-99, Oct. 2008.

[14] Gaal M., Trial design for testing and evaluation in humanitarian demining mine clearence, Ph.D. thesis, Brandenburg Technical University Cottbus, Germany, 2007. Available at http://www.itep.ws/pdf/PhD_Gaal.pdf.

[15] Gao P., Collins L., Garber P. M., Geng N., Carin L. “Classification of Landmine-Like Metal Targets Using Wibeband Electromagnetic Induction”. IEEE Trans. Geosci. Remote Sensing, vol.38, No.3, pp.1352-1361, May 2000.

[16] Garmin. Edge 205/305. GPS-enabled computer for cyclists. Owner’s Manual. 2006.

[17] Guelle D., Smith A., Lewis A. and Bloodworth T. “Metal Detector Handbook for Humanitarian Demining”. 2003.

[18] Habib M. K. (2007). “Humanitarian Demining: Reality and the Challenge of Technology - The State of the Arts”. International Journal of Advanced Robotic Systems Vol. 4, 2, 151-172.

[19] Hahne U., Alexa M. “Exposure Fusion for Time-Of-Flight Imaging”. Pacifics Graphics, Volume 30, Number 7, 2011.

[20] Hayter D. "Training Dogs to Detect Tripwires." Mine Detection Dogs: Training Operations and Odour Detection, June 2003. Geneva International Center for Humanitarian Demining. http://www.gichd.org/fileadmin/pdf/publications/MDD/MDD_ch2_part3.pdf. Accessed in July 3rd, 2013.

[21] Hirose S., Yokota S., Torii A., Ogata M., Suganuma S., Takita K., Kato K. (2005). “Quadruped Walking Robot Centered Demining System - Development of TITAN-IX and its Operation”. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pages 1284-1290, Barcelona, Spain, April 2005.

[22] Ho K.C., Collins L. M., Huttel L. G., Gader P. D. “Discrimination Mode Processing for EMI and GPR Sensors for Hand-Held Land Mine Detection”. IEEE Trans. Geosci. Remote Sensing, vol.42, No.1, pp.249-263, January 2004.

[23] Hrabar S., Corke P., Bosse M. “High Dynamic Range Stereo Vision for Outdoor Mobile Robotics”. IEEE International Conference on Robotics and Automation. pp 430-435. Kobe, Japan. 2009.

[24] Irie K., Yoshida T., Tomono M. “A High Dynamic Range Vision Approach to Outdoor Localization”. IEEE International Conference on Robotics and Automation. pp 5179-5184. Shanghai, China, 2011.

[25] Kaneko A. M. and Fukushima E. F. “Development of an Automatic Landmine Detection and Marking System for the Demining Robot Gryphon”. Journal of Advanced Computational Intelligence and Intelligent Informatics. Vol. 15 No. 6, pages 737-743, 2011.

[26] Kaneko A. M., Endo G. and Fukushima E. F. “Landmine Buried Depth Estimation by Curve Characterization of Metal Mine Detector Signals”, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Tokyo, Japan, November 2013.

[27] Kaneko A. M., Endo G. and Fukushima E. F. “Proposal of a Discrimination Method for Landmines and Metal Fragments Using Metal Detectors”. Journal of Mine Action. 2013.

[28] Kaneko A. M. and Fukushima E. F. “Humanitarian Demining Robot Gryphon - Mine Detector Scanning Procedure Optimization”. JSME Robomec 2013, Tsukuba, Japan.

[29] Kaneko A. M., Marino M., Fukushima E. F. “Humanitarian Demining Robot Gryphon: New Vision Techniques and Optimization Methods”. IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems (IROS), Taipei, Taiwan, October 2010.

[30] Kruger H., Ewald H. “Handheld metal detector with online visualization and classification for the humanitarian mine clearance”. Sensors, 2008 IEEE, pp.415-418, 2008.

[31] Li J., Kaneko A. M. and Fukushima E. F. "Humanitarian Demining Robot Gryphon - Self-Calibration Using Stereo Vision Camera". JSME Robomec 2013, Tsukuba, Japan.

[32] Mahoney A. M., Durgin A., Poling A., "Mine Detection Rats: Effects of Repeated Extinction on Detection Accuracy", Journal of Mine Action, Vol 16, No 3, pp 61-64, Fall. 2012.

[33] Mertens T., Kautz J. and Reeth F. V. “Exposure Fusion”, Pacific Graphics, 2007.

[34] Minelab Electronics. “The Minelab Eureka Gold”.Instruction Manual.

[35] Minelab Eletronics. “F3 Metal Mine Detector - Instructors Notes and Syllabus”. Issue 1.3 March, 2006.

[36] Minelab Electronics. “Explorer SE - Quick Start”. Instruction Manual.

[37] Marino M. (2009). “Development of New Artificial Vision Techniques for Improving the Gryphon Humanitarian Demining Robot”. Master Thesis, Politecnico di Milano, Italy.

[38] Narita T., Fukushima E. F., Hirose S. “Study of Discrimination Methods for Metal Fragments and Mines in Humanitarian Demining Tasks”. The 2010 International Symposium on Intelligent Systems (iFan). Tokyo, Japan. 2010.

[39] Nicoud J. D., Habib M. (1995). “The Pemex-B autonomous demining robot: perception and navigation strategies”. In Proceedings of the international Conference on intelligent Robots and Systems-Volume 1 - Volume 1 (August 05 - 09, 1995). IROS. IEEE Computer Society, Washington, DC, 419.

[40] Osuna E. E., 1985. The Psychological Cost of Waiting, Journal of Mathematical Psychology, 29: 1, 82-105.

[41] Pavkovic N., Ishikawa J., Furuta K., Takahashi K., Gaal M., Guelle D. “Test and Evaluation of Japanese GPR-EMI Dual Sensor Systems at Benkovac Test Site in Croatia”, HCR-CTRO_TECH_GPR 08-001, March, 2008.

[42] Point Grey. “Bumblebee Stereo Vision Camera Systems”. Datasheet. June 2012.

[43] Reeds J. A. and Shepp L. A. “Optimal paths for a car that goes both forward and backward”. J. Pacific Math., vol. 145, no. 2, pp. 367-393, 1990.

[44] Riggs L. S., Mooney J. E., Lawrence D. E. “Identifilcation of Metallic Mine-like Objects Using Low Frequencey Magnetic Firleds”, IEEE Trans. Geosci. Remote Sensing, vol.39, No.1, pp.249-263, January 2004.

[45] Sato M., Fujiwara J., Feng X., Zhou Z., Kobayashi T. “Development of a hand-held GPR MD sensor system (ALIS)”. Proc.Of SPIE Vol. 5794.Detection and Remediation Technologies for Mines and Minelike Targets X, pp. 1000-1007, 2005.

[46] Soueres P. and Laumond J. P., Shortest paths synthesis for a car-like robot. IEEE Trans. Automat. Contr., vol. 41, pp. 672-688, May 1996.

[47] Stanley R. J., Ho K. C., Gader P., Wilson J. N., Devaney J. "Land Mine and Clutter Object Discrimination Using Wavelet and Time Domain Spatially from Metal Detectors and Their Fusion with GPR Features for Hand-Held Units". Circuits Systems Signal Processing, vol.26, No.2, pp.165-191, 2007.

[48] Trevelyan J. "Mine Detection Dogs in Use", Demining Research, University of Western Australia.

[49] Waymond S. Jr, Schroeder C., Kim K., Cheng J. (2000). "Detection of Land Mines Using Elastic and Electromagnetic Waves". Available: http://www.acoustics.org/press/139th/larson.htm

[50] Wetzel C. M., Kneebone R. L., Woloshynowych M., et al. The effects of stress on surgical performance, Am J Surg 2006; 191: 5-10.

[51] Yamanashi Hitachi Construction Machinery. "Contributing to the Mechanization of Demining in Colombia's Mountainous Regions". CSR Report. pp 15-16. 2010. http://www.hitachi-c-m.com/global/pdf/generator/company/csr/report1 0/09.pdf. Acessed July 3rd, 2013.

# Appendix A

# Implemented Algorithms

## A.1 Transforming MMD Raw Signals Into Polynomials

```
function amkPPbuildtests =
    amkPPbuildtests(mode)

if mode == 1
num_files = 540; %all 540
elseif mode == 0
num_files = 368; %(can be any
     number according to
    convenience)
end %mode

for h=1:8
for i=1:num_files
ROOTS(i,h) = -999;
ROOTS2(i,h) = -999;
end
end

for b=1:num_files
MF(1,b) = -999;
end

for f= 1:num_files
if f > 0

if mode == 1
name1 = strcat('D:\
    N_AMK_DATA_CH1\',int2str(f
    ),'.csv');
name2 = strcat('D:\
    N_AMK_DATA_CH2\',int2str(f
    ),'.csv');
elseif mode == 0
name1 = strcat('D:\fixed\',
    int2str(f),'.csv');
name2 = strcat('D:\fixed\',
```

```
    int2str(f),'.csv');
end

M1=csvread(name1);
M2=csvread(name2);

ch_fail = 0;

%IMPORTANT!!
%Depending on the situation,
    X is Y or vice versa
unit = 1;%10;%1;
factor = 1;%255*255;%1;
X=M1(:,2)/unit; %line
Y=M1(:,1)/unit;
%signal are different for lab
     and cambodia data
A=M2(:,6); %MD signals (
    already in the form Ch2 -
    Ch1)

MDmax = 0;
maxindex = 1;
stop = 0;
MDmin = 0;

MEDIAN_A = A;
%median calculation:
%Bubble sorting:
if length(MEDIAN_A) > 0
for k = 1:length(MEDIAN_A)
for l = 1: (length(MEDIAN_A)
    -1)
if MEDIAN_A(l) > MEDIAN_A(l
    +1)
temp_A = MEDIAN_A(l);
MEDIAN_A(l) = MEDIAN_A(l+1);
MEDIAN_A(l+1) = temp_A;
end
end
end
end %if length

rest = mod(length(MEDIAN_A)
    ,2);
if rest == 0
median_value = (MEDIAN_A(
    length(MEDIAN_A)/2) +
    MEDIAN_A(length(MEDIAN_A)
    /2 + 1))/2;
elseif rest == 1
median_value = MEDIAN_A(fix(
    length(MEDIAN_A)/2) + 1);
end

A = A - median_value;

%checking line X coords
    limits:::
maxYlimit = Y(1);
minYlimit = Y(1);

for k=1:length(Y)
if Y(k) > maxYlimit
maxYlimit = Y(k);
end
```

```
if Y(k) < minYlimit
minYlimit = Y(k);
end
end

maxXlimit = X(1);
minXlimit = X(1);

for k=1:length(X)
if X(k) > maxXlimit
maxXlimit = X(k);
end
if X(k) < minXlimit
minXlimit = X(k);
end
end
%end of line limits

y_noise = 0;
x_noise = 0;

for k=1:length(X)
if k == 0
stop = 1;
break;
end
if Y(k) > (minYlimit +
   y_noise) && Y(k) < (
   maxYlimit - y_noise)
if mode == 0
if X(k) >= (minXlimit +
   x_noise) && X(k) <= (
   maxXlimit - x_noise)
if (abs(A(k))>MDmax)
MDmax = abs(A(k));
maxindex = k;
Ymax = X(maxindex);
end %if
end
else
if (abs(A(k))>MDmax)
MDmax = abs(A(k));
maxindex = k;
Ymax = X(maxindex);
end %if
end
end
end %k=1:length(X)

%for avoiding repeated points
     in MDmax
equal_MD_max = 0;
index = 0;
MDmax_repeat = A(maxindex);

for k=1:length(X)
if k == 0
stop = 1;
break;
end
if Y(k) > (minYlimit +
   y_noise) && Y(k) < (
   maxYlimit - y_noise)
if X(k) >= (minXlimit +
   x_noise) && X(k) <= (
   maxXlimit - x_noise)
```

```
if A(k) == MDmax_repeat
index = index + 1;
equal_MD_max(index) = k;
end
end
end
end

if length(equal_MD_max) > 2
old = equal_MD_max(1);
max_ind = old;
repeat = 1;
max_repeat = 0;
for k=2:length(equal_MD_max)
if (equal_MD_max(k) - old) ==
    1
if k == length(equal_MD_max)
repeat = repeat + 1;
if repeat > max_repeat
max_repeat = repeat;
max_ind = equal_MD_max(k);
repeat = 0;
end
else
repeat = repeat + 1;
end
else
if repeat > max_repeat
max_repeat = repeat;
max_ind = old;
repeat = 0;
end
end
old = equal_MD_max(k);
end
maxindex = max_ind - round(
    max_repeat/2);
Ymax = X(maxindex);
end

if A(maxindex) > 0
SIGNAL(1,f) = 1;
COPY_A = A;
pol = 1;
else
SIGNAL(1,f) = -1;
COPY_A = -A;
pol = -1;
end

MD1 = abs(COPY_A(maxindex));
   %it is OK if positive or
    negative

stop = 0;

if (maxindex+3)<length(A) & (
    maxindex-3)>0
if X(maxindex+3) > X(maxindex
    -3)
dir = 1;
else
dir = 0;
end
else
fprintf('The depth can not be
```

```
    estimated (check1)!\n')
stop = 1;
end

if stop == 0
%Lets extract the information
    of whole line:
lim_r = maxindex+1;
lim_l = maxindex-1;
left = 0;
right = 0;
go_right = 1;
go_left = 1;

if dir == 1
while X(lim_r+1)>=X(lim_r) |
   X(lim_r+2)>=X(lim_r) | X(
   lim_r+3)>=X(lim_r)
lim_r = lim_r + 1;
if (lim_r+3)>length(X)
break;
end
end
while X(lim_l-1)<=X(lim_l) |
   X(lim_l-2)<=X(lim_l) | X(
   lim_l-3)<=X(lim_l)
lim_l = lim_l - 1;
if (lim_l-3)<0
break;
end
end
else %dir == 1
while X(lim_r+1)<=X(lim_r) |
   X(lim_r+2)<=X(lim_r) | X(
   lim_r+3)<=X(lim_r)
lim_r = lim_r + 1;
if (lim_r+3)>length(X)
break;
end
end
while X(lim_l-1)>=X(lim_l) |
   X(lim_l-2)>=X(lim_l) | X(
   lim_l-3)>=X(lim_l)
lim_l = lim_l - 1;
if (lim_l-3)<0
break;
end
end
end %dir == 1

if(left == 0)
left = lim_l;
end
if(right == 0)
right = lim_r;
end

%finding targets center
%putting in an approximate "
   center" first
xmass1 = X(maxindex);
ymass1 = Y(maxindex);

X = X - xmass1;
Y = Y - ymass1;
```

```
noiselimit =
    5*10^6/(255*255*19931)
    *100;
lowest_acceptableMD = 0;
    %2.5*10^6;

final_index = 0;
for k=left:right
final_index = final_index +
    1;
v_interpol_x(final_index) = X
    (k); %(X(k) - X(maxindex))
    ;
v_interpol_y(final_index) = Y
    (k); %(Y(k) - Y(maxindex))
    ;
v_interpol_MD(final_index) =
    COPY_A(k);
end

low_n = 12;
high_n = 20;
pol_degree = high_n;

%storing all lines
%Lets extract the information
     of the whole line:
avoid_left = -999;
avoid_right = 999;

line_index = 1;
lines = 0;
line_interval = 1; %for
    removing lines for
    experiments
inserted_line = 0;
closest_line = 999;
line_with_max = 0;

max_detectable = 0;
detectable = 0;

while line_index < length(X)
if (line_index+3)<length(
    COPY_A) & (line_index-3)>0

if X(line_index+3) > X(
    line_index)
dir = 1;
else
dir = 0;
end
current_line_index =
    line_index;

if dir == 1
while X(current_line_index+1)
    >=X(current_line_index) |
    X(current_line_index+2)>=X
    (current_line_index) | X(
    current_line_index+3)>=X(
    current_line_index)
current_line_index =
    current_line_index + 1;
if (current_line_index+3)>
    length(X)
```

```
break;
end
end
else %dir == 1
while X(current_line_index+1)
   <=X(current_line_index) |
   X(current_line_index+2)<=X
   (current_line_index) | X(
   current_line_index+3)<=X(
   current_line_index)
current_line_index =
   current_line_index + 1;
if (current_line_index+3)>
   length(X)
break;
end
end
end %dir == 1

if abs(current_line_index -
   line_index) > 2
interpol_index = 0;
for k=line_index:
   current_line_index
if X(k) >= avoid_left & X(k)
   <= avoid_right
interpol_index =
   interpol_index + 1;
LINE_MD(interpol_index) =
   COPY_A(k);
LINE_X(interpol_index) = X(k)
   ;
LINE_Y(interpol_index) = Y(k)
   ;
end
end

MDfit_LINE = polyfit(LINE_X,
   LINE_Y,1);
MDfit_POL = polyfit(LINE_X,
   LINE_MD,pol_degree);

line_index =
   current_line_index;
lines = lines + 1; %also used
    later!

if rem(lines-1,line_interval)
    == 0 & length(LINE_X) >
   10
inserted_line = inserted_line
    + 1; %also used later!
LINE(1,inserted_line) =
   MDfit_LINE(1);
LINE(2,inserted_line) =
   MDfit_LINE(2);

for j=1:(pol_degree+1)
POL(j,inserted_line) =
   MDfit_POL(j);
end
end

clear LINE_MD;
clear LINE_X;
clear LINE_Y;
```

```
else
line_index = line_index + 1;
end %if abs(
    current_line_index -
    line_index) > 2
else %if (line_index+3)<
    length(A) & (line_index-3)
    >0
line_index = line_index + 1;
end
end %while i < length(X)

if ch_fail == 0
plotNum2 = plotNum2 + 1;

%from here, the rotation
    center should be already
    the
%"center" of the target
%theta = atan(MDfit(1));
theta = 0;
alfa_step = 180;
alfa = 0;

if f > 330 & f < 336
alfa = 90;
end
if f == 351 | f == 352 | f ==
     363 | f == 364 | f == 365
alfa = 90;
end

maxx = X(1);
minx = X(1);
maxy = Y(1);
miny = Y(1);
for k=1:length(X)
if X(k) > maxx
maxx = X(k);
end
if X(k) < minx
minx = X(k);
end
if Y(k) > maxy
maxy = Y(k);
end
if Y(k) < miny
miny = Y(k);
end
end

num_cuts = 0;
num_inf = 0;
max_num_inf = 0;
max_inf_angle = 0;
max_x2_angle = -999;
max_x2 = 0;
max_x2_2nd_peak = 0;
max_x2_big_neg = 0;
x2_prop = 0;
max_x2_prop = 0;
max_x2_prop_angle = 0;
posture_index = 0;

max_size = 0;
max_size_ang = 0;
```

```
while alfa < 180
%%%%%%%%%%%%%%%%%%
pol_degree = high_n; %
    important (this value is
    changed along the program,
     so very important to
    rewrite)
%%%%%%%%%%%%%%%%%%
if alfa == 0
alfa_x = v_interpol_x;
alfa_MD = v_interpol_MD;
else
tot_angle = theta + alfa;
a_d = tan(tot_angle*pi/180);
b_d = 0; %Y(maxindex) - a_d*X
    (maxindex);

final_index = 0;
for n=1:inserted_line
ai = LINE(1,n);
bi = LINE(2,n);

current_x = bi/(a_d - ai);
current_y = a_d*current_x;

for j=1:(pol_degree+1)
pol_coefs(j) = POL(j,n);
end

average = polyval(pol_coefs,
    current_x);

if abs(average) >=
    lowest_acceptableMD
if current_x >= minx &
    current_x <= maxx &
    current_y >= miny &
    current_y <= maxy & abs(
    average) <= 120
final_index = final_index +
    1;
alfa_x(final_index) = (
    current_x)*cos(tot_angle*
    pi/180) + (current_y)*sin(
    tot_angle*pi/180);
alfa_y(final_index) = -(
    current_x)*sin(tot_angle*
    pi/180) + (current_y)*cos(
    tot_angle*pi/180);
alfa_MD(final_index) =
    average;
end
end
end % for n=1:lines
end %if alfa == 0

%inputting White Gaussian
    Noise:
noise_vector_size = length(
    alfa_MD);
media = 0;
var = 0.1; % 0.1, 0.05, 0.01
noise_vector = media + var*
    randn(1,noise_vector_size)
    ;
```

```
alfa_MD_backup = alfa_MD; %
    not used, but just to
    store for debugging

% for noise_index=1:
    noise_vector_size
% alfa_MD(noise_index) =
    alfa_MD(noise_index)*(1 +
    noise_vector(noise_index))
    ;
% end

%Filtering the noise:
alfa_MD_noisy = alfa_MD; %
    just for debugging
alfa_MD = smooth(alfa_MD,5);
    %Moving Average

%FINDING XMIN E XMAX (VERY
    IMPORTANT PART!!)
xmin = 0;
xmax = 0;
second_peak_index = 0;
big_neg_inf_index = 0;
exact_MDmax = MD1;
final_ind = 0;
if final_index > 4
max_MD = 0;
max_x = 0;

for a=1:length(alfa_x)
if alfa_MD(a) > max_MD & abs(
    alfa_x(a)) <=5
max_MD = alfa_MD(a);
max_x = alfa_x(a);
end
end
alfa_x = alfa_x - max_x;

[xmin xmax exact_MDmax
    second_peak_index
    big_neg_inf_index] =
    limits4(alfa_x,alfa_MD,f);
     %its not the exact MDmax,
     but is ok...

if exact_MDmax == 0
exact_MDmax = MD1;
end

if xmin == 0 & xmax == 0
xmin = min(alfa_x);
xmax = max(alfa_x);
end

if second_peak_index == 0 &
    big_neg_inf_index == 0
pol_degree = 4;
signal_type1 = 1;
else
pol_degree = low_n;
signal_type1 = 0;
end

%saving the theoretical good
    points
```

```
range = 0; %can be anything
    needed
final_ind = 0;
for i=1:length(alfa_x)
if alfa_x(i) >= (xmin - range
    ) & alfa_x(i) <= (xmax +
    range)
final_ind = final_ind + 1;
final_alfa_x(final_ind) =
    alfa_x(i);
final_alfa_MD(final_ind) =
    alfa_MD(i);
W(final_ind) = 1;
end
end

%align signals using 1st and
    2nd peaks
if second_peak_index ~= 0
if alfa_x(second_peak_index)
    < 0
final_alfa_x = -final_alfa_x;
%We have to invert xmax and
    xmin as well (VERY
    IMPORTANT!!!):
x_aux = xmin;
xmin = -xmax;
xmax = -x_aux;
end
end %align signals
end %if final_index > 4
%::::::::::::

if final_ind > pol_degree |
    few_points == 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    finding the exact MDmax
    and position:
MDfittemp = wpolyfit(
    final_alfa_x,final_alfa_MD
    ,pol_degree,W);

for b=1:(pol_degree+1)
MD_dertemp(b) = (pol_degree -
     b + 1)*MDfittemp(b);
end

exact_x = 0;
raizes = roots(MD_dertemp);
index_inf = 0;
for i=1:length(raizes)
if imag(raizes(i)) == 0
if polyval(MDfittemp,raizes(i
    )) > 0.5*exact_MDmax &
    raizes(i) <= xmax & raizes
    (i) >= xmin
index_inf = index_inf + 1;
peak_pt(index_inf) = raizes(i
    );
peak_val(index_inf) = polyval
    (MDfittemp,raizes(i));
end
end
end

%Bubble sorting:
```

```
if index_inf > 0
for k = 1:length(peak_val)
for l = 1: (length(peak_val)
    -1)
if peak_val(l) < peak_val(l
    +1)
temp_A = peak_val(l);
peak_val(l) = peak_val(l+1);
peak_val(l+1) = temp_A;

temp_B = peak_pt(l);
peak_pt(l) = peak_pt(l+1);
peak_pt(l+1) = temp_B;
end
end
end

%in case we just want adjust
    a already found peak
%(theoretically close to 0)
exact_x = 0;
exact_MD_x = 0;
for k = 1:length(peak_pt)

if abs(peak_pt(k) - abs(
    exact_x)) < 5 & peak_val(k
    ) > exact_MD_x
exact_x = peak_pt(k);
exact_MD_x  = peak_val(k);
end
end

clear peak_pt;
clear peak_val;
end %if length
%just in case:
clear MDfittemp;
clear MD_dertemp;
clear raizes;
%%%%%%%%%%%%%%
alfa_x = alfa_x - exact_x;
final_alfa_x = final_alfa_x -
     exact_x;
xmin = xmin - exact_x;
xmax = xmax - exact_x;

if xmin ~= 0 | xmax ~= 0

alfa_x_norm = final_alfa_x;
alfa_MD_norm = final_alfa_MD;
MDfit_norm2 = polyfit(
    alfa_x_norm,alfa_MD_norm,
    pol_degree);

%just for tests if we wanna
    print non-normalized
    signals
plotNum = plotNum + 1;

%final xmin & xmax
    calculation:
second_peak_y_exact = 0;
second_peak_MD_exact = 0;
big_neg_inf_exact = 0;
[xmin xmax exact_MDmax
    MDmax_pt
```

```
    second_peak_y_exact
    second_peak_MD_exact
    big_neg_inf_exact
    big_neg_inf_exact_pos] =
    limits4b(MDfit_norm2,f,
    xmin,xmax,pol_degree,
    signal_type1);

%TEST
if xmin == 0 & xmax == 0
xmin = min(alfa_x_norm);
xmax = max(alfa_x_norm);
end

if second_peak_y_exact == 0 &
     big_neg_inf_exact == 0
pol_degree = 4;
else
pol_degree = low_n;
big_neg_inf_index;
end

%TEST
if few_points == 1
pol_degree = min_error_degree
    ;
xmin = min(alfa_x_norm);
xmax = max(alfa_x_norm);
end
%TEST

MDfit_norm2 = polyfit(
    alfa_x_norm,alfa_MD_norm,
    pol_degree);

error_interpol = 0;
zero_value = 0;
for ind=1:length(alfa_x_norm)
if alfa_MD_norm(ind) ~= 0
interpol_MD = polyval(
    MDfit_norm2,alfa_x_norm(
    ind));
error_interpol =
    error_interpol + abs((
    interpol_MD - alfa_MD_norm
    (ind))/alfa_MD_norm(ind))
    *100;
else
zero_value = zero_value + 1;
end
end

for c=1:(high_n + 1)%
if c < (high_n - pol_degree)
    + 1
COEF_MD_CUT(c,plotNum) = 0;
else
COEF_MD_CUT(c,plotNum) =
    MDfit_norm2(c-(high_n -
    pol_degree));
end
end
COEF_MD_CUT(c+1,plotNum) = f;
COEF_MD_CUT(c+2,plotNum) =
    error_interpol/(length(
    alfa_x_norm) - zero_value)
```

```
    ;
COEF_MD_CUT(c+3,plotNum) =
    alfa;
COEF_MD_CUT(c+4,plotNum) =
    xmin;
COEF_MD_CUT(c+5,plotNum) =
    xmax;
COEF_MD_CUT(c+6,plotNum) =
    second_peak_y_exact;
COEF_MD_CUT(c+7,plotNum) =
    big_neg_inf_exact_pos;
COEF_MD_CUT(c+8,plotNum) =
    pol;
COEF_MD_CUT(c+9,plotNum) =
    exact_MD_x;

num_cuts = num_cuts + 1;
else
plotNum = plotNum + 1;
for c=1:(high_n + 1)%
COEF_MD_CUT(c,plotNum) = 0;
end
end %if xmin ~= 0 | xmax ~= 0
else
plotNum = plotNum + 1;
for c=1:(high_n + 1)%
COEF_MD_CUT(c,plotNum) = 0;
end
end %if final_ind >
    pol_degree
alfa = alfa + alfa_step;
%tot_angle = theta + alfa;
num_inf = 0;
clear alfa_x_norm;
clear alfa_MD_norm;
clear alfa_x_norm;
clear v_interpol_x_norm;
clear final_alfa_x;
clear final_alfa_MD;
clear alfa_x;
clear alfa_y;
clear alfa_MD;
clear W;
end %while alfa
end % ch_fail
clear v_interpol_x;
clear v_interpol_y;
clear v_interpol_MD;
end %if stop == 0
end %if f == choice

clear M1;
clear M2;
end %end reading each file

COEF_MD_CUT = COEF_MD_CUT';

csvwrite('D:\COEFS_MD_CUT.csv
    ',COEF_MD_CUT);
csvwrite('D:\LINE_ERROR.csv',
    LINE_ERROR);
csvwrite('D:\DET.csv',DET);

clear all
```

## A.2 Discriminating Signals In the Database

```
function amkPPbuildtests =
    amkPPbuildtests
format long

warning off;

name = strcat('D:\database.
    csv');
name_ch2 = strcat('D:\
    database_ch2.csv');
M=csvread(name);
M_ch2=csvread(name_ch2);

pol_degree = 12;
num_files = length(M(:,1)); %
    database length

for f=1:340 %num_files
%vector positions:
%1 ~ pol_degree + 1: pol
    coefficients
%pol_degree + 2: data number
%pol_degree + 3: Not Used (NU
    )
%pol_degree + 4: NU
%pol_degree + 5: xmin
%pol_degree + 6: xmax
%pol_degree + 7: NU
%pol_degree + 8: NU
%pol_degree + 9: NU
%pol_degree + 10: depth
%pol_degree + 11: mine (1) or
     not (0)
%pol_degree + 12: target type
%pol_degree + 13: polarity
%pol_degree + 14: exact MDmax
%pol_degree + 15: norm/MDmax
%pol_degree + 16: g
%from the maximum MD center,
    so a0 is not the maximum
    MD)

vector_size = 16;
data_info = zeros(1,
    pol_degree + vector_size);
data_info_ch2 = zeros(1,
    pol_degree + vector_size);
for i=1:(pol_degree +
    vector_size)
data_info(i) = M(f,i);
data_info_ch2(i) = M_ch2(f,i)
    ;
end

closest_curve = 999;
closest_depth = 999;
closest_norm = 999;

g = 1;
final_mine = 1;
final_target = 0; %from 1 to
    40, according to type
```

```
target_type_old = M(1,
    pol_degree + 12);

plotNum = 0;

inserted_index = 0;

while g <= num_files %we do
    the searching here

index = 0;

target_type_actual = M(g,
    pol_degree + 12);
target_type_old = M(g,
    pol_degree + 12);

while  target_type_actual ==
    target_type_old &
    data_info(pol_degree + 13)
     == M(g,pol_degree + 13)
if g ~= f %we dont want the
    real data on the analysis
    !!
index = index + 1;
for i=1:(pol_degree +
    vector_size)
type_data_temp(index,i) = M(g
    ,i);     %HERE
type_data_ch2_temp(index,i) =
     M_ch2(g,i); %HERE
end
end %if g ~= f

g = g + 1;

target_type_old =
    target_type_actual;
target_type_actual = M(g,
    pol_degree + 12);
end %target_type_actual ==
    target_type_old

if index > 0 %if polarity is
    the same as the data

if type_data_temp(1,
    pol_degree + 14) >=
    data_info(pol_degree + 14)
     & type_data_temp(index,
    pol_degree + 14) <=
    data_info(pol_degree + 14)
     %if the data is within
    the limits of the group

num_data_per_group = 6;

if index > num_data_per_group
%program for selecting only
    close type_data from the
%data
for i=1:(pol_degree +
    vector_size)
type_data_temp(index + 1,i) =
     data_info(i);
```

```
type_data_ch2_temp(index + 1,
    i) = data_info_ch2(i);
end
%bubble sort for ordering
    from small to big values
for kk=1:(index + 1)
for ll=1:(index)
if type_data_temp(ll,
    pol_degree + 14) <
    type_data_temp(ll + 1,
    pol_degree + 14)
for i=1:(pol_degree +
    vector_size)
temp_type = type_data_temp(ll
    ,i);
type_data_temp(ll,i) =
    type_data_temp(ll + 1,i);
type_data_temp(ll + 1,i) =
    temp_type;

temp_type_ch2 =
    type_data_ch2_temp(ll,i);
type_data_ch2_temp(ll,i) =
    type_data_ch2_temp(ll + 1,
    i);
type_data_ch2_temp(ll + 1,i)
    = temp_type_ch2;
end
end
end
end

index_data = 0;
for kk=1:(index + 1)
if type_data_temp(kk,
    pol_degree + 14) ==
    data_info(pol_degree + 14)
index_data = kk;
end
end

counter = 0;
up = 1;
down = 1;

while counter ~=
    num_data_per_group
if((index_data - up) > 0)
counter = counter + 1;
for kk=1:(pol_degree +
    vector_size)
type_data(counter,kk) =
    type_data_temp((index_data
     - up),kk);
type_data_ch2(counter,kk) =
    type_data_ch2_temp((
    index_data - up),kk);
end
up = up + 1;
end

if counter ==
    num_data_per_group
break;
end
```

```
if((index_data + down) <= (
    index + 1))
counter = counter + 1;
for kk=1:(pol_degree +
    vector_size)
type_data(counter,kk) =
    type_data_temp((index_data
     + down),kk);
type_data_ch2(counter,kk) =
    type_data_ch2_temp((
    index_data + down),kk);
end
down = down + 1;
end
end % while counter ~=
    num_data_per_group

%final bubble sort for
    rearranging the data:
for kk=1:num_data_per_group
for ll=1:(num_data_per_group
   - 1)
if type_data(ll,pol_degree +
    1) < type_data(ll + 1,
    pol_degree + 1)
for i=1:(pol_degree +
    vector_size)
temp_a0 = type_data(ll,i);
type_data(ll,i) = type_data(
    ll + 1,i);
type_data(ll + 1,i) = temp_a0
    ;
temp_a0_ch2 = type_data_ch2(
    ll,i);
type_data_ch2(ll,i) =
    type_data_ch2(ll + 1,i);
type_data_ch2(ll + 1,i) =
    temp_a0_ch2;
end
end
end
end

%something not elegant, but:
index = num_data_per_group;
else
type_data = type_data_temp;
type_data_ch2 =
    type_data_ch2_temp;
end %if index >
    num_data_per_group

clear type_data_temp;
clear type_data_ch2_temp;
%HERE

interpol_depths = zeros(1,
    index);
interpol_depths_ch2 = zeros
    (1,index);
interpol_a0 = zeros(1,index);
interpol_a0_ch2 = zeros(1,
    index);
for j=1:index
interpol_depths(j) =
```

```
    type_data(j,pol_degree +
    10);
interpol_depths_ch2(j) =
    type_data_ch2(j,pol_degree
     + 10);
interpol_a0(j) = type_data(j,
    pol_degree + 14);
interpol_a0_ch2(j) =
    type_data_ch2(j,pol_degree
     + 14);
end %for j=1:index

if index > 3
fitorder = 3;
else
fitorder = 2;
end

a0_depths = polyfit(
    interpol_a0,
    interpol_depths,fitorder);
     % 3rd order fits ok
a0_depths_ch2 = polyfit(
    interpol_a0_ch2,
    interpol_depths_ch2,
    fitorder); % 3rd order
    fits ok

%if extrapolation:
if data_info(pol_degree + 14)
     > interpol_a0(1)
theor_depth = interpol_depths
    (1) + (data_info(
    pol_degree + 14) -
    interpol_a0(1))*(
    interpol_depths(1) -
    interpol_depths(2))/(
    interpol_a0(1) -
    interpol_a0(2));
elseif data_info(pol_degree +
     14) < interpol_a0(index)
theor_depth = interpol_depths
    (index-1) + (data_info(
    pol_degree + 14) -
    interpol_a0(index-1))*(
    interpol_depths(index-1) -
     interpol_depths(index))/(
    interpol_a0(index-1) -
    interpol_a0(index));
else%if interpolation
theor_depth = polyval(
    a0_depths,data_info(
    pol_degree+1));
end

%if extrapolation:
if data_info_ch2(pol_degree +
     14) > interpol_a0_ch2(1)

theor_depth_ch2 =
    interpol_depths_ch2(1) + (
    data_info_ch2(pol_degree +
     14) - interpol_a0_ch2(1))
    *(interpol_depths_ch2(1) -
     interpol_depths_ch2(2))/(
    interpol_a0_ch2(1) -
```

```
    interpol_a0_ch2(2));
elseif data_info_ch2(
    pol_degree + 14) <
    interpol_a0_ch2(index)
theor_depth_ch2 =
    interpol_depths_ch2(index
    -1) + (data_info_ch2(
    pol_degree + 14) -
    interpol_a0_ch2(index-1))
    *(interpol_depths_ch2(
    index-1) -
    interpol_depths_ch2(index)
    )/(interpol_a0_ch2(index
    -1) - interpol_a0_ch2(
    index));
else%if interpolation
theor_depth_ch2 = polyval(
    a0_depths_ch2,
    data_info_ch2(pol_degree +
     14));
end

if abs(theor_depth) > abs(
    theor_depth_ch2)
ref_depth = abs(theor_depth);
else
ref_depth = abs(
    theor_depth_ch2);
end
total_theor_depth_error = abs
    (theor_depth -
    theor_depth_ch2);%/
    ref_depth*100;
```

```
xmin_theor = 0;
xmax_theor = 0;

xmin_theor_ch2 = 0;
xmax_theor_ch2 = 0;

if theor_depth >= 0 &
    theor_depth <= 45 &
    theor_depth_ch2 >= 0 &
    theor_depth_ch2 <= 45 &
    total_theor_depth_error <
    1.4 %2.5

xmin_theor = 999;
xmax_theor = 999;
xmin_theor_ch2 = 0;
xmax_theor_ch2 = 0;

for k=1:index
if abs(xmin_theor) > abs(
    type_data(k,pol_degree+5))
xmin_theor = type_data(k,
    pol_degree+5);
end
if abs(xmax_theor) > abs(
    type_data(k,pol_degree+6))
xmax_theor = type_data(k,
    pol_degree+6);
end
if abs(xmin_theor_ch2) < abs(
    type_data_ch2(k,pol_degree
    +5))
```

```
xmin_theor_ch2 =
    type_data_ch2(k,pol_degree
    +5);
end
if abs(xmax_theor_ch2) < abs(
    type_data_ch2(k,pol_degree
    +6))
xmax_theor_ch2 =
    type_data_ch2(k,pol_degree
    +6);
end
end

pol_index = 0;
pol_index_ch2 = 0;

interpol_type = zeros(1,index
    );
interpol_type_ch2 = zeros(1,
    index);
for a=1:index
for i=1:(pol_degree+1)
interpol_type(i) = type_data(
    a,i);
interpol_type_ch2(i) =
    type_data_ch2(a,i);
end %for j=1:index
xmin = type_data(a,pol_degree
    +5);
xmax = type_data(a,pol_degree
    +6);

xmin_ch2 = type_data_ch2(a,
    pol_degree+5);
xmax_ch2 = type_data_ch2(a,
    pol_degree+6);

%^^^^^^^
for current_x=xmin_theor:0.5:
    xmax_theor
for j=1:index
interpol = zeros(1,pol_degree
    +1);
for i=1:(pol_degree+1)
interpol(i) = type_data(j,i);
end %for j=1:index
temp_xmin = type_data(j,
    pol_degree + 5);
temp_xmax = type_data(j,
    pol_degree + 6);

if current_x >= temp_xmin &
    current_x <= temp_xmax
pol_val(j) = polyval(interpol
    ,current_x);
else
pol_val(j) = 0;
end
end % for j=1:index

if data_info(pol_degree+1) >
    interpol_a0(1)
interpol_val = pol_val(1) + (
    data_info(pol_degree+1) -
    interpol_a0(1))*(pol_val
    (1) - pol_val(2))/(
```

```
    interpol_a0(1) -
    interpol_a0(2));
%sprintf('ival=%d td =%d d1=%
    d d2=%d p1=%d p2=%d',
    interpol_val,theor_depth,
    interpol_depths(1),
    interpol_depths(2),pol_val
    (1),pol_val(2))
elseif data_info(pol_degree
    +1) < interpol_a0(index)
interpol_val = pol_val(index
    -1) + (data_info(
    pol_degree+1) -
    interpol_a0(index-1))*(
    pol_val(index-1) - pol_val
    (index))/(interpol_a0(
    index-1) - interpol_a0(
    index));
else
if g > 330 & g < 341
pol_X = polyfit(interpol_a0,
    pol_val,2);
else
pol_X = polyfit(interpol_a0,
    pol_val,3);
end
interpol_val = polyval(pol_X,
    data_info(pol_degree+1));

end %if data_info(pol_degree
    +1) > interpol_a0(1) |
    data_info(pol_degree+1) <
    interpol_a0(index)
pol_index = pol_index + 1;
X_vector(pol_index) =
    current_x;
Pol_vector(pol_index) =
    interpol_val;
clear pol_val;
end %for current_x=xmin_theor
    :0.5:xmax_theor

[xmin xmax exact_MDmax
    second_peak_index
    big_neg_inf_index] =
    limits4(X_vector,
    Pol_vector,g); %just to
    know 2nd peak or
    big_neg_inf
if second_peak_index == 0 &
    big_neg_inf_index == 0
temp_pol_degree = 4;
signal_type1 = 1;
else
temp_pol_degree = 12;
signal_type1 = 0;
end

ind_temp = 0;
for k=1:length(X_vector)
if X_vector(k) >= xmin &
    X_vector(k) <=xmax
ind_temp = ind_temp + 1;
final_X_vector(ind_temp) =
    X_vector(k);
final_Pol_vector(ind_temp) =
```

```
     Pol_vector(k);
end
end
final_interpol = polyfit(
    final_X_vector,
    final_Pol_vector,
    temp_pol_degree);

%TEST (centralizing signal)
for b=1:(temp_pol_degree+1)
MD_dertemp(b) = (
    temp_pol_degree - b + 1)*
    final_interpol(b);
end

exact_MDmax = 0;
exact_x = 0;
raizes = roots(MD_dertemp);
index_inf = 0;
for i=1:length(raizes)
if imag(raizes(i)) == 0
if polyval(final_interpol,
    raizes(i)) > exact_MDmax &
     abs(raizes(i)) <= 3
exact_MDmax = polyval(
    final_interpol,raizes(i));
exact_x = raizes(i);
end
end
end
final_X_vector =
    final_X_vector - exact_x;
final_interpol = polyfit(
    final_X_vector,
    final_Pol_vector,
    temp_pol_degree);

xmin = xmin - exact_x;
xmax = xmax - exact_x;

if g == 0
temp_pol_degree
final_interpol
MD_dertemp
raizes
exact_x
exact_MDmax
end
%TEST
clear MD_dertemp;

[xmin_theor xmax_theor
    exact_MDmax MDmax_pt
    second_peak_y_exact
    second_peak_MD_exact
    big_neg_inf_exact] =
    limits4b(final_interpol,g,
    xmin,xmax,temp_pol_degree,
    signal_type1);

%saving only the polynomial
    part of each curve:
dif_degrees = 8; %12 - 4

final_pol = zeros(1,
    pol_degree+1);
```

```
final_pol_ch2 = zeros(1,
    pol_degree+1);
for i=1:(pol_degree+1)
data_pol(i) = data_info(i);
if temp_pol_degree == 12
final_pol(i) = final_interpol
    (i);
else
if i <= dif_degrees
final_pol(i) = 0;
else
final_pol(i) = final_interpol
    (i - dif_degrees);
end
end

%calculating interpol signal
    norm:
norm = 0;
for i=1:(pol_degree+1)
norm = norm + final_pol(i)^2;
end
norm = sqrt(norm)/final_pol(
    pol_degree + 1);
%norm

data_xmin = data_info(
    pol_degree+5);
data_xmax = data_info(
    pol_degree+6);
final_xmin = xmin_theor;
final_xmax = xmax_theor;
mine = type_data(j,pol_degree
    +11);
target = type_data(j,
    pol_degree+12);
dif_sizes = abs(data_xmin -
    final_xmin) + abs(
    data_xmax - final_xmax);

%TEST (for controlling the
    limits during the ditance
    calculation)
if data_info(pol_degree + 7)
    == 0 & data_info(
    pol_degree + 8) == 0
data_xmin = xmin_theor;
data_xmax = xmax_theor;
end
%TEST

if pol_index > 0
clear X_vector;
clear Pol_vector;
clear X_vector_ch2;
clear Pol_vector_ch2;
clear final_X_vector;
clear final_Pol_vector;
clear final_X_vector_ch2;
clear final_Pol_vector_ch2;
end

if abs(data_xmin) <= 50 & abs
    (data_xmax) <= 50 & abs(
    final_xmin) <= 50 & abs(
    final_xmax) <= 50 &
```

```
    data_xmin < data_xmax &
    final_xmin < final_xmax %&
     dif_sizes <= 15 %& abs(
    data_xmin_ch2) <= 50 & abs
    (data_xmax_ch2) <= 50 &
    abs(final_xmin_ch2) <= 50
    & abs(final_xmax_ch2) <=
    50 & data_xmin_ch2 <
    data_xmax_ch2 &
    final_xmin_ch2 <
    final_xmax_ch2 &
    dif_sizes_ch2 <= 15
%now, we compare the curves:
distance = D_max_final(f,g,
    data_pol,data_xmin,
    data_xmax,final_pol,
    final_xmin,final_xmax);

total_distance = distance;

if total_distance <
    closest_curve
closest_curve =
    total_distance;
closest_depth = theor_depth;
closest_norm = abs(norm -
    data_info(pol_degree + 15)
    );

final_mine = mine;
final_target = target;
%
end
end
clear interpol;
clear interpol_ch2;

inserted_index =
    inserted_index + 1;
iError(f,inserted_index) =
    total_distance;
iIndex(f,inserted_index) =
    target;
iDepth(f,inserted_index) =
    theor_depth;

if inserted_index > 0
%Bubble sorting:
for k = 1:inserted_index
for l = 1:(inserted_index-1)
if iError(f,l) > iError(f,l
    +1)
temp_A = iError(f,l);
iError(f,l) = iError(f,l+1);
iError(f,l+1) = temp_A;

temp_B = iIndex(f,l);
iIndex(f,l) = iIndex(f,l+1);
iIndex(f,l+1) = temp_B;

temp_C = iDepth(f,l);
iDepth(f,l) = iDepth(f,l+1);
iDepth(f,l+1) = temp_C;
end
end
end
```

```
end

end %if theor_depth >= 4 &
    theor_depth <= 40
clear interpol_a0;
clear interpol_depths;
clear type_data;
clear interpol_a0_ch2;
clear interpol_depths_ch2;
clear type_data_ch2;

end % if type_data(1,
    pol_degree + 14) >=
    data_info(pol_degree + 14)
     & type_data(index,
    pol_degree + 14) <=
    data_info(pol_degree + 14)
else
g = g + 1;
end %if polarity is the same
end %while g <= num_files

if closest_curve ~= 999
ANS(f,1) = closest_curve;
ANS(f,2) = closest_depth;
for i=1:(pol_degree+1)
ANS(f,2+i) = closest_pol(i);
end
ANS(f,pol_degree+4) =
    final_mine;
ANS(f,pol_degree+5) =
    final_target;
ANS(f,pol_degree+6) =
    closest_norm;
end
end %for f=1:num_files

csvwrite('D:\ANS.csv',ANS);
csvwrite('D:\iERROR.csv',
    iError);
csvwrite('D:\iIndex.csv',
    iIndex);
csvwrite('D:\iDepth.csv',
    iDepth);

clear all
```

## A.3 Program for Comparing Polynomials in Database

```
function amkPPbuildtests =
    amkPPbuildtests
format long

name = strcat('D:step_errors.
    csv');

M1=csvread(name);

pol_degree = 12;
p = 1;
num_file = 362;

for ii=1:num_file
%first, lets keep the
    information of the
    reference line:

for j=1:(pol_degree+1)
coefs1(j) = M1(ii,j);
end

left1 = M1(ii,pol_degree+5);
right1 = M1(ii,pol_degree+6);
aux_roots1(1) = left1;
aux_roots1(2) = right1;
polarity_1 = M1(ii,pol_degree
    +9);
aoB_1 = M1(ii,pol_degree+12);
aoC_1 = M1(ii,pol_degree+1);
raizes1 = roots(coefs1);

num_roots1 = length(
    aux_roots1);
for a=1:length(raizes1)
if imag(raizes1(a)) == 0
if raizes1(a) <= right1 &
    raizes1(a) >= left1
num_roots1 = num_roots1 + 1;
ROOTS(ii,num_roots1) =
    raizes1(a);
aux_roots1(num_roots1) =
    raizes1(a);
end
end
end

%Bubble sorting:
if length(aux_roots1) > 0
for k = 1:length(aux_roots1)
for l = 1: (length(aux_roots1
    )-1)
if aux_roots1(l) > aux_roots1
    (l+1)
temp_A = aux_roots1(l);
aux_roots1(l) = aux_roots1(l
    +1);
aux_roots1(l+1) = temp_A;
end
```

```
end
end
end %if length

%finding MDmax of curve 1
for b=1:(pol_degree+1)
MD_dertemp1(b) = (pol_degree
   - b + 1)*coefs1(b);
end
raizes_der1 = roots(
   MD_dertemp1);
exact_x1 = 0;
exact_MDmax1 = 0;
for a=1:length(raizes_der1)
if imag(raizes_der1(a)) == 0
if abs(polyval(coefs1,
   raizes_der1(a))) >
   exact_MDmax1 & raizes_der1
   (a) <= right1 &
   raizes_der1(a) >= left1
exact_MDmax1 = abs(polyval(
   coefs1,raizes_der1(a)));
exact_x1 = raizes_der1(a);
end
end
end

if p == 0
Total1 = exact_MDmax1;
else %if p == 0
p_coefs1 = coefs1;
if p > 1
for k=1:(p-1)
p_coefs1 = conv(p_coefs1,
   coefs1);
end
end

int_p_coefs1 = polyint(
   p_coefs1);
total_error = 0;
for j=1:(length(aux_roots1)
   -1)
result = abs(polyval(
   int_p_coefs1,aux_roots1(j
   +1)) - polyval(
   int_p_coefs1,aux_roots1(j)
   ));
total_error = total_error +
   result;
end
Total1 = total_error^(1/p);
end %if p == 0
clear aux_roots1;

min_error = 999;

%now, lets compare to the
   remaining signals:
index_error3 = 0;
for i=1:num_file
if ii~=i
for j=1:(pol_degree+1)
coefs2(j) = M1(i,j);
end
```

```
left2 = M1(i,pol_degree+5);
right2 = M1(i,pol_degree+6);
polarity_2 = M1(i,pol_degree
    +9);
target_2 = M1(i,pol_degree
    +12);

aoB_2 = M1(i,pol_degree+12);
aoC_2 = M1(i,pol_degree+1);

target_depth = M1(i,
    pol_degree+10);

aux_roots2(1) = left2;
aux_roots2(2) = right2;

raizes2 = roots(coefs2);
num_roots2 = length(
    aux_roots2);
for a=1:length(raizes2)
if imag(raizes2(a)) == 0
if raizes2(a) <= right2 &
    raizes2(a) >= left2
num_roots2 = num_roots2 + 1;
ROOTS(i,num_roots2) = raizes2
    (a);
aux_roots2(num_roots2) =
    raizes2(a);
end
end
end

%Bubble sorting:
if length(aux_roots2) > 0
for k = 1:length(aux_roots2)
for l = 1: (length(aux_roots2
    )-1)
if aux_roots2(l) > aux_roots2
    (l+1)
temp_A = aux_roots2(l);
aux_roots2(l) = aux_roots2(l
    +1);
aux_roots2(l+1) = temp_A;
end
end
end
end %if length

%finding MDmax of curve 2
for b=1:(pol_degree+1)
MD_dertemp2(b) = (pol_degree
    - b + 1)*coefs2(b);
end
raizes_der2 = roots(
    MD_dertemp2);
exact_x2 = 0;
exact_MDmax2 = 0;
for a=1:length(raizes_der2)
if imag(raizes_der2(a)) == 0
if abs(polyval(coefs2,
    raizes_der2(a))) >
    exact_MDmax2 & raizes_der2
    (a) <= right2 &
    raizes_der2(a) >= left2
exact_MDmax2 = abs(polyval(
    coefs2,raizes_der2(a)));
```

```
exact_x2 = raizes_der2(a);
end
end
end

if p == 0
Total2 = exact_MDmax2;
else %if p == 0
p_coefs2 = coefs2;
if p > 1
for k=1:(p-1)
p_coefs2 = conv(p_coefs2,
   coefs2);
end
end

int_p_coefs2 = polyint(
   p_coefs2);
total_error = 0;
for j=1:(length(aux_roots2)
   -1)
result = abs(polyval(
   int_p_coefs2,aux_roots2(j
   +1)) - polyval(
   int_p_coefs2,aux_roots2(j)
   ));
total_error = total_error +
   result;
end
Total2 = total_error^(1/p);
end %if p == 0
clear aux_roots2;

%Finally, the difference
   between curves 1 and 2:
dif_coefs = coefs1 - coefs2;
lim_dif(1) = left1;
lim_dif(2) = right1;
lim_dif(3) = left2;
lim_dif(4) = right2;

%Bubble sorting:
if length(lim_dif) > 0
for k = 1:length(lim_dif)
for l = 1: (length(lim_dif)
   -1)
if lim_dif(l) > lim_dif(l+1)
temp_A = lim_dif(l);
lim_dif(l) = lim_dif(l+1);
lim_dif(l+1) = temp_A;
end
end
end
end %if length

if lim_dif(1) == left1
lim12 = coefs1;
elseif lim_dif(1) == left2
lim12 = coefs2;
end
if lim_dif(2) == left2 &
   lim_dif(3) == right2
lim23 = dif_coefs;
elseif lim_dif(2) == left2 &
   lim_dif(3) == right1
lim23 = dif_coefs;
```

```
elseif lim_dif(2) == left1 &
   lim_dif(3) == right1
lim23 = dif_coefs;
elseif lim_dif(2) == left1 &
   lim_dif(3) == right2
lim23 = dif_coefs;
elseif lim_dif(2) == right2 &
    lim_dif(3) == left1
lim23 = zeros(1,pol_degree+1)
   ;
elseif lim_dif(2) == right1 &
    lim_dif(3) == left2
lim23 = zeros(1,pol_degree+1)
   ;
end
if lim_dif(4) == right1
lim34 = coefs1;
elseif lim_dif(4) == right2
lim34 = coefs2;
end

aux_roots_dif(1) = left1;
aux_roots_dif(2) = right1;
aux_roots_dif(3) = left2;
aux_roots_dif(4) = right2;

%finding roots
raizes_dif = roots(dif_coefs)
   ;
num_roots_dif = length(
   aux_roots_dif);
for a=1:length(raizes_dif)
if imag(raizes_dif(a)) == 0
   if raizes_dif(a) <= lim_dif
      (4) & raizes_dif(a) >=
      lim_dif(1)
   num_roots_dif = num_roots_dif
       + 1;
   ROOTS(i,num_roots_dif) =
      raizes_dif(a);
   aux_roots_dif(num_roots_dif)
      = raizes_dif(a);
   end
   end
   end

   %Bubble sorting:
   if length(aux_roots_dif) > 0
   for k = 1:length(
      aux_roots_dif)
   for l = 1: (length(
      aux_roots_dif)-1)
   if aux_roots_dif(l) >
      aux_roots_dif(l+1)
   temp_A = aux_roots_dif(l);
   aux_roots_dif(l) =
      aux_roots_dif(l+1);
   aux_roots_dif(l+1) = temp_A;
   end
   end
   end
   end %if length

   %finding MDmax of curve dif
   for b=1:(pol_degree+1)
   MD_dertemp_dif_12(b) = (
```

```
    pol_degree - b + 1)*lim12(
    b);
MD_dertemp_dif_23(b) = (
    pol_degree - b + 1)*lim23(
    b);
MD_dertemp_dif_34(b) = (
    pol_degree - b + 1)*lim34(
    b);
end

raizes_der_dif_12 = roots(
    MD_dertemp_dif_12);
raizes_der_dif_23 = roots(
    MD_dertemp_dif_12);
raizes_der_dif_34 = roots(
    MD_dertemp_dif_12);

exact_MDmax_12 = 0;
for a=1:length(
    raizes_der_dif_12)
if imag(raizes_der_dif_12(a))
     == 0
if raizes_der_dif_12(a) >
    aux_roots_dif(1) &
    raizes_der_dif_12(a) <
    aux_roots_dif(2)
if abs(polyval(lim12,
    raizes_der_dif_12(a))) >
    exact_MDmax_12
exact_MDmax_12 = abs(polyval(
    lim12,raizes_der_dif_12(a)
    ));
end
end
end

exact_MDmax_23 = 0;
for a=1:length(
    raizes_der_dif_23)
if imag(raizes_der_dif_23(a))
     == 0
if raizes_der_dif_23(a) >
    aux_roots_dif(2) &
    raizes_der_dif_23(a) <
    aux_roots_dif(3)
if abs(polyval(lim23,
    raizes_der_dif_23(a))) >
    exact_MDmax_23
exact_MDmax_23 = abs(polyval(
    lim23,raizes_der_dif_23(a)
    ));
end
end
end
end

exact_MDmax_34 = 0;
for a=1:length(
    raizes_der_dif_34)
if imag(raizes_der_dif_34(a))
     == 0
if raizes_der_dif_34(a) >
    aux_roots_dif(3) &
    raizes_der_dif_34(a) <
    aux_roots_dif(4)
```

```
if abs(polyval(lim34,
    raizes_der_dif_34(a))) >
    exact_MDmax_34
exact_MDmax_34 = abs(polyval(
    lim34,raizes_der_dif_34(a)
    ));
end
end
end
end

%we still have to check the
    borders for maximums:
exact_MDmax_border2 = abs(
    polyval(lim23,
    aux_roots_dif(2)));
exact_MDmax_border3 = abs(
    polyval(lim23,
    aux_roots_dif(3)));

max_dif(1) = exact_MDmax_12;
max_dif(2) = exact_MDmax_23;
max_dif(3) = exact_MDmax_34;
max_dif(4) =
    exact_MDmax_border2;
max_dif(5) =
    exact_MDmax_border3;

%Bubble sorting:
if length(max_dif) > 0
for k = 1:length(max_dif)
for l = 1: (length(max_dif)
    -1)
if max_dif(l) < max_dif(l+1)
temp_A = max_dif(l);
max_dif(l) = max_dif(l+1);
max_dif(l+1) = temp_A;
end
end
end
end %if length
exact_MDmax_dif = max_dif(1);

if Total1 > Total2
Total_ref = Total1;
else
Total_ref = Total2;
end

if p == 0
Total = exact_MDmax_dif/
    Total_ref*100;
else %if p == 0
p_coefs_12 = lim12;
p_coefs_23 = lim23;
p_coefs_34 = lim34;

if p > 1
for k=1:(p-1)
p_coefs_12 = conv(p_coefs_12,
    lim12);
p_coefs_23 = conv(p_coefs_23,
    lim23);
p_coefs_34 = conv(p_coefs_34,
    lim34);
end
```

```
end

int_p_coefs_12 = polyint(
    p_coefs_12);
int_p_coefs_23 = polyint(
    p_coefs_23);
int_p_coefs_34 = polyint(
    p_coefs_34);

total_error = 0;
for j=1:(length(aux_roots_dif
    )-1)
if aux_roots_dif(j+1) >=
    lim_dif(1) & aux_roots_dif
    (j+1) <= lim_dif(2) &
    aux_roots_dif(j) >=
    lim_dif(1) & aux_roots_dif
    (j) <= lim_dif(2)
result = abs(polyval(
    int_p_coefs_12,
    aux_roots_dif(j+1)) -
    polyval(int_p_coefs_12,
    aux_roots_dif(j)));
total_error = total_error +
    result;
elseif aux_roots_dif(j+1) >=
    lim_dif(2) & aux_roots_dif
    (j+1) <= lim_dif(3) &
    aux_roots_dif(j) >=
    lim_dif(2) & aux_roots_dif
    (j) <= lim_dif(3)
result = abs(polyval(
    int_p_coefs_23,
    aux_roots_dif(j+1)) -
    polyval(int_p_coefs_23,
    aux_roots_dif(j)));
total_error = total_error +
    result;
elseif aux_roots_dif(j+1) >=
    lim_dif(3) & aux_roots_dif
    (j+1) <= lim_dif(4) &
    aux_roots_dif(j) >=
    lim_dif(3) & aux_roots_dif
    (j) <= lim_dif(4)
result = abs(polyval(
    int_p_coefs_34,
    aux_roots_dif(j+1)) -
    polyval(int_p_coefs_34,
    aux_roots_dif(j)));
total_error = total_error +
    result;
end
end
Total_dif = total_error^(1/p)
    ;
Total = Total_dif/Total_ref
    *100;
end %if p == 0

clear aux_roots_dif;

if Total == 0
Total = 999;
end

Total = Total;
```

```
ERROR(ii,i) = Total;

if Total < min_error
min_error = Total;
min_dm(ii,1) = M1(i,
    pol_degree+7);
min_dm(ii,2) =  M1(i,
    pol_degree+8);
min_dm(ii,3) =  M1(i,
    pol_degree+9);
min_dm(ii,4) =  i;
min_depth = target_depth;
end

if polarity_1 == polarity_2
index_error3 = index_error3 +
     1;
iError(ii,index_error3) =
    Total;
iIndex(ii,index_error3) =
    target_2;
iDepth(ii,index_error3) =
    target_depth;
end
end %if ii~=i
end %for i=1:num_file

if index_error3 > 0
for k = 1:index_error3
for l = 1: (index_error3-1)
if iError(ii,l) > iError(ii,l
    +1)
temp_A = iError(ii,l);
iError(ii,l) = iError(ii,l+1)
    ;
iError(ii,l+1) = temp_A;

temp_B = iIndex(ii,l);
iIndex(ii,l) = iIndex(ii,l+1)
    ;
iIndex(ii,l+1) = temp_B;

temp_C = iDepth(ii,l);
iDepth(ii,l) = iDepth(ii,l+1)
    ;
iDepth(ii,l+1) = temp_C;
end
end
end
end %if length
end %for ii=1:num_file

csvwrite('D:\ERROR.csv',ERROR
    );
csvwrite('D:\SERROR.csv',
    min_dm);
csvwrite('D:\iERROR.csv',
    iError);
csvwrite('D:\iIndex.csv',
    iIndex);
csvwrite('D:\iDepth.csv',
    iDepth);

clear all
```

## A.4 Program for Setting and Calculating FAR from Database

```
function amkPPbuildtests =
    amkPPbuildtests
format long

interpolated = 3; %0=d, 1=i,
    2=itop, X=noise

if interpolated == 1
name_input = strcat('D:\
    input_i.csv');
name_database = strcat('D:\
    database_potential_mines_i
    .csv');
threshold_maxE = 15;
elseif interpolated == 0
name_input = strcat('D:\
    input_itop2_d.csv');
name_database = strcat('D:\
    database_potential_mines_d2
    .csv'); %for 2nd sub JMA
threshold_maxE = 10;
elseif interpolated == 2
name_input = strcat('D:\
    input_i2.csv');
name_database = strcat('D:\
    database_potential_mines_noise
    .csv'); %MF and ITOP
threshold_maxE = 10;
else
name_input = strcat('D:\
    input_noise2.csv'); %no
    noise=remove database
    noise=remove database
name_database = strcat('D:\
    database_potential_mines_noise
    .csv');
threshold_maxE = 15;
end

% name_input = strcat('D:\
    input_itop.csv');

M_database=csvread(
    name_database);
M_input=csvread(name_input);

pol_degree = 12;
num_input = length(M_input
    (:,1)); %input length
num_database = length(
    M_database(:,1)); %
    database length

%building potential targets
    list
threshold_dE = 0; %2.9; %
    IMPORTATN!!!
potential_index = 0;
depth_margin = 4.0;
```

```
matrix_index=0;
for threshold_dE=0:5:100
potential_index = 0;

matrix_index = matrix_index +
    1;

%first, the mines we know:
for f=1:num_database
closest_MF_error = M_database
    (f,6); %F
closest_Mine_error =
    M_database(f,9); %I
target_ID = M_database(f,1);
    %A
target_depth = M_database(f
    ,3); %C
target_type = M_database(f,2)
    ; %B
closest_MF_type = M_database(
    f,4); %D
closest_MF_depth = M_database
    (f,5); %E

dE = closest_Mine_error -
    closest_MF_error;

%1) REAL mines:
if target_type >= 70
potential_index =
    potential_index + 1;
potential_mines_type(
    potential_index) =
    target_type;
potential_mines_depth(
    potential_index) =
    target_depth; %actually
    doesnt matter depth if its
     already a mine
end

%2) Mines closer to MFs than
    MFs:
if target_type < 70
if closest_Mine_error <
    closest_MF_error
potential_index =
    potential_index + 1;
potential_mines_type(
    potential_index) =
    target_type;
potential_mines_depth(
    potential_index) =
    target_depth; %actually
    doesnt matter depth if its
     already a mine

%new for super safe
    (09052013)
potential_index =
    potential_index + 1;
potential_mines_type(
    potential_index) =
    closest_MF_type;
potential_mines_depth(
    potential_index) =
```

```
    closest_MF_depth; %
    actually doesnt matter
    depth if its already a
    mine
end
end

%3) MFs closer to Mines than
    Mines:
if target_type >= 70
if closest_Mine_error >
    closest_MF_error
potential_index =
    potential_index + 1;
potential_mines_type(
    potential_index) =
    closest_MF_type;
potential_mines_depth(
    potential_index) =
    closest_MF_depth; %
    actually doesnt matter
    depth if its already a
    mine
end
end

%4) dE criteria
if target_ID > 305 %we know
    mines data start from 306
dE = -dE; %closest_MF_error -
     closest_Mine_error
end
if dE <= threshold_dE
potential_index =
    potential_index + 1;
potential_mines_type(
    potential_index) =
    target_type;
potential_mines_depth(
    potential_index) =
    target_depth; %actually
    doesnt matter depth if its
     already a mine

%new for super safe
    (09052013)
potential_index =
    potential_index + 1;
potential_mines_type(
    potential_index) =
    closest_MF_type;
potential_mines_depth(
    potential_index) =
    closest_MF_depth; %
    actually doesnt matter
    depth if its already a
    mine
end
end %for f=1:num_database

%now lets check the results:
for f=1:num_input%num_files
%input_ID = M_input(f,1);
%input_type = M_input(f,2); %
    dont use... cause its the
    answer
```

```
input_error = M_input(f,3);
input_closest_target =
   M_input(f,4);
input_depth = M_input(f,5);

final_discrimination = 0;

if input_closest_target >= 70
final_discrimination = 1;
else
if input_error >=
   threshold_maxE
final_discrimination = 1;
else
for j=1:potential_index
if input_closest_target ==
   potential_mines_type(j)
if input_depth <= (
   potential_mines_depth(j) +
    depth_margin) &
   input_depth >= (
   potential_mines_depth(j) -
    depth_margin)
final_discrimination = 1;
break;
end
end
end %for j=1:potential_index
end
end
MAP_discrimination(f,
   matrix_index) =
   final_discrimination; %+1
   since 0 is not allowed
end %for f=1:num_input
clear potential_mines_type;
clear potential_mines_depth;
end %for threshold_dE
```

## A.5 Calculating the Median

```
Bubble_sort(MD);

//Calculating the median:
 int size = MD.size();
 int rest = size%2;
 index = size/2; //the
    intermediary element
//in the vector
 int median;

 if(size != 0){
  if(rest == 0){//size is
     even
   median = (MD[index] + MD[
      index-1])/2;
  }
  else if(rest == 1){//size
     is odd
   median = MD[index];
  }
 }
```

## A.6 Calculating the Half-peaks

```
//first we store all peaks
peaks_index_searching_algorithm
    ();

int fp,lp;//first and last
    points

for(int p=0;p<=peaks_index.
    size();p++){
//finding the initial (A1,B1)
     and final (A2,B2)
//points for substituting in
    the
//line equation y = ax + b
 if(p == 0){
  fp = 0;
  lp = peaks_index[0];
  }
 else if(p == peaks_index.
     size()){
  fp = peaks_index[
      peaks_index.size()-1];
  lp = MD.size()-1;
  }
 else{
  fp = peaks_index[p-1];
  lp = peaks_index[p];
  }

 A1 = X[fp];
 B1 = MD[fp];
 A2 = X[lp];
 B2 = MD[lp];

 if(A2==A1)
  a = 0;
 else
  a = (B2 - B1)/(A2 - A1);

 b = B1 - A1 * a;
 for(int m=fp;m<lp;m++){
  new_MD.push_back(a*X[m]+b)
      ;
 }
}//for peaks_index
```

## A.7 Potential Targets Searching Algorithm

```
//erasing noise
for(int i=0;i<MD.size();i++){
 if(neg_threshould<MD[i] &&
MD[i]<pos_threshould){
  MD[i] = 0;
 }
}

//while there are small
    points remaining:
```

```
while(total1 != 0){
 total1 = 0;
 int m;
 for(int i=0;i<MD.size();i++)
    {
  if(MD[m] != 0){
   m = i;
   i = i+1;
   while(i<MD.size() && MD[i
       ]!=0){
    i = i + 1;
   }
   total1 = (i-m);
   if(total1<=Target_Size){
    for(int k=m;k<i;k++){
     index.push_back(k);
    }
   }
   else
    total1 = 0;
  }
}

 for(int g=0;g<index.size();g
    ++){
  MD[index[g]] = 0; //erasing
      points
 }
}
```

## A.8 Local Maximum Searching Algorithm

```
for(int i=0;i<MD.size();i++){
 int max_MD = abs(MD[i]);
 int max_index, max_before;
 real dist;

 max_before = 0;
 max_index = 0;
 int first = 0;

 for(int j=0;j<MD.size();j++)
    {
  dist = sqrt((Xi - Xj)^2+(Yi
      - Yj)^2);
  if(dist < radius){
   if(abs(MD[j]) >= max_MD){
        max_MD = abs(MD[j]);
        max_index = j;
        first = 1; //first
            center point
       }
  }
 }//for j

 while(max_before !=
     max_index){
  max_before = max_index;
  for(int j=0;j<MD.size();j
     ++){
```

```
   dist = sqrt((Xi - Xj)^2+(
      Yi - Yj)^2);
   if(dist < radius){
    if(abs(MD[j]) >= max_MD){
          max_MD = abs(MD[j])
              ;
          max_index = j;
         }
   }
  }//for j
}//while

//check if the point hasnt
   been stored yet
if(first == 1){
 int similar = 0;
 for(int j=0;j<peaks_index.
    size();j++){
  if(peaks_index[j] ==
     max_index)
  similar = similar + 1;
 }
 //if not, save it
 if(similar == 0){
  if(stream[max_index]!=0){
   peaks_index.push_back(
      max_index);
  }
 }
}//if first
}// for i
```

# Appendix B

# Achievements

## B.1 Papers in refereed journals

1. A. M. Kaneko, M. Marino and E. F. Fukushima. "New Artificial Vision Technique for Terrain Mapping Using Stereo Vision Under Extreme Sunlight Conditions in Demining Operations". *Journal of Advanced Robotics*. 2013. **(under review)**.

2. A. M. Kaneko, G. Endo and E. F. Fukushima. "Proposal of a Discrimination Method for Landmines and Metal Fragments Using Metal Detectors". *Journal of Mine Action*. 2013. **(to be published in March, 2014)**

3. A. M. Kaneko and E. F. Fukushima. "Development of an Automatic Landmine Detection and Marking System for the Demining Robot Gryphon". *Journal of Advanced Computational Intelligence and Intelligent Informatics*. Vol. 15 No. 6, pages 737-743, 2011.

## B.2 Conferences

### B.2.1 Refereed

1. J. Li, A. M. Kaneko and E. F. Fukushima. "Basic Methods for Humanitarian Demining Robot Manipulator Self-Calibration Using Stereo Vision Camera",

*IEEE/RSJ Int. Conf. on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014. **(under review)**

2. A. M. Kaneko, G. Endo and E. F. Fukushima. "Landmine Buried Depth Estimation by Curve Characterization of Metal Mine Detector Signals", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013.

3. A. M. Kaneko, M. Marino and E. F. Fukushima. "Humanitarian Demining Robot Gryphon: New Vision Techniques and Optimization Methods", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

4. A. M. Kaneko and E. F. Fukushima. "Basic Studies on Computer-Aided Marking Task Operation for the Humanitarian Demining Robot Gryphon", *Int. Symposium on Intelligent Systems (iFAN)*, Tokyo, Japan, September 2010.

### B.2.2 Non-refereed

1. A. M. Kaneko, G. Endo and E. F. Fukushima. "Noise Influence Analysis in Landmine Discrimination by Curve Characterization Method". *In proceedings of The 31st Annual Conference of the Robotics Society of Japan*, Tokyo, Japan, September 2013.

2. A. M. Kaneko and E. F. Fukushima. "Humanitarian Demining Robot Gryphon - Mine Detector Scanning Procedure Optimization". 1A1-R15. *JSME Robomec*, Tsukuba, Japan. 2013.

3. J. Li, A. M. Kaneko and E. F. Fukushima. "Humanitarian Demining Robot Gryphon - Self-Calibration Using Stereo Vision Camera". 1A1-R11. *JSME Robomec*, Tsukuba, Japan. 2013.

4. A. M. Kaneko and E. F. Fukushima. "Metal Detector Signal Characterization for Landmine Discrimination". 1P1-C10. *JSME Robomec*, Hamamatsu,

Japan. 2012.

5. J. Baumann, A. M. Kaneko and E. F. Fukushima. "GPS Based Autonomous Point-to-point Driving of All Terrain Vehicle Gryphon". *In proceedings of The 29th Annual Conference of the Robotics Society of Japan*, Tokyo, Japan, September 2011.