TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Automated anomaly detection in geophysical survey

Doctoral Thesis

Study programme:	3901 – Applied sciences in engineering
Study branch:	3901V055 – Applied sciences in engineering
Author:	**Ing. Lenka Kosková Třísková**
Supervisor:	Ing. Josef Novák, Ph.D.

Liberec 2017

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Automatická detekce anomálií při geofyzikálním průzkumu

Disertační práce

Studijní program:	3901 – Aplikované vědy v inženýrství
Studijní obor:	3901V055 – Aplikované vědy v inženýrství
Autor práce:	**Ing. Lenka Kosková Třísková**
Vedoucí práce:	Ing. Josef Novák, Ph.D.

Liberec 2017

Declaration

I hereby certify that I have been informed that Act 121/2000, the Copyright Act of the Czech Republic, namely Section 60, Schoolwork, applies to my dissertation in full scope. I acknowledge that the Technical University of Liberec (TUL) does not infringe my copyrights by using my dissertation for TUL's internal purposes.

I am aware of my obligation to inform TUL on having used or licensed to use my dissertation in which event TUL may require compensation of costs incurred in creating the work at up to their actual amount.

I have written my dissertation myself using literature listed therein and consulting it with my supervisor and my tutor.

I hereby also declare that the hard copy of my dissertation is identical with its electronic form as saved at the IS STAG portal.

Date:

Signature:

Abstract

The study *Automated anomaly detection in geophysical survey* is the application of machine learning and computer vision techniques to the geophysical data. The two main applications were tested during the research. The research is mainly focused on the surface geophysics. The fast scanning of an area for an appearance of a set of predefined anomalies is the main focus of the thesis. The research was applied to potential fields. Three types of detection were tested: image processing techniques, the supported machine learning with classifiers and adaptive neural networks. The second application mentioned in the thesis is the application of the research results to a continuous monitoring process. The structure of the object is known and all the significant temporal changes in the data are to be detected and interpreted. The thesis gives a summary of the state of the research on the selected topic. It includes a proposal of the algorithms and it summarizes the achieved results.

Keywords: Geophysics, Potenital fields, Seismics, Computer Vision, Machine Learning

Abstrakt

Práce nazvaná *Automatická detekce anomálií při geofyzikálním průzkmu* je aplikací metod strojového učení a počítačového vidění v oblasti zpacování gyofyzikálních data. Během výzkumu byly testovány dvě možné aplikace. Výzkum je zaměřen hlavně na průzkum oblasti blízko povrchu s cílem detekovat výskyt předem definovaných anomálií. Výzkum byl aplikován v oblasti potenciálových polí, testovány byly tři možné typy detekce: počítačové vidění, metody asistovaného učení s klasifikátory a adaptivní neuronové sítě. Druhou aplikací výzkumu zmiňovanou v práci byla aplikace výsledků výkumu na průběžné monitorování. Struktura monitorovaného objektu je známa a jakékoliv významné změny v datech musí být detekovány a interpretovány. Práce poskytuje shrnutí stávajícího výzkumu ve zvolených oblastech, návrh algoritmů a shrnuje výsledky výzkumu.

Klíčová slova: Geofyzika, Potenciálová pole, Seismika, Počítačové vidění, Strojové učení

Acknowledgements

I would like to thank my supervisor Ing. Josef Novák, Ph.D. for all his support, knowledge, kind feedback and patience.

I would like to thank my colleagues for their feedback, cooperation and of course friendship. In addition I would like to express my gratitude to the Mgr. Jiří Vraný, Ph. D., for his support and work done during the implementation of the adaptive neural network.

I would like to thank my all family members for their patience and support.

Contents

List of Figures

List of Tables

List of abbreviations

ANN	Adaptive Neural Network
ConvNet	Convolutional Neural Network
CV	Computer vision
ERT	Electric Resistivity Tomography
G	Gravitational Constant ($G = 6.67 \cdot 10^{-11} Nm^2kg^{-2}$)
GPR	Ground Penetrating Radar
GUI	Graphical User Interface
IP	Induced polarization
kNN, KNN	K Nearest Neigbour computing
MLT	Machine Learning Tehcniques
SNR	Signal to Noise Ratio
SVM	Support Vector Machine
SML	Supervised Machine Learning
UXO	Unexploded Ordnance

1 Aims of the Thesis

Any geophysical survey ends up with a set of data describing the properties of the materials hidden under the Earth surface. The acquired data set has to be analyzed and interpreted; such complex process requires specialist with knowledge of the geophysical theory as well as with a lot of practical experience. In general, such process cannot be replaced by any automated software in general. But subtasks can be defined where the automated or semi-automated data preprocessing can be helpful and useful. The main objective of the presented work is the research and development of algorithms dedicated to semi-automated interpretation of the geophysical data.

The work was originally inspired by the idea to speed up the recovery operation after a disaster such as flooding or an earthquake. In the case of disease and recovery operation, it is necessary to detect the cavities or other contrast bodies under the surface. The fast detection of buried infrastructure networks such as electricity or gas pipes is also very important. After a disaster another danger situation can occur: the area can be endangered by landslides, the stability of dams can be impaired. Such structures can be detected using geophysical methodology. Of course - the geophysical survey and data interpretation requires a fully qualified specialist and rescue team members typically have no experience with geophysics at all.

Geophysical survey can help to scan subsurface in such situation, but it is necessary to preselect the method and predefine the methodology. The members of the rescue team cannot directly use the geophysical measuring equipment without any training. They should be trained for a defined set of predefined data acquiring procedures with preselected methodology and tools. Regardless of the selected methodology the acquired data must be interpreted. Such knowledge is far behind the capabilities of the professional rescuers. Fortunately the fast scan of the near surface concentrated to find significant predefined anomalies in the affected area does not require to define a detailed model of the subsurface. During the fast scan of the area, several typical questions have to be answered, such as what is the probability of appearance of this anomaly in selected area. It means that a semi automated fast scan of the data might be done in situ before the fully qualified data interpretation to speed up the whole process.

So before any special methodology for the rescue team is selected or any special equipment designed, the question is if the semiautomated data preanalysis is applicable. The first part of this thesis is dedicated to proposal and test of such fast scanner. The work is focused to the potential field data with special focus to the gravity data. Several types of hidden bodies were preselected and synthetical data sets were created. To detect the anomaly structure in the data, computer vision techniques were used at first do detect the anomaly presence in the data. Noise tolerance was tested as well with several noise models. The original data set was sampled, thresholded, converted to sets of black and white images and scanned for structures which typically appear in the data if anomaly body is presented under the surface. The position of the structures and its size and shape were used to estimate the anomaly type and its parameters.

For an initial study of the fast scan algorithm were selected the methods based on potential field models (gravity, electrical polarization). The second part of the work presents another study focused to application of fast scan algorithms to the process of the nuclear waste repository monitoring. In the case of disposal, the key issue is a very long term monitoring of the conditions of the repository. When suitable monitoring process is still the question of the research, the geophysical methods in general should be taken in focus. In general, geophysics offers non-invasive monitoring methods of the physical processes running in the repository. Regardless of the finally selected methodology and monitoring procedure, the data interpretation means to detect significant temporal changes or anomaly in the data. Machine learning methods and structure detection algorithms can be used as a useful support method for the classical geophysical data interpretation. The algorithm designed for the potential field data was updated for the seismical models of the repository.

Regardless of the monitoring technology, the physical conditions in the repository such as water saturation or temperature should either remain unchanged or change in a known manner. If any difference in monitored data is captured, it is necessary to identify the cause of the change. Physical parameters in the repository can slightly oscillate around the equilibrium, which can be understood as a normal behaviour, or they can more dramatically increase/decrease. Such situation can be sign of a problem in the repository – for example the surrounding barrier may be corrupted and safety of the repository can be endangered.

The repository itself is strictly defined – it is a structure with defined and well known geometry, with stable homogeneous surrounding. It is possible to start pre monitoring to get the stable data stream as a reference training set. The other training set of the data can be a set of models of anomaly data which correspond to predefined problems occurring in the repository (increasing temperature over the prediction, modified water saturation, modified geometry etc.). The task is to scan in the data for any similarity with predefined anomaly situations.

The following chapters of the presented work summarize step by step the design of the algorithm and the tests. In all the experiments were used the synthetical data as the work is a first part of the research. The main aim of the thesis was to study the computer vision and machine learning techniques and test its aplicability in the geophysical data processing context. The chapter Detailed Task Definition describes, how the anomalies were modelled and what types of data were selected. Initial part of the Chapter Anomaly detection implementation stands for the current state of art review for all the technologies used in the research. The chapter shortly summarizes the current applications of the machine learning techniques in the geophysical data processing and also shortly describes the algorithm used for the feature extractions and machine learning techniques. The chapter The Achievements gives a description of all proposed algorithms and obtained results. The main focus is on the potential field data, the data sets for the nuclear waste repository model are kept as an illustration of another algorithm application.

The adaptation of the algorithm and all the tests for the seismical data were realized with the support of the Modern2020 project[1] (Work package 3, Task 3.5). This project has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 662177. The overall objective of the Modern2020 Project is to provide the means for developing and implementing an effective and efficient repository operational monitoring programme, that will be driven by safety case needs, and that will take into account the requirements of specific national contexts (including inventory, host rocks, repository concepts and regulations, all of which differ between Member States) and public stakeholder expectations (particularly those of local public stakeholders at (potential) disposal sites).

[1] http://www.modern2020.eu

2 Detailed Task Definition

2.1 Near surface fast scan

Gravity anomaly is a deviation of observed gravity from gravity predicted for the location from a model of Earth gravity field. The gravity is usually measured in the units of acceleration. The gravity anomaly value is typically smaller than values of gravity itself. The gravity field can be measured using high resolution, with grid step measured in kilometers to detect densities located deep below the Earth surface. As the grid goes more granular it is possible to identify anomalies located closer to the surface.

The gravity anomaly indicates different density of materials under the surface. In target application it is usable to detect heavy objects or cavities with density contrast[1]. This detection can be used for example for fast dam diagnostics or landslides danger detection.

The acquired gravity does not reflect only the geological sources. The measured gravity value is always influenced by tidal forces, altitude and terrain topography. Therefore it is necessary to apply all the gravity standard corrections such as Bourger correction or free air correction before the proposed algorithm is used. A priori information including the information about known anomalies in the neighborhood or deep subsurface (such as location of buildings, constructions, water resources or subway) can help to pre-process acquired data and fast up the detection process.

2.1.1 Gravity Anomaly Forward Models

Gravity effect of any object is proportional to object's density. Considering the body of defined volume and density ρ with corresponding gravitational potential V and its vertical component V_z is expressed in Equation 2.1 (quoted from [25]). G is the gravitational constant ($G = 6.67 \cdot 10^{-11} Nm^2kg^{-2}$).

$$V = G\int_{\tau}\frac{\rho}{r}\mathrm{d}\tau \quad V_z = \frac{\partial V}{\partial z} \tag{2.1}$$

The Equation (2.1) can be used to deduce the horizontal component of gravity effect of an object with defined geometry. The analytical field description derived

[1] The density contrast is the difference of anomaly density and the density of the surrouding material.

from Equation (2.1) for such bodies are listed in all texts focused to the gravitational field theory (for example in [25], [5], or [32]).

For the algorithm design and tests, simple geometrical bodies were selected: a sphere, a horizontal infinite cylinder, a vertical semi infinite cylinder. As it was already declared, during the fast subsurface scan it is not important to define precisely the anomaly geometry. Important is to quickly assess whether in the area any anomaly is present and if yes, where and how deep it is located and what is estimate density contrast.

A general function describing a symmetric potential field anomaly can be expressed by following equation (cited from [33]):

$$f(r) = \frac{F}{(r^2 + z^2)^q} \tag{2.2}$$

Anomaly type	**F**	**q**	**M**
Sphere	GMz	$\frac{3}{2}$	$\frac{4}{3}\pi R^3 \rho$
Horizontal Cylinder	$2GMz$	1	$2\pi R^2 \rho_c$
Vertical Cylinder	GM	$\frac{1}{2}$	$2\pi R^2 \rho_c$

Table 2.1: The F and q factor for simple geometrical bodies, gravity field. The G is the gravitational constant, M is the mass for the sphere and density contrast times cross-sectional area for the cylinder, z is the depth of the anomaly.

In the Equation 2.2, the F is an amplitude factor, the q is a shape factor characterizing the shape of the anomaly. The r is the distance from the middle point of the anomaly to the observation point on the surface. Detailed summary of q and F values for different simple geometrical bodies both for gravity and magnetic sources is given for example in [33], and it is listed in Table 2.1. Parameters listed in the Table 2.1 were used to compute the test data set. The figures 2.1, 2.2 and 2.3 show the meaining of the pamameters listed in the table for all the anomaly bodies.

All the data used in the simulations were generated by the script *get_data.py* which is attached to the thesis (see Chapter 7 for details). The script uses the parameters identification as it is depicted in the pictures. The input parameters are marked by red color in the figures. For the spherical anomaly it is a set of central point coordinates $[XPos, YPos, ZPos]$. The total mass M is given by the radius R and the density contrast ρ. The Rn is used to compute the field value in the surface point $[Xn, Yn]$. It corresponds with the r parameter from equation 2.2. The vertical cylinder body has the same set of parameters, the $ZPos$ value is the depth of the cylinder top plane. The horizontal cylinder is an infinite body located parallel to the surface plane, the central line of the cylinder is given by two points with coordinates $[XPos, YPos]$, $[XPos2, YPos2]$. The depth $ZPos$ is the depth of the central line. The density contrast for the both horizontal and vertical cylinder is marked as ρ_c as it is given as a density per 1 m.

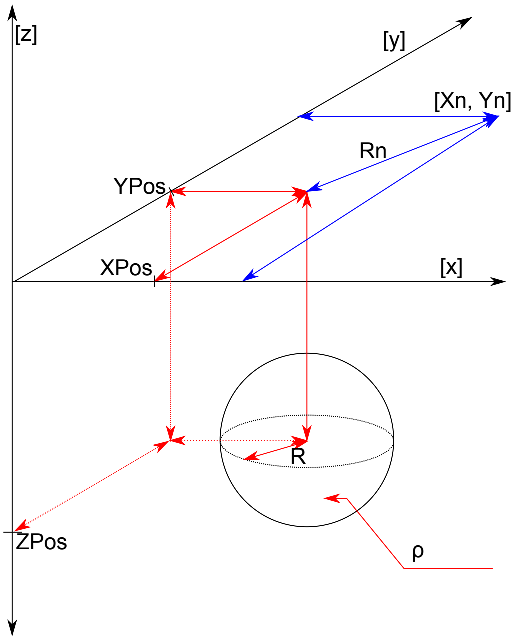


Figure 2.1: The spherical anomaly model.

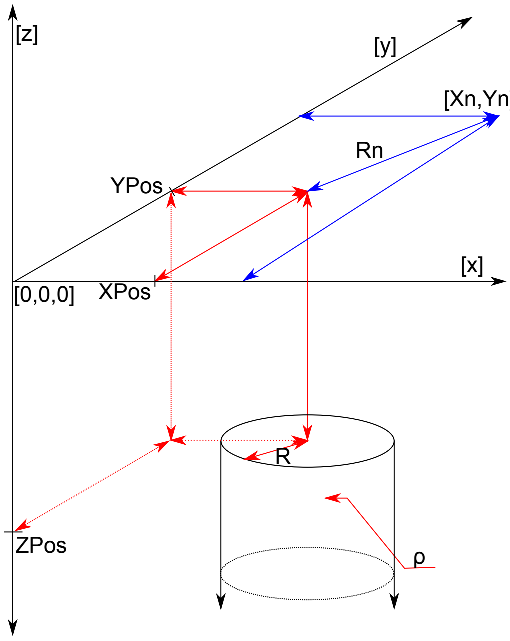


Figure 2.2: The semi infinite vertical cylinder anomaly model.

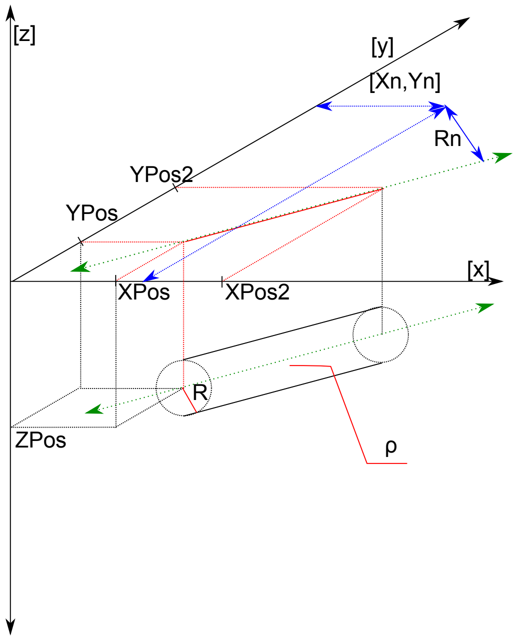


Figure 2.3: The infinite horizontal anomaly model.

The rectangular prism model presented in the Equation 2.3 was used according to the [5], pages 192–213. The rectangular prism model is used to illustrate that during the fast scan it is not important to define precisely the anomaly body geometry. If the prism is detected as a sphere, still we can estimate the depth and density contrast. All the presented anomaly models are defined for ideally smooth surface, homogenized surrounding subsoil and constant density contrast in the whole anomaly volume. The prism is defined by its top left corner with coordinates $[XPos, YPos, ZPos]$ and the down right corner $[XPos2, YPos2, ZPos2]$ and it have homogeneous density contrast ρ as it is demonstrated in the Figure 2.4.

$$V_z = G\rho \int_{x_1}^{x2} \int_{y1}^{y2} \int_{z1}^{z2} \frac{z}{(x^2+y^2+z^2)^{\frac{3}{2}}} \tag{2.3}$$

$$V_z = G\rho \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} \mu_{ijk} \left[z_k arctan \frac{x_i y_j}{z_k R_{ijk}} - x_i ln(R_{ijk} + y_j) - y_j ln(R_{ijk} + x_i) \right]$$

$$R_{ijk} = \sqrt{x_i^2 + y_j^2 + z_k^2}$$

$$\mu_{ijk} = (-1)^i (-1)^j (-1)^k$$

For the more complex anomaly body, we can see models based on the collections of the rectangular prisms, rectangular blocks, laminas and similar regular bodies (for details see [25] and [5]).

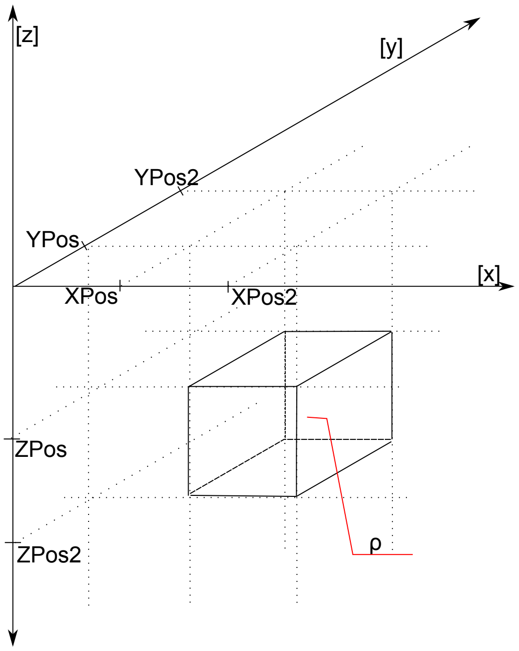


Figure 2.4: The rectangular prism anomaly model.

The density contrast of the anomaly is modeled according to the target application. The possible scenario is an anomaly body filled by the air, water or a construction material such as debris. The surrounding subsoil can in general consist of any material. The Table 2.2 lists the combinations of the most likely combinations of densities for the anomaly and surrounding subsoil. The densities used in the table were used according to [22] and [6].

For the initial modeling, it is not necessary to model all of the density combinations listed in Table 2.2. Figure 2.5 shows a distribution of values listed in the Table 2.2. A set of groups can be seen in the picture. The main sets of the test data used the small positive density contrast value set to 1 gcm^{-3}. This model stands for the anomaly created by a rock or concrete. The density value is not the most important parameter of the model, in all the models it just stands as a multiplying factor and its value has no influence to the shape of the field. The shape of the field is affected by the position parameters and by the anomaly type in general. Figures 2.6, 2.7, 2.8 and 2.9 illustrate all types of predefined anomalies, all the bodies are located ideally in the middle of the area.

The synthetical models were created for the area of size 100 × 100 m. The real data set is acquired over several linear profiles. The field workers are passing through the area following approximately the linear path, each such path corresponds with one profile. The set of profiles is obtained by repeated measurement at the locality. Acquired data can be interpolated into a rectangular network and such interpolation called a gridding process is a standard part of the commercial geophysical software such as Oasis Montaj ([28]). The gridding algorithms and the gravimetry data corrections (terrain corrections, Bourger corrections etc.) are not part of this study. The algorithm input consits of already gridded data with necessary corrections.

Anomaly material	**Anomaly density** $[gcm^{-3}]$	**Surrounding matter**	**Surrounding density** $[gcm^{-3}]$	**Density contrast** $[gcm^{-3}]$
Air	0.0	Loam	1.7	-1.7
		Mudstone (claystone, marlstone)	2.0	-2.0
		Sedimentary rock (limestone)	2.3	-2.3
		Volcanic rock (basalts)	3.15	-3.15
		Concrete (compact concrete with steel reinforcement)	2.5	-2.5
		Rubble	1.3	-1.3
Water	1.0	Light rocks	2.5	-1.5
		Heavy rocks	3	-2.0
		Soil (loam)	1.7	-0.7
Concrete	2.5	Light rocks	2.5	0.0
		Heavy rocks	3	-0.5
		Soil (loam)	1.7	0.8
		Rubble	1.3	1.2
		Gravel and sand	1.0	1.5

Table 2.2: The expected anomaly density contrasts in the real application.

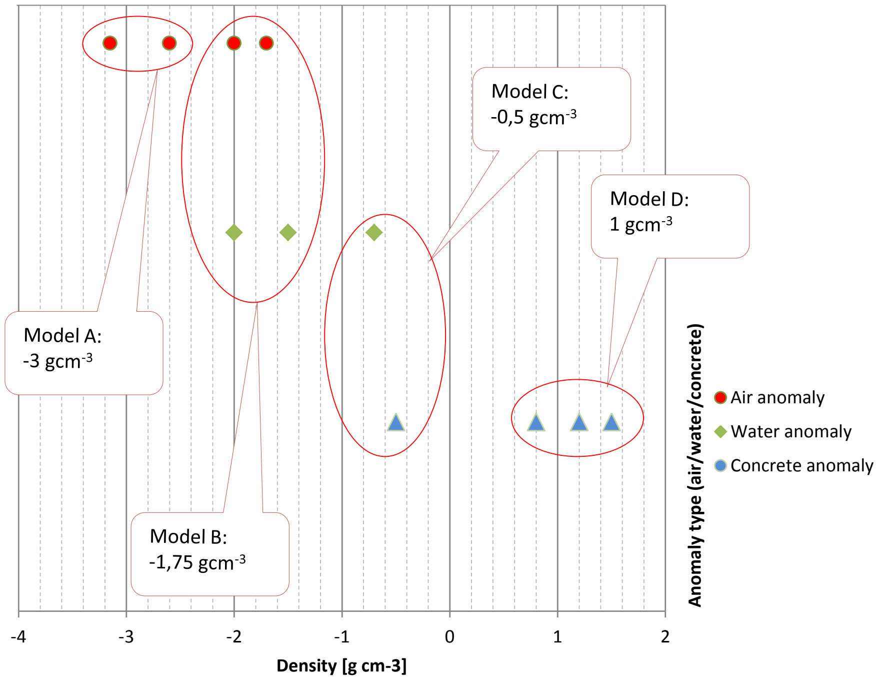


Figure 2.5: The groups of the density models defined according to the target application. The shape of the field is not affected by the density value, the density stands in all the equations as a multiplying factor. Therefore the group D was finally selected for all the models.

Initial algorithm tests were done with synthetical data containing only one model of the anomaly, the other data set contained smooth data with the noise (the selection of the noise model is explained later in this chapter). The main data set was created by examples spherical bodies and both types of cylinders. The density was set to constant value, the spatial parameters were randomly generated.

For each type of anomaly a set of randomly generated examples was used. A reference data set was generated also with the noise, with SNR 20 dB and 40 dB. Described data sets were also used to train and test the artificial neural network (ANN) and the classifiers. Table 2.3 summarizes the intervals of initial parameters values used to generate the random data. The values of $XPos, YPos, XPos2, YPos2$ were set randomly from $0 - 100$ to cover all the area. In the case of rectangular prism it was always selected to have $XPos < XPos2$, $YPos < YPos2$ and $ZPos < ZPos2$.

The initial testing of the fast scan algorithm was done using a smaller data set of 30, 100 and 1000 samples of each anomaly body. The final testing was done with the data set of 1000 examples. The ANN was trained using 10000 examples and tested with another set of the same size.

Anomaly type	**R**	**ZPos**	**ZPos2**
Sphere	1 m - 20 m	1m - 50 m	Not used
Horizontal Cylinder	1 m - 20 m	1m - 50 m	Not used
Vertical Cylinder	1 m - 20 m	1m - 50 m	Not used
Rectangular prism	Not used	1m - 50 m	1m - 50 m

Table 2.3: The definition intervals for anomaly data sets used to generate the train and test data.

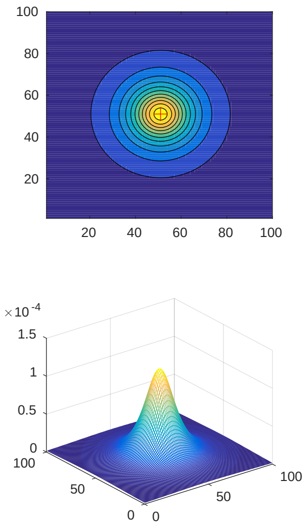


Figure 2.6: The example of used input data, spherical anomaly, density contrast 1 gcm^{-3}, radius 5 m, situated in the middle of the area, located 15 m under the surface.

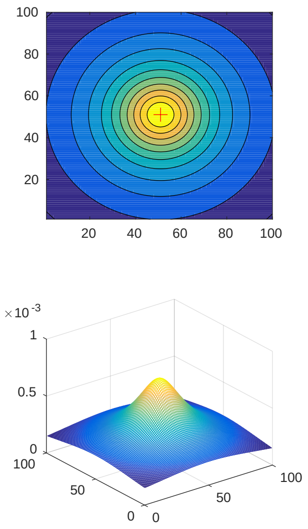


Figure 2.7: A model of vertical cylinder, density contrast 1 gcm^{-3}, radius 5 m, situated in the middle of the area, located 15 m under the surface.

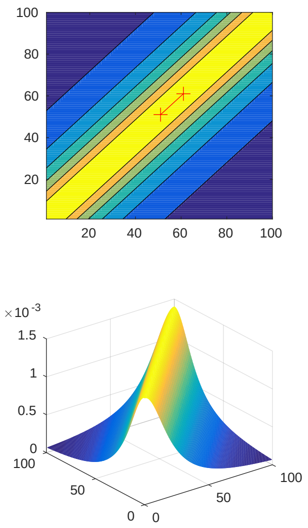


Figure 2.8: A model of horizontal cylinder, density contrast 1 gcm^{-3}, parallel with the surface, running diagonally, radius 5 m, located 15 m under the surface.

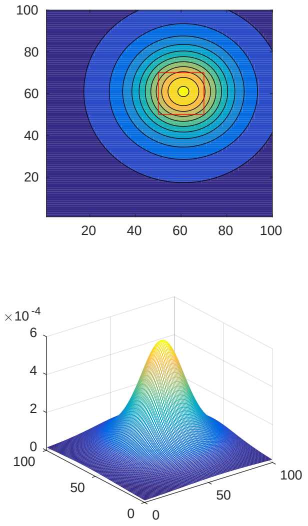


Figure 2.9: A model of rectagonal prism, width is 10 m, height is 20 m, located 15 m under the surface.

2.1.2 The noise models

As any other real data, the real geophysical data can contain a noise. The source and the nature of the noise depends on the data acquiring methodology. The clean anomaly picture in the data can be overshadowed by the influence of other source bodies. Typically, a set of corrections are applied to data before the data are analyzed (such as free-air, Bourger, terrain or building correction for gravity data). Such correction is a standard, well described, widely used procedure, which is often automated or semi-automated. Therefore in this text it is assumed that corrections were already applied to the data.

Another source of the noise is the noise of the measuring equipment itself, the random or systematical errors or the noise of the surrounding environment (swell noise in seismics for example [9]). To test the resistance of the algorithm to the noise in the data, the analytical signals used in this thesis were combined with white noise. The white noise was selected as universal noise model with no systematical distortion for the data. To model the white noise a random matrix with normal distribution was used, mean value was set to is zero, standard deviation was 1. The random signal is related to the maximum value in input data. Two noise models were used, with SNR set to 20 dB and 40 dB. Such a noise contamination of synthetical data sets can be seen also in other experiments with potential field data (for example in [12] and [15]).

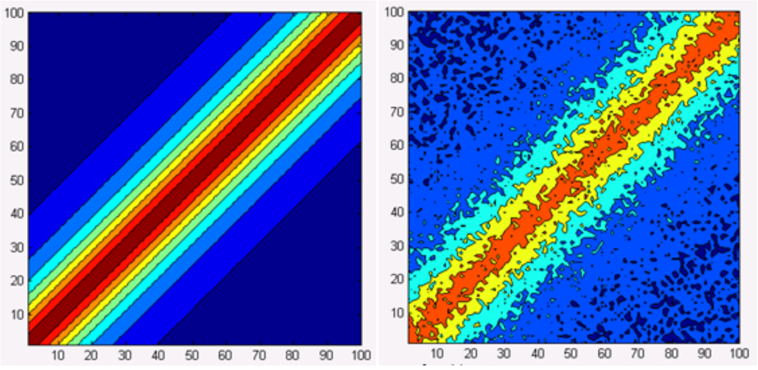


Figure 2.10: The input data converted to the images - the original vertical cylinder (left) and the noise corrupted data (right).

2.2 The continuous monitoring process

In the case of the waste deposit monitoring, the computer vision or machine learning techniques can be applied to detect significant anomalies in the data stream. Regardless of the monitoring technology, the physical conditions in the repository such as water saturation or temperature should either remain unchanged or they sould change in a known manner. If any difference in monitored data is captured, it is necessary to recognize the cause of the change. Physical parameters in the repository can slightly oscillate around the equilibrium, which can be understood as a normal behaviour, or they can more dramatically increase/decrease. Atypical changes in the acquired data stream can alarm a problem in the repository – for example the surrounding barrier may be corrupted and safety of the repository can be endangered.

The repository itself is strictly defined – it is a structure with defined and well known geometry, with stable homogeneous surrounding. The process of selecting and building of the repository lasts for years and it is very well planned and prepared. During the preparation process it is possible to start pre monitoring of to get the stable data streams as reference training sets describing the correct operation mode of the repository.

The monitoring proces can be either focused just to detect any modification which is different than the normal operational mode or it can be prepared to detect predefined abnormal situations. According to the selected monitoring methodology the set of anomaly data can be prepared. The modeled anomalies would describe the expected abnormal situations occurring in the repository – increasing temperature over the prediction, modified water saturation, modified geometry etc. The task for the monitoring process is than modified: the algorithm searches the known abnormal situations.

The geophysical monitoring of the nuclear waste repository is a part of the research of the project called Modern2020[2]. The Technical university in Liberec is one of the participating research organisations in the project. The author of the thesis is responsible for the research related to the geophysical data processing described in the presented thesis.

The geophysical monitoring of the repositories is just a part of the research. The geophysical methods included to the project are: Electric resistivity tomography (ERT), Induced polarisation (IP) and Seismic methods (SM). For the ERT it is planned to set up a real monitoring experiment in the real operating condition of the repository. The IP is to be also run in the real operating condition with ERT as a supplementary method to distinguish the influence of changing water saturation and the temperature. For the SM, the full waveform seismic inversion is to be adjusted for the target application. The machine learning techniques are to be used as a supplementary method for the full waveform inversion ([29], [24], [23]).

The initial task related to the presented research was to adapte existing anomaly detecting algorithms to be applicable as a secondary methodology of the data interpretation for the seismical data and to test its usability in this application. The aim of the research is to test if any modification in the reservoir configuration can be automatically extracted from the seismical data.

The application is based on the synthetical seismical data. The model repository is a ciruclar shaped tunnel. Two monitoring boreholes toward each other at an acute angle are located in the plane perpendicular to the tunnel. The wave sources are located in one of the boreholes, the receivers in the other one (the situation is sketched in the Figure 2.11). The configuration is based on the experiments and research done by ETH Zurich ([24], [23]).

[2] http://www.modern2020.eu

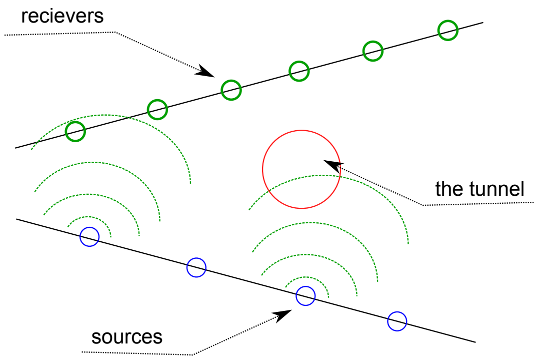


Figure 2.11: The initial model configuration for the continous monitoring process.

The Figure 2.12 shows the inputs for the data modeling. It consits of the map of the density (the left side in the image) and seismic velocity (the right side in the image) of the material in the modeled area. The modeled tunnel is located in the middle of the area.

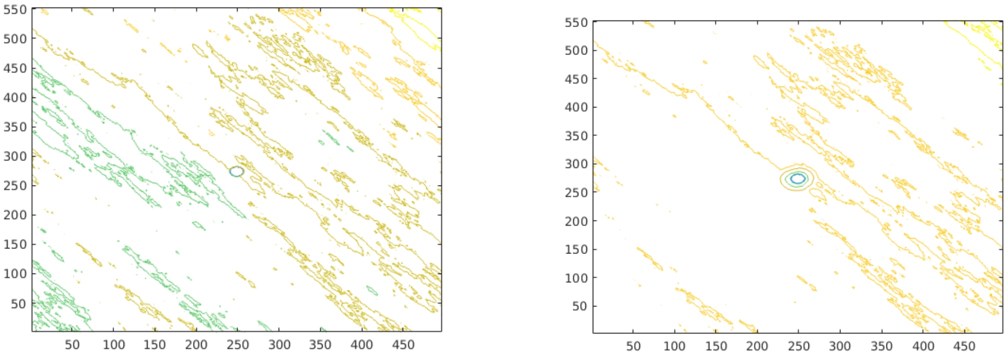


Figure 2.12: The configuration of the data model – the density map (left) and seismic velocity map (right) of the area.

The model contains 114 sources and 104 receivers. Additionally to the recievers in the reciever borehole, a set of 8 recievers located around the tunnel is added (these recievers are not presented in the Figure 2.11 to keep the figure comprehensible). The signal from each of the sources is sampled in 2000 samples in each of the recievers. The final data set is a cube of 114 × 104 × 2000. Initially the research started with the data model created for the completely dry tunnel, the fully water saturated tunnel and as a reference was generated a set of the tunnels with different geometry. The models were created and calculated by our project partner from ETH Zurich ([29]).

The fast scan algorithm takes the initial data cube and divides it into a 104 images which are understood as the 104 samples of the current repository configuration. The example of one of such data sample is available in the Figure 2.13. The upper part of the image contains the signal collected from the 50th source without any modification. The lower part of the image shows the input of the algorithm: normalized data matrix.

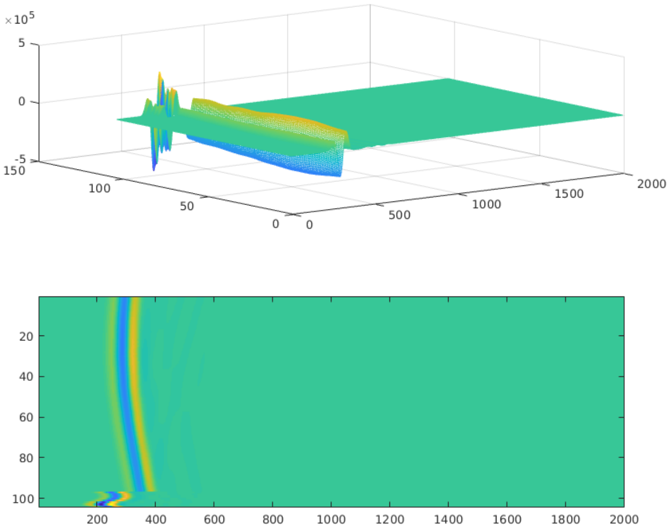


Figure 2.13: The input data for the continous monitoring scan, the original input (top) and the normalized input (bottom).

In this case the main part of the work is to find and define the structures in the data which are related to the modification of the repository conditions. The environment seismics velocity varies with the water saturation. Therefore it was decided to create several models of different water saturation in the model and to test if it creates a detectable footprint in the data. The research is not finished yet so only first outputs are presented in the thesis in the chapter 3.5.

3 Anomaly detection implementation

3.1 The current application of computer vision and machine learning in geophysics

The presented work is focused to the application of the computer vision techniques to the geophysical domain with defined application. The study was originally commissioned as a first test if the idea of semi automated data processing in geophysics is available. It was decided to start with simple models to verify, if the application is possible. The author of the study is a computer vision and data processing specialist with no preliminary experience in the field of geophysics.

The presented short research of current state of the research and application focuses to the geophysical data interpretation done with the support of computer vision and machine learning techniques. Even if these techniques are used and tested already in geophysics it is still a minority technology. The most of the research of the data intepretation in geophysics is still focused to the classical methods based on forward models and data inversion techniques.

When the research was prepared the focus was on the applications where the geophysical data are processed as images regardless of the data acquiring methodology. The attention was paid mostly to the classification problems, structure detection and feature extraction. The very actual overview of the actual applications of the MLT with the general overview of applied technologies in the geosciences is given in [20] with several practical examples. The current research is focused to all the typical techniques of the CV and MLT, including simple structure detectors based on the computer vision techniques or more complex solutions using the self organizing structures such as neural networks.

When geophysical data are interpreted as images lines or curves are typical structures of interest. The Hough transform and its modifications are used to detect the structures for the long time (the work [11] from 1998 optimizes Hough transofrm for geophysical data) - an application can be find in [13] where the Ground penetrating radar (GPR) data are converted to the image and using the Hough Transform scanned for linear structures or in [13] where the Hough Transform is used to identify the planar and linear structures in the GPR data [14] or to identify the structures [21] with support of learning algorithm.

The example of a task similar to the presented target application is the landmine and unexploded ordnance (UXO) detection. There anomaly – landmine – has typical shape, material and it is located close to the source. The CV and MLT is used in the field of landmine detection – the data can be processed using fusion algorithm ([38]) or with the neural network ([34]).

Another similar application when a typical structure is searched close to the surface is the location of buried plastic pipes. A multi aged supported detection is described in [1], the neural network and pattern recognition based on the Hough transofrm is used in the [30].

The other typical application when the MLT and CV are very useful is the computer vision-based rock-type classification - it can be based on the pattern recognition [27], neural network [31] or the self-organizing map neural network ([16]).

A lot of applications of the neural networks and geophysical data were published last years. Considering the task defined in this thesis, interesting is the application of the celular neural network to detect the edges in the data ([2]) or to process the Bourger anomaly map ([3]). In the seismics domain the self organizing maps were adopted for the characterization of 2D seismic lines ([19]). To process the gravity data a neural network was applied to Bouguer data to obtain depth, density contrast, and locations of the structures ([19]). Inspiring is also the work where the neural network is used to evaluate the gravity data ([15]) where the first tests are also done using synthetical spherical models.

3.2 Fast scan based on structure detection

3.2.1 Gravimetry: The spatial parameters estimation

The determination of the centre of the mass or the top of the anomaly body is the one of the major importances of the gravity data analysis.

Using the forward models for the simple anomaly bodies such as sphere, horizontal cylinder, vertical cylinder, prism or thin sheet, the relation between the gravity anomaly and its depth can be easily derived directly from the forward model equation. The base idea is to use the forward model to express the half width $x_{0.5}$ of the anomaly.

The situation is depicted in the Figure 3.1 - the half width $x_{0.5}$ is depicted as X_half. The V_z stands for vertical part of the gravity field, the d is the depth ($ZPos$ in the Figure 3.1).

By fitting the Vz_half value into the equation, we can determine the relation between the $x_{0.5}$ and d. The analytical expression for this parameter derivation is demonstrated in the Equation 3.1. For other anomaly bodies the the same derivation procedure can be used (see [25] page 55-57 or [32], page 51-52).

$$V_z = (V_z)_{max} \times \frac{d^3}{(x^2+y^2)^{\frac{3}{2}}} = \frac{(V_z)_{max}}{\left(\left(\frac{x}{d}\right)^2 + 1^2\right)^{\frac{3}{2}}} \tag{3.1}$$

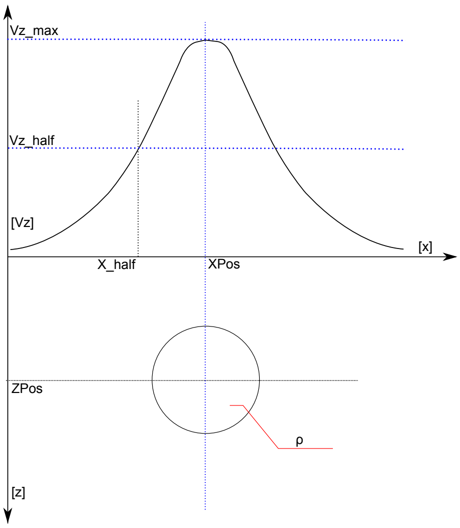


Figure 3.1: The spatial parameters estimation, spherical body.

$$
\begin{aligned}
V_z(x_{0.5}) &= \frac{(V_z)_{max}}{2} = \frac{(V_z)_{max}}{\left(\left(\frac{x_{0.5}}{d}\right)^2 + 1^2\right)^{\frac{3}{2}}} \\
2 &= \left(\left(\frac{x_{0.5}}{d}\right)^2 + 1^2\right)^{\frac{3}{2}} \\
d &= 1.305 \times x_{0.5}
\end{aligned}
$$

The value $(V_z)_{max}$ is available as a maximum value in the input data set and the d can be set, both values can be used to estimate the total mass M of the anomaly - for the spherical body it is demonstrated in the Equation 3.2. The r is the surface distance from the central point of the anomaly (see Figure 2.1). At the $(V_z)_{max}$ the $r = 0$. Similar derivation can be found for all the other simple anomaly bodies. The Table 3.1 lists all the simple models with its depth and total mass estimations for all the simple bodies.

$$
V_z = \frac{GMd}{(r^2 + d^2)^{\frac{3}{2}}} \tag{3.2}
$$

$$
(V_z)_{max} = \frac{GMd}{(d^2)^{\frac{3}{2}}}
$$

$$
M = \frac{(V_z)_{max} d^2}{G}
$$

Anomaly type	**The depth estimation**	**The total mass estimation**
Sphere	$d = 1.305 x_{0.5}$	$M = \frac{(V_z)_{max} d^2}{G}$
Horizontal cylinder	$d = x_{0.5}$	$M = \frac{(V_z)_{max} d}{2G}$
Vertical cylinder	$d = \frac{\sqrt{3}}{3} x_{0.5}$	$M = \frac{(V_z)_{max} d}{G}$

Table 3.1: The simple anomaly bodies with d, ρ and ρ_c parameters extracted from the models.

The relation between the $x_{0.5}$ and d was the initial inspiration for the fast scan algorithm. The general idea was simple: to use the image procession techniques to get the $x_{0.5}$ value from the data, to compute the estimated field and to compare it with the input data. The following section describes what structures were to be detected in the field data to classify the anomaly type.

3.2.2 The structures in the data

When the field model values of a simple anomaly body is depicted in colours in XY plane as in Figures 2.6, 2.7, 2.8, one can easily notice that each type of anomaly creates a simple structure in the 2D picture of the data. For the spherical and vertical cylinder anomaly, the structures are circles. The horizontal cylinder creates parallel lines in the picture.

This fact was used to estimate the $x_{0.5}$ value for all the anomaly types. If an ideal spherical anomaly body is hidden under the surface, we can cut the field at several levels. For example, in the picture 3.2 the spherical and vertical cylinder anomaly fields are cut at levels equal to $0.25 \times (V_z)_{max}$, $0.5 \times (V_z)_{max}$ and $0.75 \times (V_z)_{max}$. The outline of the cut is always spherical.

The two other types of anomaly body - the horizontal cylinder and rectagonal prism - have different cut outlines. The outline of the cut of the horizontal cylinder is a pair of parallel lines, for the rectangular prism, it can be a two pairs of oblique lines as it is demonstrated in the Figure 3.3.

For the spherical body and both types of cylinders, the outline of the cut is always of the same shape. The anomaly parameters (depth and density contrast) only affect the radius of the circle or the distance of the lines. For the rectangonal prism, the situation is different. Only for high density contrast near the surface we can see the cut outline as it is demonstrated in the Figure 3.3. With the increasing

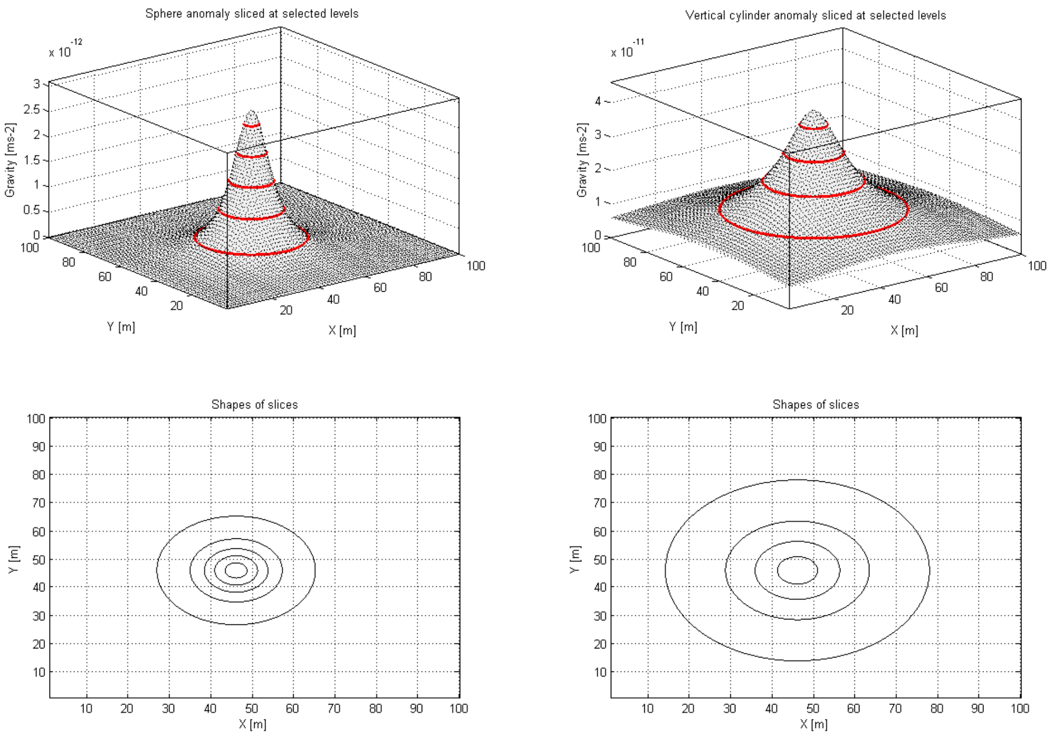


Figure 3.2: Spherical anomaly (left) and vertical cylinder anomaly (right), the outlines of the field cut at several levels.

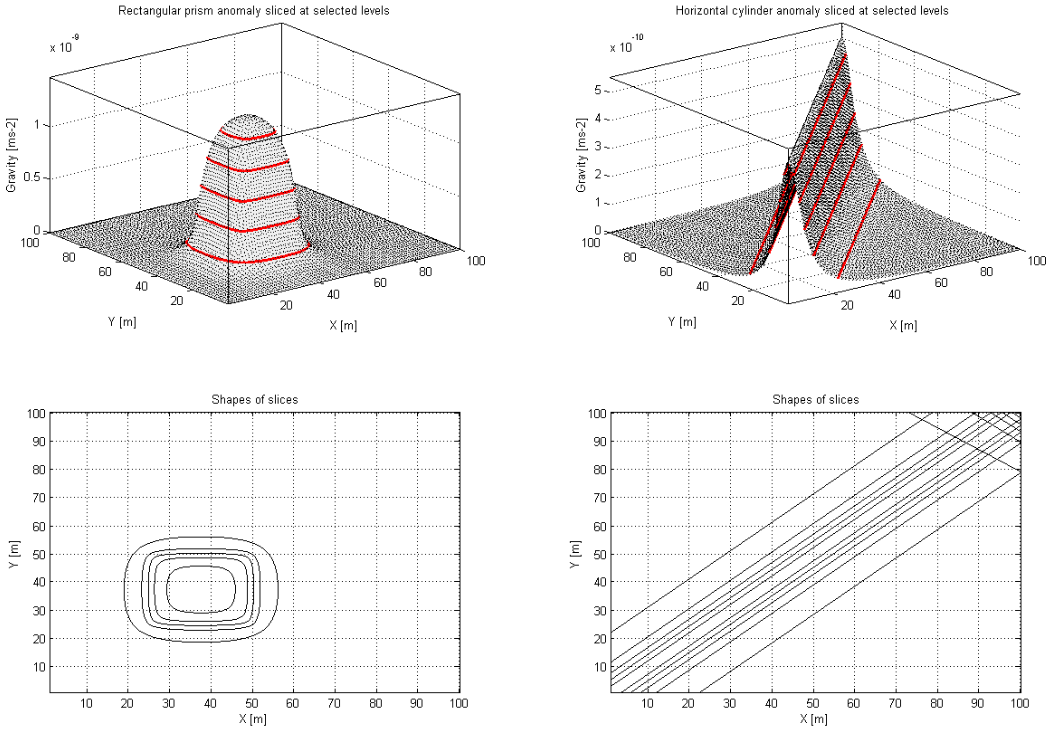


Figure 3.3: Anomaly characteristics for the rectagonal prism (left) and horizontal cylinder (right).

depth or decreasing density contrast, the cut outline is very close to the circle shape - see the Figure 3.4. In this case, both anomaly bodies are located 5 m under the surface, the radius of the sphere is 5 m, the prism is a cube with the edge line equal to 4 m.

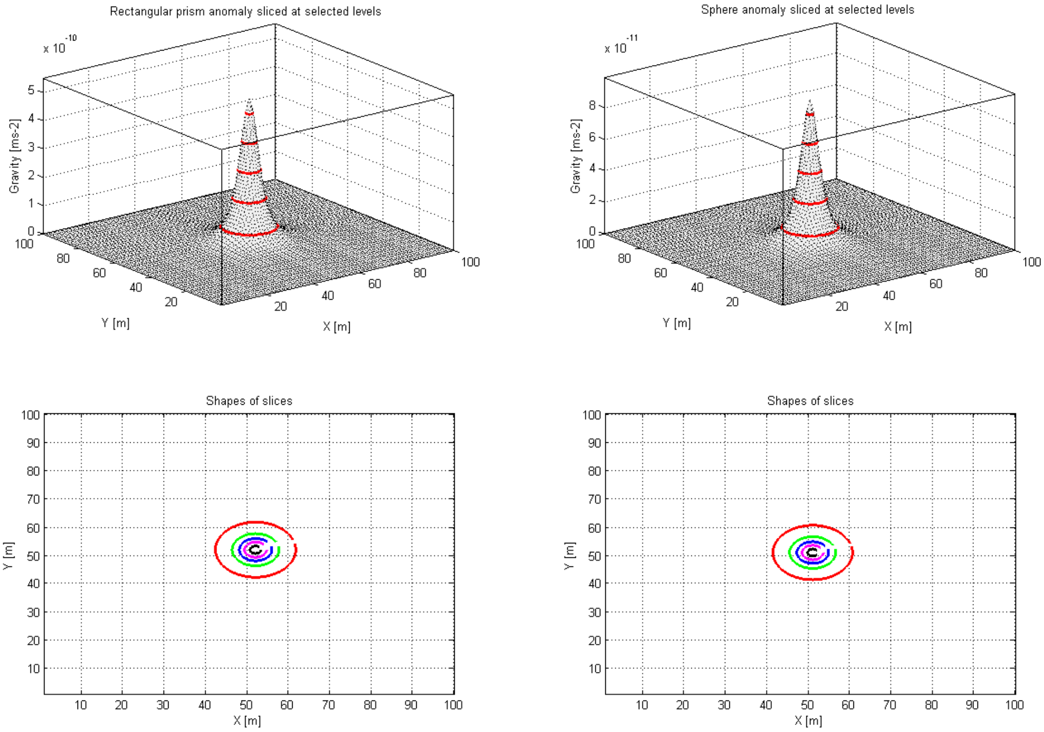


Figure 3.4: When the sphere and rectangular prism have similar cut outlines.

In this case the algorithm will probably misfit the prism body with the spherical body, as the circle structures will be detected in the data picture. In fact, such situation is not dramaticall, if the depth of the anomaly center and the total mass of the hidden body will be estimated with acceptable precision. The target application should provide the fast scan of the data and it should detect the areas for the future more detailed exploration, it does not have to precisely distinguish between the anomaly body types. In fact, no real anomaly have exactly spherical or cubical form.

At the beginning of the shape detection process the input field is normalized from original values to the interval of $(0-1)$, according to the Equation 3.3. Depending on the density contrast value the input data matrix can be both positive and negative. For the algorithm design, initially only one anomaly was modeled in the data, so data values are all negative or all positive. The normalization procedure takes the absolute value of data. If multiple anomaly body would be present in the data, $|V_z|$ cannot be used. The model with multiple anomaly body with different contrast densities is discussed in the Section 3.2.7.

$$(V_z)_{Norm} = \frac{|V_z| - (|V_z|)_{min}}{(|V_z|)_{max} - (|V_z|)_{min}} \tag{3.3}$$

The normalized data field is now thresholded at several selected levels. The initial idea was to use just one threshold at the value $(V_z)_N : N = 0.5$ to get the shapes to detect the value of $x_{0.5}$. The initial classifier searched through the image for linear and circle structures. Than it measured the radius of the circle or the distance of the lines (if parallel lines were detected). The initial classifier had simple logic:

- **Circle structure detected** – a spherical anomaly or a vertical cylinder anomaly is detected. The circle radius was used to estimate the d and M_s for a sphere and M_c for a cylinder.

- **Two parallel lines detected** – a horizontal cylinder anomaly is detected. The half of distance of the lines is $x_{0.5}$ and it is used to estimate both d and M_c.

- **Two pairs of oblique lines detected** – a rectangular prism, the direct estimation of the d and M is not possible just by derivation of the field definition equation. Therefore a look up table was calculated.

- **Any other structure detected** – an unknown type of anomaly is hidden in the data, no parameters estimation is done.

Such classification works correctly only for the ideal smooth data, with the anomaly positioned close to the middle of the area. Unfortunately, a lot of misdetection can appear. If the white noise is given to the classifier, it can try to find horizontal cylinder in the data as well as the rectangular prism, because a lot of lines is detected. If the smooth data are combined with the noise as it was described in the section 2.1.2, a lot of small circles can be misdetected, linear structures can be corrupted and remain undetected. Therefore it was decided to use a bigger set of thresholds and to design a more complex classifier.

The normalized field is thresholded at 9 levels to cut the V_N at levels from 0.1 to 0.9. The detection of the lines and circle structures is done at all the levels. This part of algorithm is implemented in the Matlab environment. To detect the line structures, according to the theory of the line detection given in [7], the application of Hough transform was selected as the best methodology. The Matlab Hough transform implementation was used (the functions *hough*, *houghpeaks* and *houghlines* were used).

For the circle structure detection, the detection was initially done by the originally implemented algorithm. The algorithm tested, if there is a connected region in the picture of a near circular shape, but a lot of false data were indentified as a circular or vertical cylinder anomaly. Therefore it was decided to use the more precise circle structure detection with the Circular Hough Transform. The current version of the algorithm uses the Matlab implementation *imfindcircles*, which was introduced in Matlab in 2012.

If a horizontal cylinder is the source of the anomaly in the data, in ideal smooth data parallel lines with always the same direction should be detected in all the thresholded images. Such situation is depicted in the Figure 3.5. In the Figure 3.5 all 9 thresholded images are shown as well as the original data. The top right image is cut done at level 0.9, the low right image is the cut at level 0.1. The red line shows detected lines. In all the thresholded images just 2 pairs of lines were detected, running always the same direction. The depicted cylinder is located 19 m under the surface, its radius is set to 1 m.

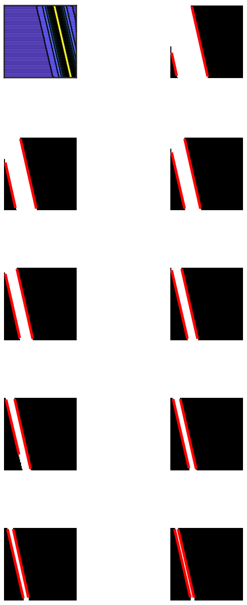


Figure 3.5: The ideal line detection for the horizontal cylinder.

The situation is not always as ideal as it is depicted in the Figure 3.5. Several causes of the misfit were identified during the preliminary research and the classifier was updated to be able to classify directly the anomaly type in such a situation.

Problem 1: An indistinctive anomaly body close to the border of the area. An example of a such configuration is depicted in the 3.6 (on the left). The depth of the presented anomaly body is set to 63 m and radius was set to 1 m. Therefore the field structure is flat and due to the position of the cylinder only one edge of the structure is detected. The other one was always out of the image and therefore it was not detected. Even if the classifier logic would be updated to search through all the slices for a single line with uniform direction, the missing second line means that the $x_{0.5}$ value cannot be estimated from the data. But when the target application is taken into the account, such an indistinctive anomaly would not probably be the target of interest.

Problem 2: The noise destroys the linear structures. The Figure 3.6 (on the right) shows the data with the noise at level 20 dB. The anomaly body is again quite indistinctive and with the noise the linear structures at the level 0.3 and 0.4. At the levels 0.8 and 0.9 more lines are detected because of the noise, with different directions. To avoid such situations, noise filters were applied to smooth the data. To clean the linear structures a set of morphology operations can be used before the structure detection is started. This can again slightly increase the success rate of the detection (see the Section 3.2.3 for details).

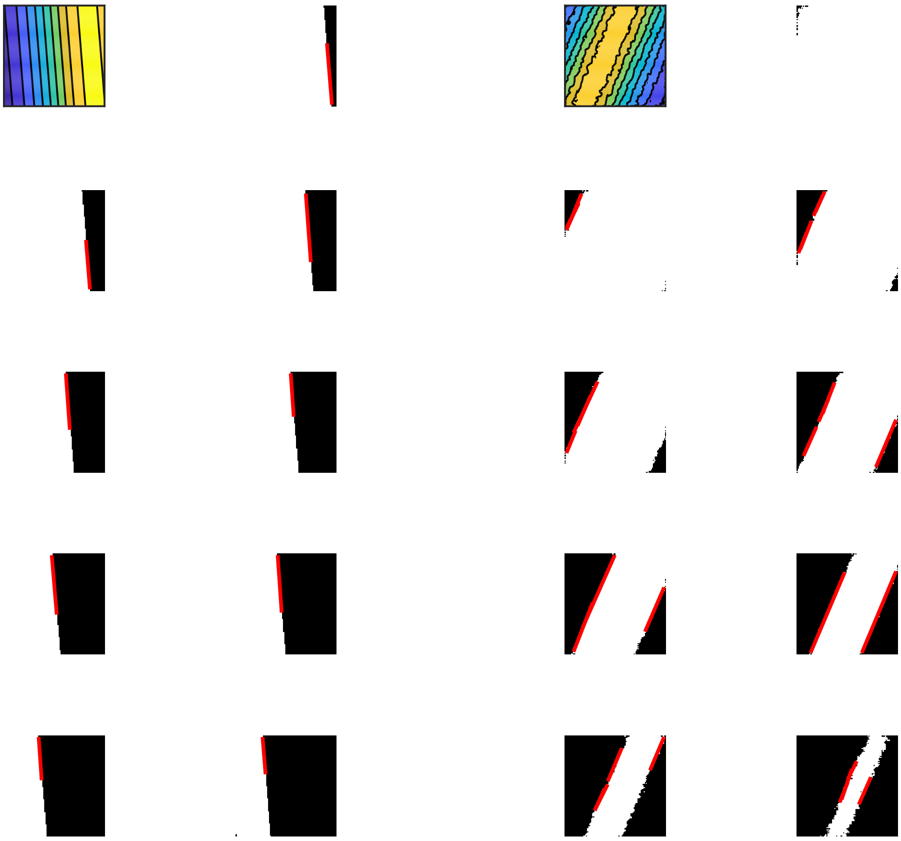


Figure 3.6: The undetected cylinder: on the left the structure is so flat, that only one line is found in the thresholded images, on the right the noise destroyed the linear structures in the data.

Problem 3: False line detections in the false data. The Figure 3.7 on the left side depictes a random data processed by the detection algorithm. The lines are detected at all the thresholded levels, a lot of parallel lines is detected. Due to this misdetection it was decided to detect the lines at all the levels. The detection algorithm tests, if the direction of lines is the same at all the levels where lines are detected. The direction of the line is measured in the angle θ between the line and low border of the image. The classifier decides that all the lines have the same direction, if the $\Delta\theta$ is lower than 5 degrees for all the detected lines at all the levels.

Problem 4: False line detections at the other anomaly type. If the body of spherical or vertical cylinder anomaly has a bigger radius, at levels 0.1 to 0.4 lines can be also detected as it is illustrated in the Figure 3.7 on the right side. The picture is used to demonstrate the other situation, when simple line detection at just one level is not enough. It must by always tested, if detected lines have the same direction at all the levels where lines were detected.

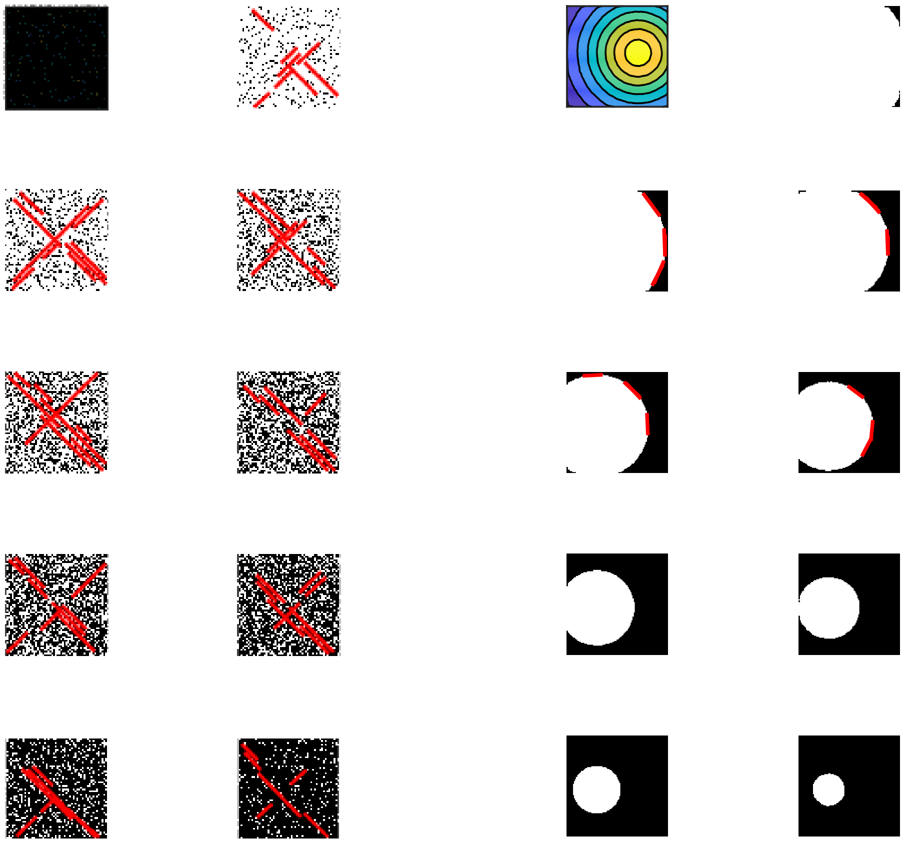


Figure 3.7: False lines detected: on the left in the white noise, on the right a misdetection at levels $0.1-0.3$ for vertical cylinder with a large radius.

Similar detection problems can appear with circle structure detection. Figure 3.8 demonstrates the situation for a smooth spherical anomaly (left) and vertical cylinder (right). Even in a such ideal case, the circle structure is not detected in all the thresholded images. As for the line detection, with circle structure detection a set of false detections must be avoided. First of all, at levels 0.1 to 0.3 at any kind of anomaly source, as well as for the reference false data, a lot of small circle structures can be detected. Situation is demonstrated in the Figure 3.9.

Therefore the classifier omits all the detected circles with radius smaller than a given threshold. (For an area 100 × 100 m the best solution is this threshold set to 5 m.) If more than one circle is detected, the biggest one is selected.

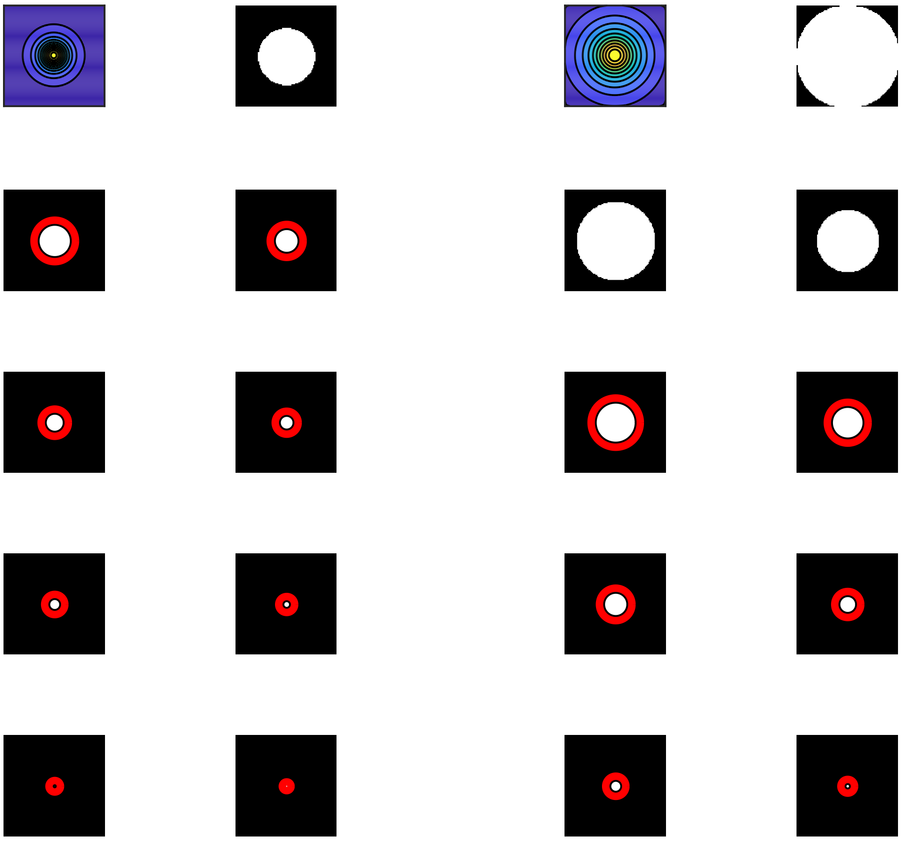


Figure 3.8: Ideally placed, ideally smooth data and circle detection for a sphere (left) and an vertical cylinder (right).

The structure detection ends up with following parameters:

- **Parallel lines detected** (true/false) – a parameter is set to true, if parallel lines are detected at levels 0.4, 0.5 and 0.6 and all detected lines have the same direction (with predefined tolerance $\Delta\theta < 5°$).
- **The main θ value** (numerical value, 0-180) – if parallel lines were detected,

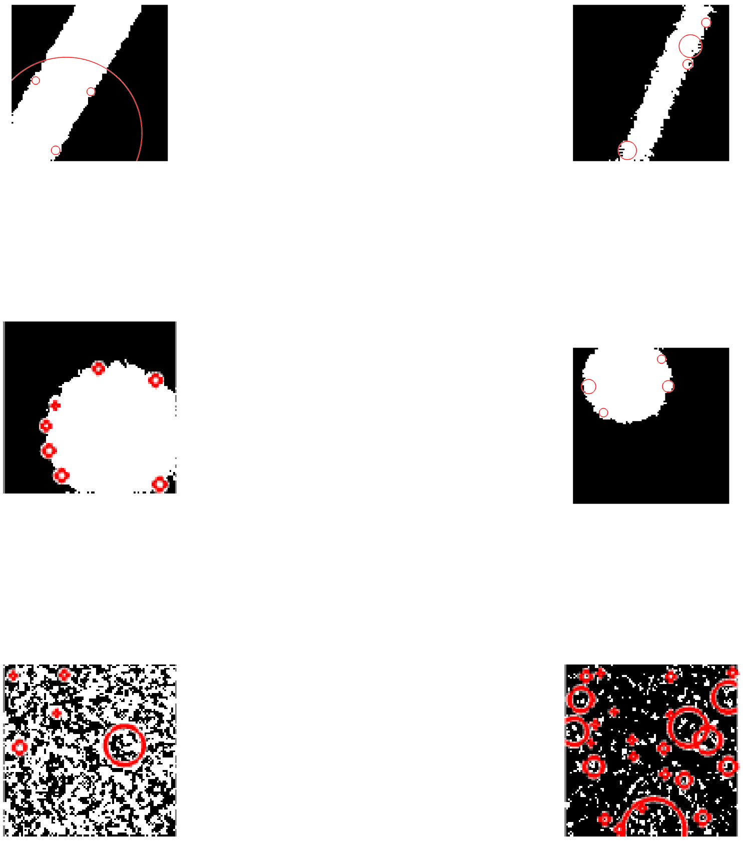


Figure 3.9: False detection of circle structures: horizontal cylinder (top left), unclear borders of the circle structure (top right, middle left, middle right), a false data (bottom left and right).

final θ value is set to a median of the measured values at all significant levels (0.4, 0.5, 0.6).

- **Line distance** (numerical value, 0-100) – if parallel lines are detected, at the level 0.5 the distance between lines is measured, this distance is equal to $2x_{0.5}$ for a horizontal cylinder.
- **Main circle detected** (true/false) – a parameter is set to true, if at least at 3 levels a circle was detected with center point at the same place (with predefined tolerance $\Delta XPos < 3m$ and $\Delta YPos < 3m$).
- **Main circle central point** (a pair of numerical values, both 0-100) – final values of the central point $[XPos, YPos]$ are set to median value of all detected central points.
- **Main circle radius at level 0.5** (a numerical value, 0-50) – the radius of a circle at level 0.5. The radius is equal to $x_{0.5}$ for a spherical or a vertical cylinder anomaly body.
- **Other radiuses at detected levels** (a vector of numerical values, 0-50) – if no significant circle was detected at level 0.5, but still a set of circles was detected at other levels, a vector of other radiuses is given to estimate the level at 0.5. This estimation is done by classifier.

3.2.3 Morphology operations to clean noise distortion

If the algorithm gets as the input clean data with no added noise, the detection of the structures is good. The more noise is presented in the data, the more false detection, mostly for small circles is done (as it was illustrated in the Figure 3.9). Due to the noise the originally smooth border between black and white area is crooked. Such distortion is present in the data even if the data were prefiltered using any denoising filter.

If the thresholded slices of the data were morphed using the propriate morphology operation, the border between the black and white area should be smoother. The erosion operation was selected to close the boundaries of the objects in the black and white images. To keep the precision and to avoid distortion of the structures in the image, the erosion was tested with structural element of size 3×3 points (a small cross). If the structural element is bigger, the erosion itself distorts the structures in the images.

The algorithm gave best results if the erosion operation was repeated twice for noise corrupted data. The number of misdetected small circles went down, but not to zero. Therefore it was decided not to use the erosion process - it can possibly distort original data, it takes time and it does not give reasonable results.

For the first implementation of the algorithm where the circular structures were detected by measuring the connected regions in the image, the morphology operations were important to increase the precission of the detection. With the current implementation based on the Hough transform it is not so important. Therefore instead of erosion, the circle detection algorithm is now designed to it ignore all the circles with radius smaller than a given threshold. This value is a parameter of the algorithm, for the test data it was set to 5 points. The morphology is still implemented in the algorithm and can be switched on. The structural element can be redefined in the algorithm configuration.

3.2.4 The anomaly type classification

The anomaly type classification is based on the parameters defined by the structure detection part and its decision process is depicted in Figure 3.10. The input of the process is the information, if the parallel lines were detected in the area. If parallel lines were detected, the anomaly type is set to horizontal cylinder. In this case, circle structures are ignored, because a circle misdetection at the line borders appears quite often.

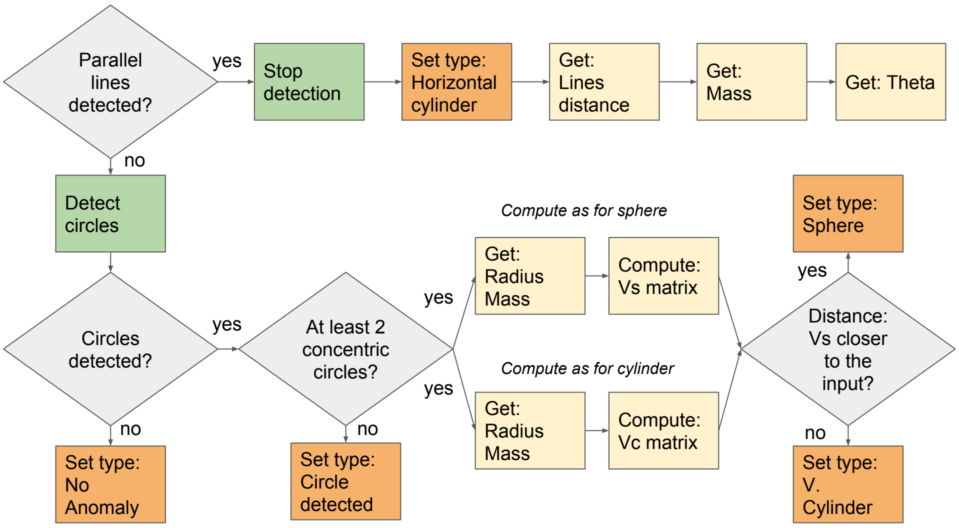


Figure 3.10: Anomaly type classification – the decision process.

If no lines are detected in the picture, the detection process continues to search for circle structures. If circles are detected, the classifier must correctly identify the anomaly type. The original idea was to take the thresholded image at level 0.5, to measure the circle radius and to estimate the d and M parameters for both anomaly types (spherical, cylindrical) using the $x_{0.5}$ to d relation described in the Table 3.1. With the estimated d_s, M_s for the sphere and d_c, M_c for the cylinder, the V_s and V_c matrices were calculated. Than the original input V_z matrix was compared with V_s and V_c. If the V_s values were closer to the V_z, the anomaly type was set to sphere, otherwise vertical cylinder was selected:

The first problem of such solution was the selection of a metrics used to compute the distance of the orignial and proposed field. Several metrics were tested: the total sums of differential (Equation 3.4 and the second power of the euclidean distance (based on [8], page 242, Equation 3.5):

$$\begin{aligned} E_{sD} : e_{sD}[i,j] &= |V_z[i,j] - V_s[i,j]| \qquad (3.4) \\ E_{cD} : e_{cD}[i,j] &= |V_z[i,j] - V_c[i,j]| \\ \Delta_D &= \sum_{i=1}^{n}\sum_{j=1}^{n} e_{sD} - \sum_{i=1}^{n}\sum_{j=1}^{n} e_{cD} \end{aligned}$$

$$\begin{aligned} E_{SE} : e_{sE}[i,j] &= V_z[i,j]^2 - V_s[i,j]^2 \qquad (3.5) \\ E_{CD} : e_{cE}[i,j] &= V_z[i,j]^2 - V_c[i,j]^2 \\ \Delta_E &= \sum_{i=1}^{n}\sum_{j=1}^{n} e_{sE} - \sum_{i=1}^{n}\sum_{j=1}^{n} e_{cE} \end{aligned}$$

Both type of distances can be used to differentiate the anomaly type, the more suitable is the Euclidean distance as it gives bigger Δ and therefore it is used in the algorithm.

If a circle structure was correctly detected at the $x_{0.5}$ level, the precision of the anomaly parameters estimation and the anomaly type classification was quite good: The anomaly type was set correctly in 95 %, the position parameters were set with 1-2 m precision. But as the cylinder structure creates in the data a circle with quite a big radius, in 90 % the circle was not detected. Only part of the circle arc was presented in the picture. At the upper levels the circles were detected correctly.

It was necessary to measure the radius at all the levels, where circular structures were detected. The original relation between the $x_{0.5}$ and the d parameter can be modified for any other data cut if the threshold level is expressed as a fraction of $(V_z)_{max}$. If the levels 0.1, 0.2,..., 0.9, the corresponding V_N can be defined (for the $N = 5$ the $r_N = x_{0.5}$):

$$V_N = \frac{N}{10}(V_z)_{max}, N = \{1, 2, ..., 9\} \qquad (3.6)$$

The V_N value lies at the border of the detected circle with radius r_N. Similar to $x_{0.5}$ to d ratio can be derived a relation between the r_N the d parameter for all the values of N used in the image thresholding. The Equation 3.7 defines the r_N to depth for a spherical anomaly, the Equation 3.8 for the vertical cylinder and the Equation 3.9 for the horizontal cylinder.

$$\frac{N}{10}(V_z)_{max} = \frac{(V_z)_{max}}{\left(\frac{r_N}{d_s} + 1\right)^{\frac{3}{2}}}$$

$$d_s = \frac{r_N}{\sqrt{\left(\frac{10}{N}\right)^{\frac{2}{3}} - 1}} \tag{3.7}$$

$$\frac{N}{10}(V_z)_{max} = \frac{(V_z)_{max}}{\sqrt{\left(\left(\frac{r_N}{d_c}\right)^2 + 1\right)}}$$

$$d_c = \frac{r_N}{\sqrt{\left(\frac{10}{N}\right)^2 - 1}} \tag{3.8}$$

$$\frac{N}{10}(V_z)_{max} = \frac{(V_z)_{max}}{\left(\frac{r_N}{d_c} + 1\right)^{\frac{1}{2}}}$$

$$d_h = \frac{r_N}{\sqrt{\frac{10}{N} - 1}} \tag{3.9}$$

The values of r_N are set using the outputs of the structure detection algorithms. The Matlab function *imfindcircles* used to detect circular structures returns a vector of radiuses of detected circles. When parallel lines are detected, the output of the Hough Transform returns the values of the rho and θ values. If lines are parallel, the θ must be identical and the distance of the lines is the difference $|\theta_1 - \theta_2|$ (see 3.11 for the definition of the rho and θ).

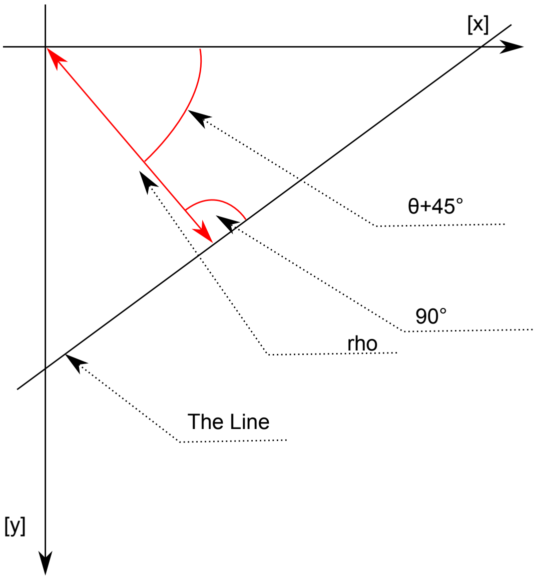


Figure 3.11: The Hough transform outputs: The rho and θ values.

The rho and θ can be used to define the line by the following formula:

$$rho = xcos(\theta) + ysin(\theta) \tag{3.10}$$

To increase the number of correctly detected anomalies, the algorithm measures the radius of all concentric circle structures in the data. Next it computes the d_s and d_c value for all the available circles. Regardless of the anomaly type, if a spherical or vertical cylinder was modelled in the input data, the variation of the computed d_s and d_c vectors is always small. The median values are used to estimate the M_s and M_c.

As a next step, the estimated fields are computed for the detected parameters $XPos$, $YPos$, d_s/d_c, M_s/M_c and compared with the original data using the Euclidean distance Δ_E. If Δ_E is positive, the original field is closer to the spherical model, otherwise it is set to vertical cylinder.

The final output of the classifier is following set of parameters:

- **Anomaly type** can be set to *Sphere*, *Vertical Cylinder*, *Horizontal cylinder* or *Other*.
- **Depth** is the estimated depth of the center of the anomaly, it is set to 0, if type is set to *Other*.
- **Mass** is the estimated total mass of the anomaly, it is set to 0, if type is set to *Other*.
- **Center** is the pair of $[XPos, YPos]$, it is set to 0, if type is set to *Other*.
- **Direction** is set only if the *Horizontal cylinder* is detected in the data to detected value of θ. Otherwise it is set to 0.

The more detailed output of the classifier is written to a table containing all the information describing the detected structures for all the input data samples. The images are indexed from 1 to 9 according to the threshold level used to get the black and white images (the index i stands for the thresholding level $\frac{i}{10}$). All the described parameters are sets of 9 values for each of the thresholding levels.

- **Lines detected**: it is set to 1, if any line structure was detected in the image.
- **Parallel lines detected**: the value is set to 1, if main pair of parallel lines detected with the main θ as it is described above.
- **Circles detected**: the parameter is set to 1, if a correct circle structure was detected at corresponding level.
- **Circles radiuses**: if a circle was detected, the value is set to the circle radius in pixels. If no circle is detected, the value is set to 0.
- **Circles X**: if a circle is detected, set to first coordinate of the central point, otherwise set to 0.

- **Circles Y**: if a circle is detected, set to second coordinate of the central point, otherwise set to 0.

All the detection outputs are stored in the **.csv* file and remain available also in the Matlab table in the workspace when the detection is finished.

3.2.5 The noise

To test the resistance of the algorithm to the noise in the data, the analytical signals used in this thesis were combined with two random signals, with normal and uniform distribution as it was described in Section 2.1.2. Following noise reduction algorithms were tested:

- Averaging filters with kernel 3x3 and 5x5
- Gaussian filter with kernel 3x3 and 5x5
- Median filter with kernel 3x3 and 5x5
- Wiener filter with 3x3 and 5x5

All the algorithm and filter kernels were implemented according to the description given in [36].

The proposed algorithm is to be based on the pattern recognition and mostly the lines, edges and other simple patterns should be detected in the data in general. As the illustration, analytical gravimetry data with horizontal cylinder were used. The clean analytical and noise corrupted data are presented in the 3.12. The SNR in this case is set to 40 dB. Without any prefiltering, the detection algorithm is not able to detect any lines in the picture.

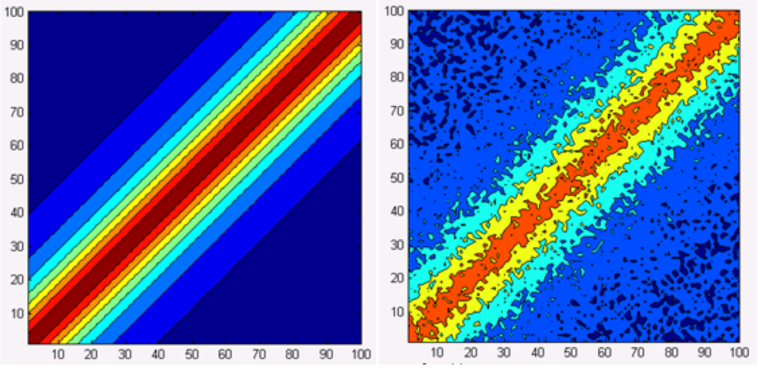


Figure 3.12: The original analytical data (left) and the noise corrupted data (right).

The test results are depicted in Figure 3.13. It is not surprising that the bigger kernel results with better smoothed data. But in the case of Gauss smoothing and median filtering the bigger kernel except the noise smoothing also distorts the shape of the anomaly, which leads to incorrect parameter estimation. The best illustration for such situation gives the $5x5$ kernel for the Gauss filtration (see Figure 3.13, the right column). The data are smoothed, but the hidden anomaly is smoothed as well and the original central line is lost. It means that the detection software will detect horizontal cylinder anomaly, but the depth and density contrast will not be correct. The anomaly is deformed and will be misunderstood as a deep and heavy object. The best results were obtained with Wiener filtering (again, see the Figure 3.13, the forth row). Wiener filter provides the best smoothing without damaging the characteristics of the anomaly.

The filter types were not tested only for presented anomaly example, but also for another types of anomalies as well as for another noise models. As the line structures are the most sensitive to the noise, the horizontal cylinder example was selected for the pictures.

The adaptive Wiener filter was selected as the preprocessing denoising filter. The only limitation of the Wiener filtration is its performance at the border of the image. The smoothing of the center area of the image is always satisfactory. But depending on the kernel size, the border of the image is unsmoothed. The unsmoothed image border disturbs the feature detection in the next algorithm steps (see 3.14). If the input data matrix is sufficiently large, the best solution is to reduce the size of the denoised image and omit the image edges from the next processing. For the final testing was used the Wiener filter with kernel 3 x 3, because it offers acceptable level of filtering and the smallest destruction of the image borders.

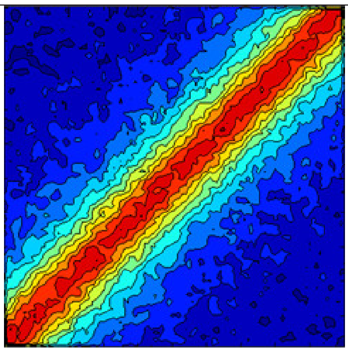
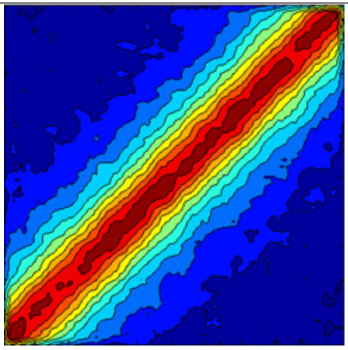
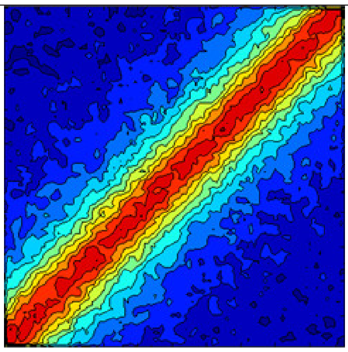
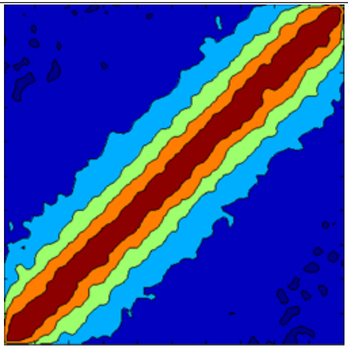
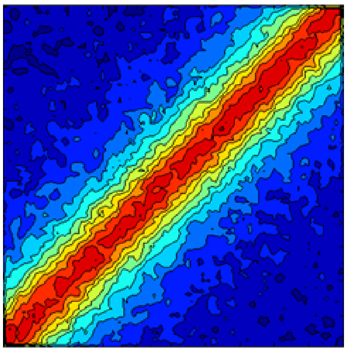
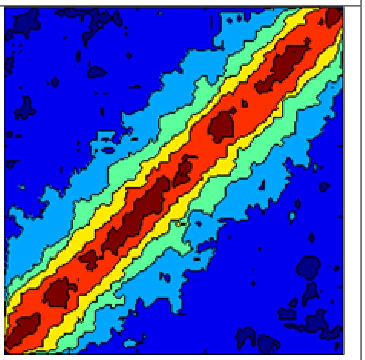
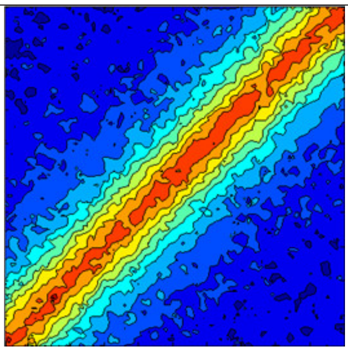
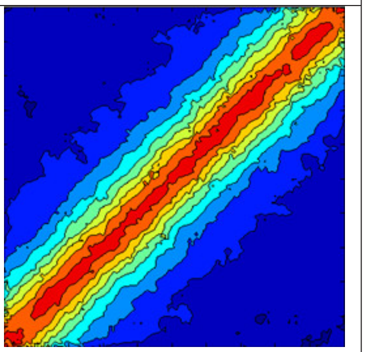
Kernel size	3 x 3	5 x 5
Averaging filter		
Gauss filter		
Median filter		
Wiener filter		

Figure 3.13: The summary for the noise elimination study.

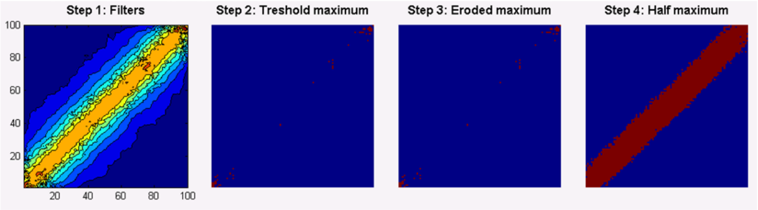


Figure 3.14: The weak point of the Wiener filtration. The image borders are not smoothed. The central high line in the image is overshadowed by noise spikes at the edges of the image and the line structure is not detected. The anomaly will be classified incorrectly.

If the input data matrix is really small (less than 20x20 pixels), none of the pixels can be omitted, otherwise the size of the picture would be dramatically reduced. Therefore the Wiener filter should be replaced by another type of smoothing filter with smaller kernel. Based on a set of tests, Median filter was selected as appropriate solution in this case.

The noise filters can decrease the anomaly detection efficiency, see final summary in Chapter 4 for the detailed statistics.

3.2.6 The general algorithm

The previous sections were describing in detail the parts of the fast scan detection algorithm. In general, all the blocks of the algorithm are connected one after another, as it is depicted in the Figure 3.15. The input of the algorithm is a matrix of vertical part of the potential field (V_z). At first, denoising Wiener filter is applied. Next step is the normalization procedure as described in Equation 3.3. The output of this block is a matrix of values from the interval $< 0, 1 >$.

The next step is the conversion of the normalized data into a set of thresholded black and white images - block *Thresholding* in the Figure 3.15. The nine threshold levels are set from 0.1 to 0.9. The output of the block is a set of 9 binary matrices. Optionally, an erosion of the data can be switched on (block *Morphology*). This is the end of data preprocessing.

When the preprocessing is done, the structure detection and anomaly type classification is done as it was described in the Sections 3.2.1 and 3.2.4. The internal structure of the block *Classifier* is depicted in the Figure 3.10.

The whole algorithm was implemented and tested in the Matlab environment. All the parts of the algorithm are implemented as Matlab functions, called from one main file. To simplify the interaction with the code, a simple graphical user interface (GUI) was created, where all the algorithm input parameters can be set. The algorithm and the GUI were packed using the Matlab packaging tool and the package can be installed from the appended CD.

The application itself can be used with or without the GUI according to the preference of the user. The general algorithm configuration is separated from the algorithm code to a config file. The structure of the algorithm implementation is depicted in the Figure 3.17.

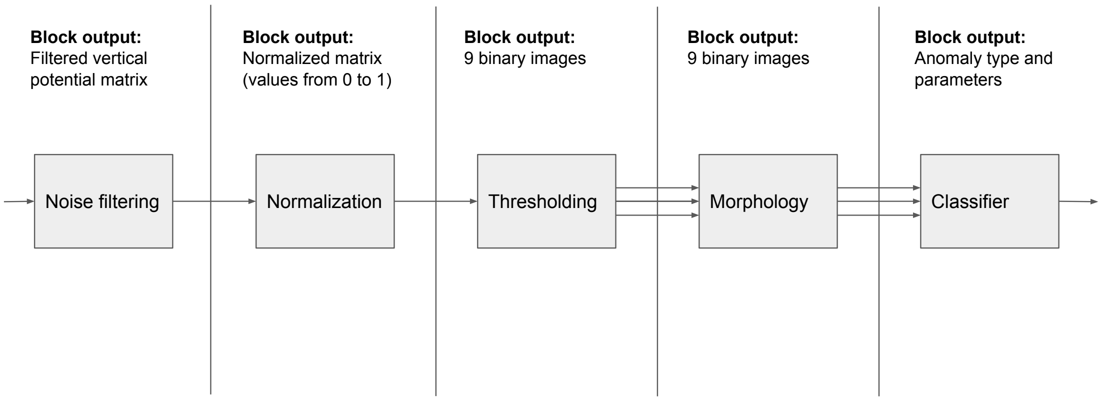
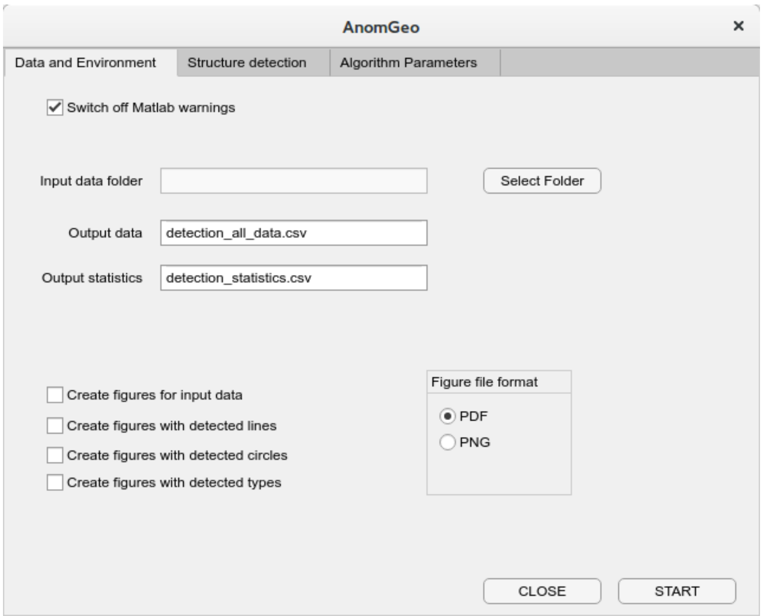


Figure 3.15: The general fast scan algorithm with all function blocks and signals.



☑ Switch off Matlab warnings

Input data folder

Output data: detection_all_data.csv

Output statistics: detection_statistics.csv

□ Create figures for input data

□ Create figures with detected lines

□ Create figures with detected circles

□ Create figures with detected types

Figure file format

● PDF

○ PNG

Figure 3.16: The graphical user interface of the fast scan algorithm.

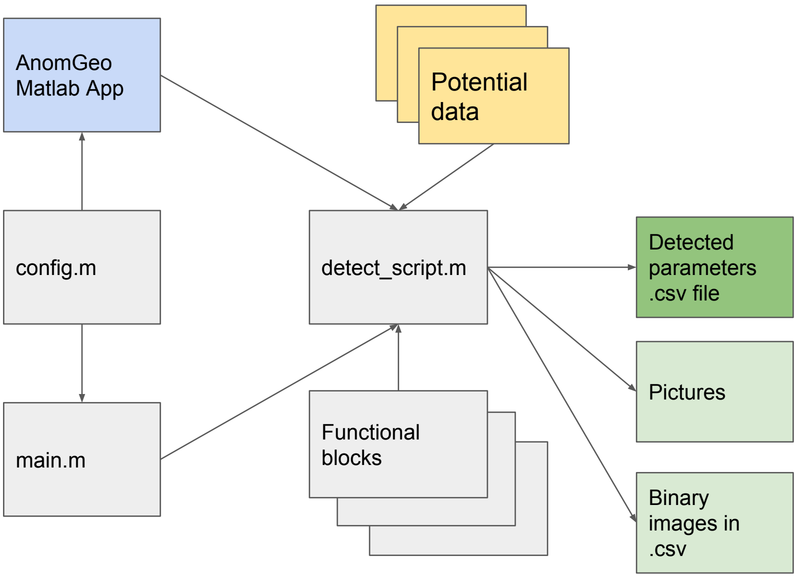


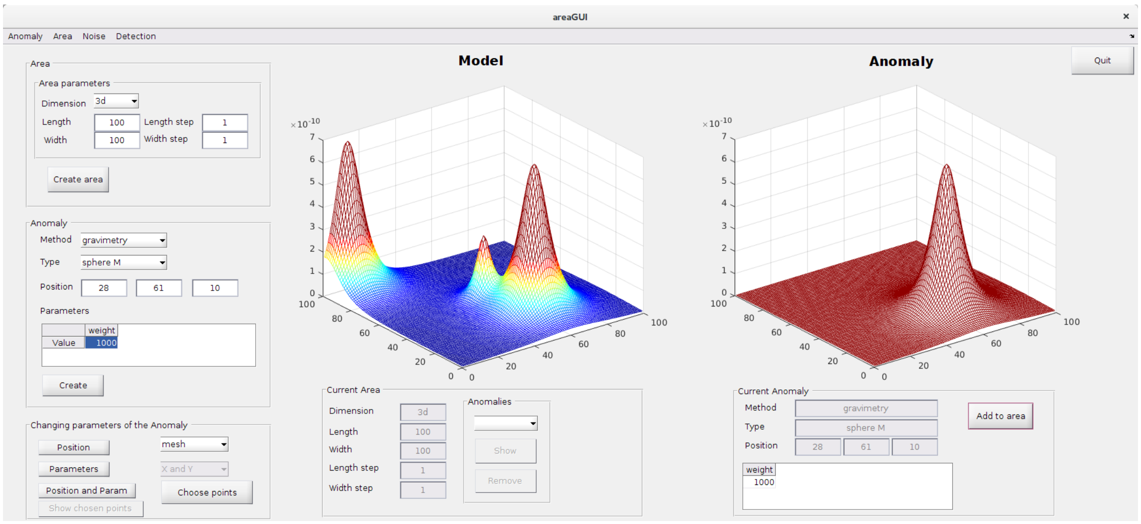
Figure 3.17: The Matlab application architecture.

All the algorithm outputs are stored in the **.csv* file as a table. The output of the algorithm is the set of all the extracted information: the logical information if structures were detected for each of thresholded levels, the radiuses of detected circles, the distances of the lines, the Δ_E values, the estimated anomaly parameters such as d etc.

When the algorithm is finished, it also shows a short statistics in the Matlab environment containing the information about detection efficiency.

Another implementation of the algorithm with the GUI was prototyped in the early development phase. The software cannot process a set of fields, it was implemented as a test environment for the fast scan application development. The main part of the GUI allows to define the input data model (Figure 3.18. It can also create a multiple anomaly field by combining several anomaly models. The noise model can be defined using the other part of the GUI. The last part is the first implementation of the fast scan algorithm with possibility to edit the parameters of the algorithm. This application currently supports the gravity data models (sphere, both cylinders and rectagonal prism) and also the models of simple anomaly for the spontanne polarization. The structure of the application is prepared to add any other existing model or to read the input data from a folder.

The GUI have defined a simple process how to add the new anomaly model without the modification of the original GUI sources. The anomaly generating function must be defined and placed into correct folder in the application sources and it is automatically added to the list of available models when the application starts. The fast scan development part for example shows all the detected structures



Area
Area parameters
Dimension 3d
Length 100
Length step 1
Width 100
Width step 1

Anomaly
Method gravimetry
Type sphere M
Position 28 61 10
Parameters

	weight
Value	1000

Changing parameters of the Anomaly
mesh
X and Y

Current Area
Dimension 3d
Length 100
Width 100
Length step 1
Width step 1

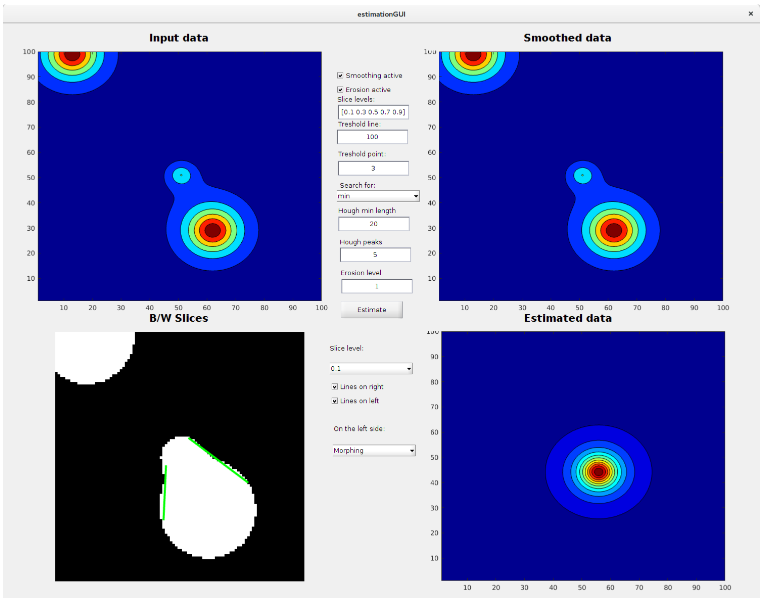
Anomalies

Current Anomaly
Method gravimetry
Type sphere M
Position 28 61 10

weight
1000

Figure 3.18: The GUI of the development environment of the fast scan algorithm - the input data definition.

at the selected levels and other estimated parameters. This part of the GUI was used for example to select the correct noise filter or to set correct input parameters of the Hough transform.



☑ Smoothing active
☑ Erosion active
Slice levels:
[0.1 0.3 0.5 0.7 0.9]
Treshold line:
100
Treshold point:
3
Search for:
min
Hough min length
20
Hough peaks
5
Erosion level
1

Slice level:
0.1
☑ Lines on right
☑ Lines on left
On the left side:
Morphing

Figure 3.19: The fast scan development GUI presenting the original image, the proposed model and the data slices.

3.2.7 Multiple anomaly body in the data

The presented classifier is capable of detecting the most significant anomaly in the field and it will fail on the model depicted in the Figure 3.20. Presented example was created by combining the horizontal cylinder anomaly with two spheres of different total mass. The algorithm will in this case focuse to the biggest field and it will propose a spherical anomaly body.

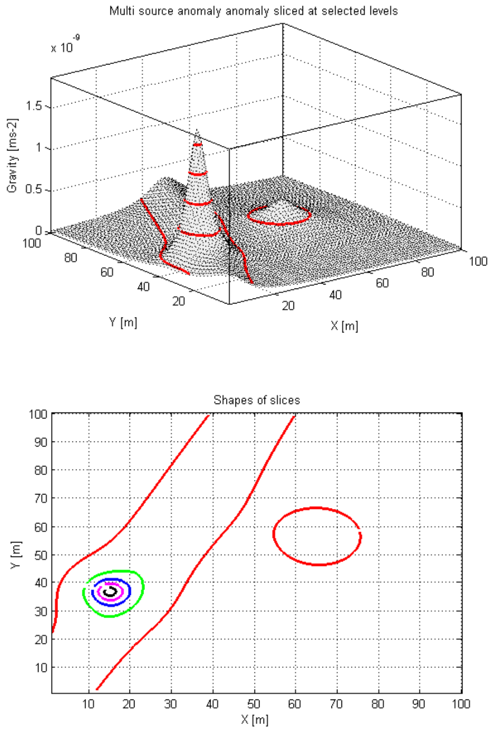


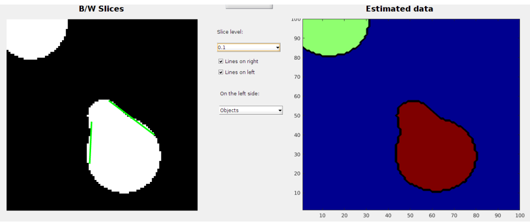
Figure 3.20: The multiple anomaly body - two spheres and a vertical cylinder.

If several horizontal cylinders running in different directions the algorithm will classify the field as containing unknown anomaly because the horizontal cylinder is detected only when all the detected lines run in the same direction. When circle structures are being detected the algorithm focuses only to the biggest circle at each slide. So if multiple anomaly body would be presented in the output, the algorithm would detect the most significant one. The other structures would be ignored. Such situation is modeled in the Figures 3.18 and 3.19. The snapshot of the GUI shows the multiple anomlaly (3 different spheres of similar total mass and depth). The snapshot of the detecting part shows in the right bottom picture the estimated field.

To improve this behaviour the algorithm should be improved for the multiple anomaly: Before the classification starts, it can be tested if more than just one structure is present in the data. For each detected structure such as circular structure the algorithm can focus to the area: at the level 1 (corresponds with the $\frac{1}{N}V_z$) will be detected the lowest shape. Than from the original field will be extracted a square containing just this structure. This part of the field will be processed separately. The same can be done for the parallel line detection: when several pairs of parallel lines are detected, at the level 1 will be detected the appearance of the line and again the area will be extracted from the original field and processed separately.

The current version of the algorithm also presumes that the input contrast density of objects in the model is either positive for all the objects or negative for all the objects. It means that all the input data are either positive or negative values for the whole area. If the input model would be created with bodies with both positive and negative contrast density the input data field could contain both positive and negative values and the normalization procedure would deform the structures.

To prevent such situation, for the future implementation it is planned to detect at the beginning if the input data field contains both positive and negative values. If yes, the whole datafield would be shifted to have only positive values.



Slice level:

0.1

☑ Lines on right

☑ Lines on left

On the left side:

Objects

Figure 3.21: The multiple anomaly body - the separation of objects in slices.

This part of algorithm is not yet implemented in the main application but it was tested in the development GUI. The algorithm now can detect several objects in the slices and can propose how the original area should be divided to focus to separate anomalies in detail. The different anomaly body separation is based on the application of Matlab function *bwlabel*. The output of the function is a matrix containing numbers identifying the separate structures. This matrix is used to propose the new areas of detection.

When the multiple body anomaly separation will be finished, it will be added to the original algorithm and its general schema will modify according the Figure 3.22. The orange box is the part of the algorithm where the detection of multiple body is realized. The most significant anomaly is detected and the algorithm in the next step focuses to a new updated area. The whole process is repeated, including the data normalization to get a new set of slices focused to the second object. This process should be repeated for all the shapes detected at the first run of the algorithm.

The algorithm should detect the position of the anomalies and should classify the type of the anomaly. The depth estimation uses only the structural geometrical information measured in the image and therefore the depth should be estimated with the same accuracy as it is estimated for single anomaly model. The total mass estimation depends also on the value of $(V_z)_{max}$. Therefore the mass will never be estimated with high precission for the multiple anomalies.

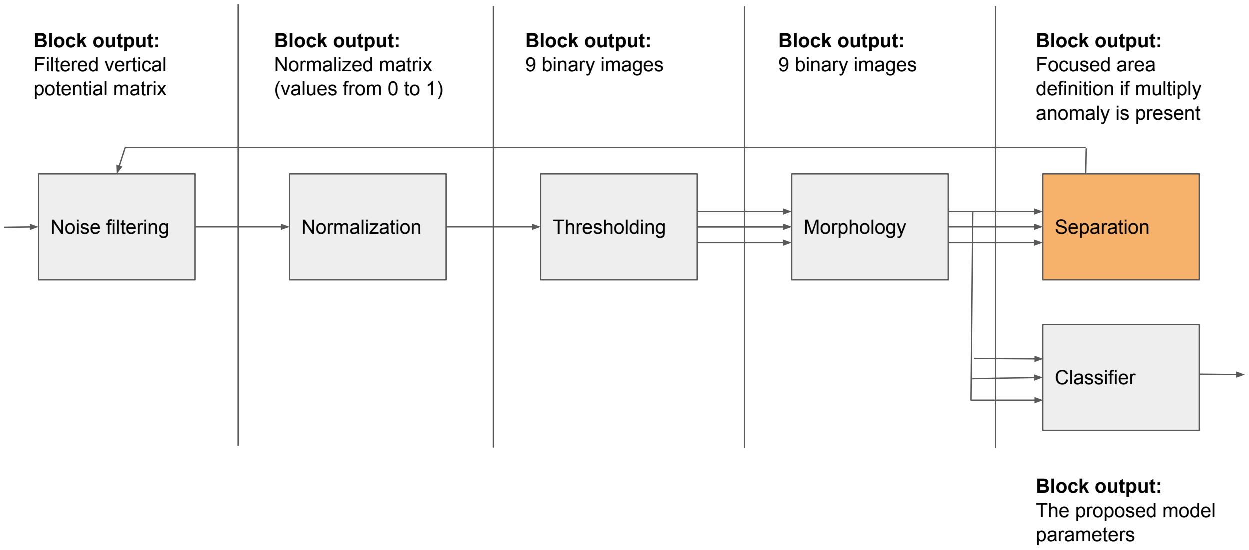


Figure 3.22: The fast scan algorithm updated to detect multiple bodies.

3.3 Anomaly type classification with supervised machine learning

The anomaly type classification described in the previous sections can be done only when an equation defining the anomaly field is available. For the more complicated anomaly body, such as the rectagonal prism, such classification is unreachable. Therefore it was also tested, if a classifier based on the supervised machine learning can be used.

A lot of types of classifiers appeared in last years (see for example a brief reference in [17]). A supervised machine learning is a process where the machine is trained using the input data vectors containing the input features and the output value. In our application, the input values is the information about detected structures and its parameters (number of lines, line angles, line distances, circle radiuses etc.). The output value is the anomaly type. The output of the machine learning process is in this case a function called *classifier* which is used to predict the output value for unknown set of features.

With the set of inputs x_N and outputs y_M the classification function f should convert the input space X to the output space Y. The function f is mostly based on probability and statistics models. Based on the f type, the classifiers are usually divided into following groups:

- **Decision trees**: this type of classifier is very close to the fast scan classifier presented in the previous section. The decision tree creates a tree structure.

The nodes of the tree contain the input features, the leaves of the tree contain the output values. In the node, the input feature are tested and according to the result of the test the decision process selects a next node or a leave. The learning process has to set the test levels to have the best classification. Decision trees are simple classifiers, the learning process is fast, it can be easily implemented and the implementation is not demanding a lot of memory.

- **Discriminant analysis** assumes that different classes generate data based on different Gaussian distributions. To train a classifier, the fitting function estimates the parameters of a Gaussian distribution for each class. The learning process is fast, memory usage depends on the fitting function - if a quadratic function is used, the classifier may need a lot of memory.

- **Support vector machines** (SVM) classify the data by finding the best hyper plane that separates data points of one class from those of the other class. The best hyper plane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the slab parallel to the hyper plane that has no interior data points. Compared to previous classifiers, the SVM is slower and it needs more memory, the implementation can be hard for nonlinear SVM types.

- **Nearest neighbor classifiers** (KNN): the idea is based on the idea that categorizing query points based on their distance to points (or neighbors) in a training dataset can be a simple yet effective way of classifying new points. Various metrics can be used to determine the distance. Given a set X of n points and a distance function, k-nearest neighbor (kNN) search finds the k closest points in X to a query point or set of points. kNN-based algorithms are widely used as benchmark machine learning rules (see [4] for details). The classifier needs a lot of memory, the implementation is hard.

Three versions of the decission tree architecture were used: with maximal number of splits set to 4, 20 and 100. All the trees used the Gini index (see [10] for details). The discriminant analysis succeeded only in the case of linear discriminant. Six types of the SVM were tested: with the linear, quadratic and cubic kernel function (3 types) and three different sizes of the Gaussian kernel (1.9, 7.6, 30). Finally five types of KNN classifiers were tested: the KNN with Euclidean metrics and maximum amount of neighbours set to 1, 10 and 100, the KNN with cosine metrics and 10 neighbours, the KNN with cubic metrics and 10 neighbours (see 4.9 for details).

To test the usability of the supervised machine learning and described classifiers, the output table of the fast scan algorithm was used. The data preprocessing was the same - the data were normalized and converted to a set of binary black and white images. The linear and circular structures were searched in the data. As the input vector were used all the structure describing parameters, the output vector was the initial type information.

The whole statistics reporting the accuracy of the classification on the selected training data sets is presented in the Chapter 4 together with other results.

Several types of classifiers reached better accuracy of classification than the original fast scan classifier. But the advantage of the fast scan classifier is its ability to estimate other anomaly parameters, such as the depth, mass and location of the anomaly. The advantage of the machine learning is a possibility to train the classifier to detect another anomaly type such a rectangular prism.

The given example cannot get dramatically better achievements compared to the fast scan classifier, because it uses the information about the structures identified in the data. The fast scan algorithm fails in the anomaly detection when it is located close to the border of the area - in this case the circular structures are not correctly identified.

The reliability of classification should increase if the classifier would extract such information directly from the image. In this case, each of the pixels in all the black and white slices can be understood as a feature. If all the slices pixels are converted into one vector, it gives a vector of $100 \times 100 \times 9 = 90000$ features. The vectors were created, labeled and stored in one table. Unfortunately, the Matlab environment is not able to use this table to train the classifiers because of the lack of memory. The tests were done on the laptop Lenovo ThinkPad X220 with 8 GB of RAM.

The Matlab is able to keep the table in the memory but it is not able to parse the table for the classifiers learning environment. One of the plans for the future research is to use the Python and its libraries to implement the selected classifiers and to run the classifier training in the cluster with more RAM.

3.4 Fast scan with ANN

Deep neural networks, pioneered in works of Yann LeCun and Geoffrey Hinton, changed and improved the state-of-art in many fields of machine learning. In 2012 Krizhevsky et al. ([18]) proved that deep convolutaional neural networks (CNN) can be trainend as very efficient mage classfiers. From then many other papers were published and CNN become the standard solution for image classification problem.

Convolutional neural networks are enhancement of ordinary Neural Networks. The input is an image and the network architecture preserves some important image features. Typical convolutional network consist the input layer, several convolutional layers, RELU activation layers, pooling layers and finally one or more fully connected layers as the output. The output layer contain the class score for each image. More information about CNN can be found in [37].

As the original idea of the research was to process the geophysical data as images the CNN architecture was selected to test if it is capable of anomaly type classification.

3.4.1 The implemented ANN

The final network architecture for the experiment consist of two convolutional layers and one fully connected layer. Compared to the last published architectures it may look too simple, but as the results shows, it is good enough for presented case. The first convolutional layer has 128 filters of 7 × 7 pixels size with stride 1 × 1. Then there is the RELU activation layer, and the MaxPool layer with stride 2 × 2. After the pooling layer, the data are reduced from 100 × 100 to 50 × 50 shape. Then the second convolutional layer with 256 filters of 5 × 5 size with strided 1 × 1. And once again RELU and MaxPooling. Then there is fully connected layer with 1024 neurons On this layer we used dropout 40 % to prevent overfitting. Last part of the network is the readout layer that classify the data to final class.

For the training 10000 of samples of each anomaly type generated with the algorithms described earlier were used. At first 1000 samples was randomly selected as a test and put it out of the training data. Each remaining sample was first normalized using L2 norm and then two new copies with vertical and horizontal flip of each sample were created. Using this simple data augmentation method 72000 samples was recieved for training and 15000 samples for the validation during the network training.

For the implementation of CNN we used the TensorFlow[1], the open source machine learning library written in Python. Minibatches of size 100 and three epochs of training were used. That means that the network saw each data sample three times. Average training time on Nvidia K5200 GPU was aproximately 20 minutes.

3.5 Monitoring process with structure detection

The section 2.2 gives the definition of the input task for the monitoring process. The input consists of the model seismical data generated from a set of sources and recorded in the set of recievers. The alogirthm should detect any temporal significant temporal change in the data. The configuration of the model and example of the input data is given in the Section 2.2.

The application of the fast scan algorithm in the domain of processing the seismical data during the monitoring process have one big limitation: The original fast scan algorithm uses a forward model to compute the anomaly candidates because the gravity models are very simple and can be fastly computed.

This part of the classifier has to be modified and solved in a different way. One of the possible solution is to precalculate the accepted anomaly situations and to compare the original field with the precalculated data set. The precalculated data can be also used to train a simple classifier. When this thesis is prepared to being published the research is still under process.

The available data models were sampled and converted to black and white figures to estimate the difference between the data. The situation is depicted in the Figure 3.23. On the left side of the image the original input data are presented. On the left side is the model of the tunnel with zero water saturation, on the right side is the tunnel with the high water saturation called wet model. The picture selects the information from the 30th source. The presented waveforms are detected in the 10th, 20th,... 100th reciever.

[1] https://www.tensorflow.org/about/bib

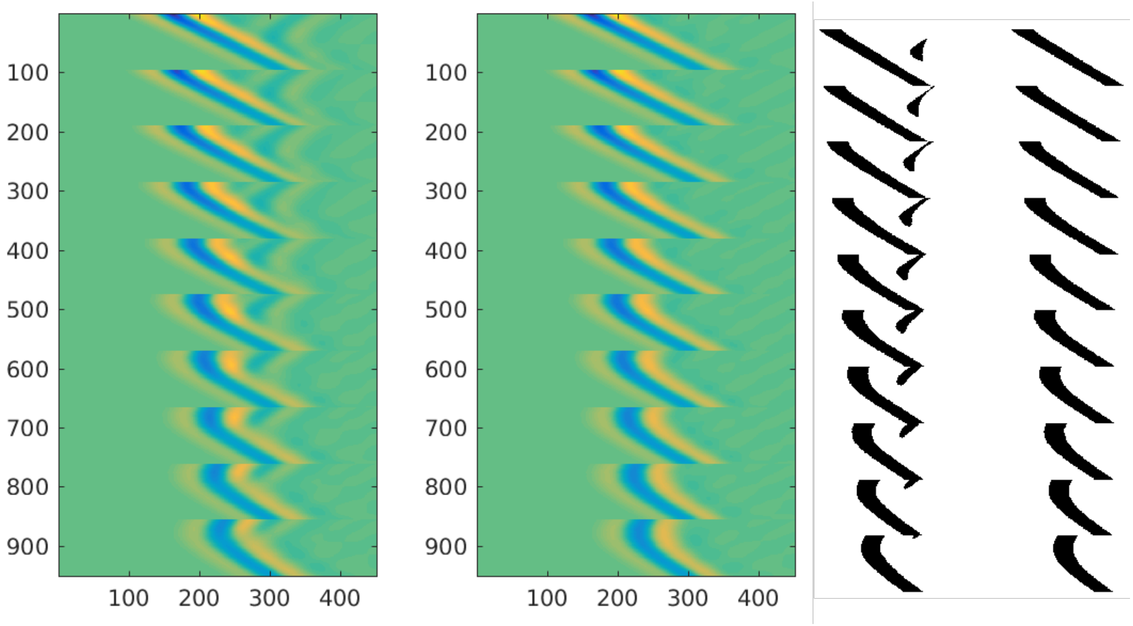


Figure 3.23: The footprint of the dry tunnel in the seismics data (left) and thresholded data (right).

The normalized, morphed and sliced data are presented in the right part of the figure in the same order: first is the dry model, second is the wet model. The presented slice (30 % of the maximum) is selected as the best illustration of the tunnel footprint in the data. The footprint is visible for the dry tunnel, for the wet tunnel it cannot be detected as the wave propagation in the tunnel is very close as in the surrounding rock.

The typical footprint of the dry tunnel is the structure behind the first arrival wave – the focused and thresholded structure is depicted in the Figure 3.24 - the wet tunnel have no footprint (left side), the dry model have typical arcs (right side). The figure uses a different level of thresholding and the different source than the examples in the figure 3.23, therefore the shape of the footprint is different.

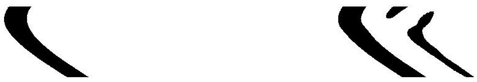


Figure 3.24: The difference between the wet and dry model after the preprocessing.

One of the ideas for the monitoring process is to scan the data for any footprint appearing after the first arrival because the typical mode of operation of the tunnel is the wet model. The structure of the footprint cannot be estimated in advance but typically it will be present as a separate structure after the first arrival. If more than one main arc is detected in the data, it maybe the abnormal condition of the repository. So the classifier may use just two labels: *N* for the normal operation and the *A* for the anomaly situation.

The thresholded and binarized footprint of the dry tunnel contains more than just one object. The detection of the footprint therefore can be based on the object detection and separation. The Figure 3.25 illustrates the separation of the objects in the data based on the Matlab *bwboundaries* function.

Figure 3.25: The separation of the objects in the footprint - wet tunnel (left) and dry tunnel (right).

The shape of the tunnel footprint depends on the source location. The footprint itself is the most expressive in the data from the sources located directly close to the tunnel. The real monitoring application will not have such amount of sources but just one or a few and its location will be important for the detection.

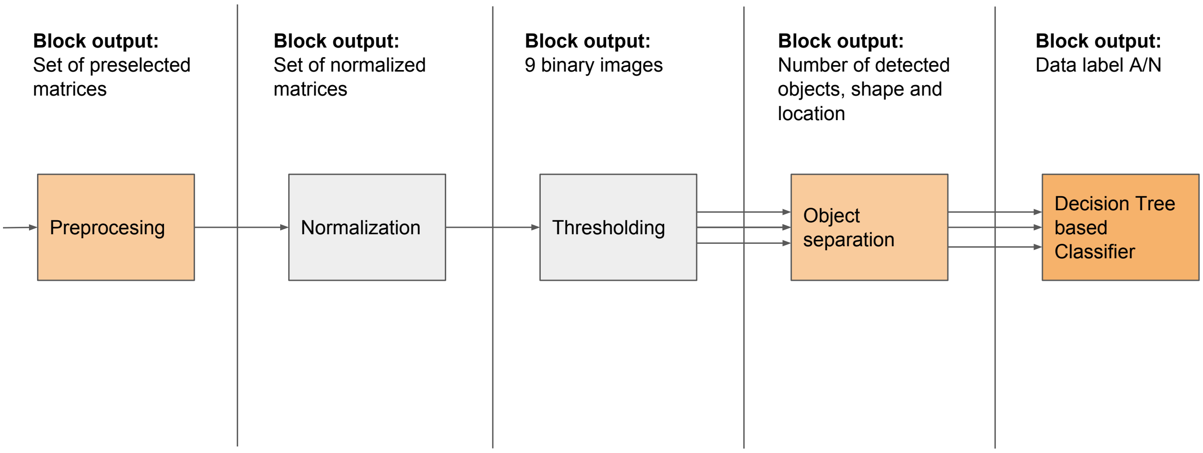


Figure 3.26: The proposed structure of the updated Fast scan algorithm.

The footprint will be present in the data when the humidity of the tunnel will move down. The current plan for the research is to create more models of the tunnel with different water saturation, to extract the footprints, to count the objects and to train the classifier.

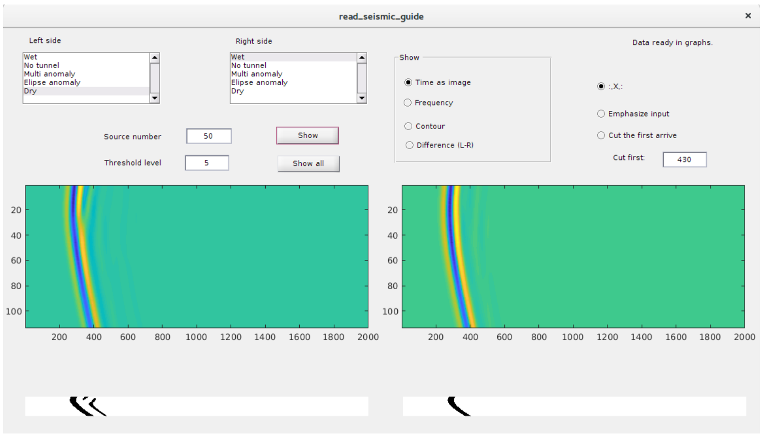
The modified fast scan algorithm will use the information about the detected objects in the slices, its location and size as the main input information. The initial model of the clasifier was set to simple decission tree because the data set seem to be really separated. Because of the lack of examples the neural network will not be used.

The detection of an arc in the waveform data with the Hough transform was already tested for the GPR data ([21]) and it may be tested also for this application. The classification of the normal and anomal operation will be based on the number of the arcs presented in the image. The advantage of the arc detection is the future possibility of the interpretation by measuring the arc parameters. The algorithm again will use the same preprocessing with data normalization and thresholding.

The whole proposed scheme for the updated fast scan algorithm is depicted in the figure 3.26. The initial block called *Preprocessing* selects from the original cube of data the significant matrices containing the data from the sources located close to the tunnel. Next steps are the *Normalization* when the seismics data are converted to image and *Thresholding* where the one input image is converted into the set of slices as it was done in the Fast scan algorithm. The next step is the counting of the objects in each of slices, setting the centers of the detected objects and estimating the shape of the object. The Hough transform here can be used to detect the arcs and measure its parameters. The last step of the algorithm is the classifier with two ouptut labels.

The blocks depicted in the Figure 3.26 are already implemented as standalone functions. The next steps of the research is to connect all the block, to create the final application and to test it with the bigger data set.

As for the fast scan development algorithm a simple GUI in Matlab was created to help with the initial data processing and optimal algorithm selection (Figure 3.27).



read_seismic_guide

Data ready in graphs.

Left side: Wet, No tunnel, Multi anomaly, Elipse anomaly, Dry

Right side: Wet, No tunnel, Multi anomaly, Elipse anomaly, Dry

Source number: 50

Threshold level: 5

Show
● Time as image
○ Frequency
○ Contour
○ Difference (L-R)

● :,X,:
○ Emphasize input
○ Cut the first arrive

Cut first: 430

Figure 3.27: The development GUI for the data preprocessing.

4 The Achievements

This chapter contains all the outputs of the tested algorithms. The presented results were obtained with several anomaly models, all the data sets and detailed results are available on the attached DVD. To get the final statistics for the gravity data and the fast scan algorithm, following data sets were used:

1. **Test set 01** – *ideal models* data with no additional noise, 1000 spheres, 1000 vertical cylinders, 1000 horizontal cylinders.

2. **Test set 02** – *ideal models with low noise* data with additional noise, level 40 dB, 1000 spheres, 1000 vertical cylinders, 1000 horizontal cylinders.

3. **Test set 03** – *ideal models with high level noise* data with additional noise, level 20 dB, 1000 spheres, 1000 vertical cylinders, 1000 horizontal cylinders.

4. **Test set 04** – *ideal models mixed with false data* consists of 100 spheres, 100 vertical cylinders, 100 horizontal cylinders, 100 cubes, 100 random data and 200 false anomalies.

The data sets were created using the models decscribed in the Section 2.1.1 with the script *get_data.py*. The changeable parameters such as the depth of the anomaly body, the surface location and the total mass were for each model generated randomly from the intervals defined in the Table 2.3.

The first three data sets were used to test the classifier ability classify correctly the anomaly type and its parameters. The last data set was used to test, if the classifier is able to set correctly the unknown anomaly body. To train the classifiers, the data set 01 and 04 was used.

The test set 04 was used to verify, if the classifier labels correctly only the data containing any real anomaly model. Therefore another set of mathematical functions were used to create similar data with no real geophysical model. The idea was to create the data looking similar to the anomaly models – a data field with one peak which creates concentric circle structures in thresholded black and white slices of data.

For the first part of the false data, the Equation 2.2 was used with the wrong q and F factors. The defining functions for the two types of tested false data (f_1 and f_2) are given in following equations. The r factor is the surface distance from the anomaly center, the M is the total mass, the z is the depth and the G is the gravitational constant.

$$f_1(r) = \frac{GMz}{\left(r^2+z^2\right)^2} \tag{4.1}$$

$$f_2(r) = \frac{GM}{\left(r^2+z^2\right)^4} \tag{4.2}$$

The tests done with the neural network were done with the data set similar to data set 01. It was generated using the same Python script as the data set 01, only the total number of samples is set to 10 000 to have enough data for the network training.

4.1 Fast scan based on structure detection

4.1.1 Anomaly type classification

The input data set was labeled using 3 labels: *HC* for horizontal cylinder, *S* for sphere, *VC* for vertical cylinder in data sets 01, 02 and 03. In the data set 04, according to the anomaly types included in the data, were added the labels *N* for completely random data, *C* for cubes, *F2* and *F4* for false data with the q factor set to 2 (f_1 in the 4.1) or 4 (the f_2 in the 4.1). The classifier itself uses four labels: *HC*, *S*, *VC* and *O* for all the unclassified data. Ideally, the input labels *N*, *C*, *F2* and *F4* should be in the output labelled as *O*.

The results are depicted at first in tables as confusion matrices. Each row of the tables shows the number and percents of correctly classified anomalies (green cells) and incorrectly classified anomalies (red cells). Each input label has its own row. The number of columns is given by the number of the output labels of the classifier. The confusion table for the data set 01 is presented in the Table 4.1, the data set 02 in the Table 4.3, the data set 03 in the Table 4.4 and finally the confusion matrix of data set 04 is in the Table 4.7. All the presented confusion matrices in this section contain both numbers of observations and percents.

		Detected Type			
		HC	**S**	**VC**	**O**
Input Type	**HC**	976 (97.6 %)	0	0	24 (2.4 %)
	S	1 (0.1 %)	547 (54.7 %)	304 (3.4 %)	148 (14.8 %)
	VC	2 (0.2 %)	0	671 (67.1 %)	327 (32.7 %)

Table 4.1: Confusion Matrix, Test set 01.

What is very noticeable in the first demonstrated classifier output presented in the Table 4.1 is the fact that a lot of spheres were misclassified as vertical cylinder. With deeper look to the classifier output it is significant that the $|\Delta_E|$ value the difference in distance of the estimated spherical and vertical cylinder fields from the original field is always very low in the case of misclassified sphere - $|\Delta_E|$ is from the interval $(1.87 \times 10^{-7} - 7 \times 10^{-4})$. In the case of the real vertical cylinder, the $|\Delta_E|$ is from the interval $(0.006 - 0.92)$.

Therefore the classifier was updated to avoid the misclassification of the spheres. The new criterion was added to the classifier: if $|\Delta_E| < 0.001$, the anomaly is always classified as a sphere. The Table 4.2 contains confusion matrix for updated classifier. All the following results were obtained with updated classifier with the Δ_E check switched on.

		Detected Type			
		HC	**S**	**VC**	**O**
Input Type	**HC**	979 (97.6 %)	0	0	24 (2.4 %)
	S	1 (0.01%)	851 (85.1 %)	0	148 (14.8 %)
	VC	2 (0.02%)	2 (0.02 %)	669 (66.9 %)	327 (32.7 %)

Table 4.2: The data set 01, with noise filter on, with the Δ_E check on.

For the unclassified spheres it is typical that the main peak of the input field is located close to the border of the image and the circle structures are not detected as circles. It means that one of $XPos$ or the $YPos$ value is outside the interval $(15, 85)$. The 85 % of unclassified spheres were mispositioned this way. Rest of the unclassified spheres were too small (the $Radius < 3m$) or too close to the surface $(d < 3m)$.

The worst results were obtained for the vertical cylinders, 32.7 % was not correctly classified, because not enough circle structures was detected in the pictures. The situation is quite analogical to the sphere classification. The peak of the anomaly field is more flat compared to the spheres and therefore more structures remain undetected. When the classifier output was analyzed, it was found that for the 80 % of the unclassified vertical cylinders were located close to the border of the area – the $XPos$ or the $YPos$ value was outside the interval $(16, 84)$.

In the case of vertical cylinder classification is also important the depth of the anomaly body. If the vertical cylinder is located far from the area border and it is not classified correctly, mostly it is located deeper than 25 meters, the majority was lower than 40 m as it is depicted in the histogram in the Figure 4.1. The Figure 4.2 shows the error of the detection as a function of the anomaly body depth. The mispositioned unclassified models were removed to demonstrate that with increasing depth the reliability of classification goes down. For the selected density contrast the critical depth is 30 m. Value 1 means correctly classified cylinder, value 0 stands for the unclassified cylinder. This can be in the future improved by modification of the structure detection part the same way as it is proposed for spheres: the algorithm should detect not only the whole circle structures, but arcs as well.

If data are combined with the noise, the reliability of the classification goes down,

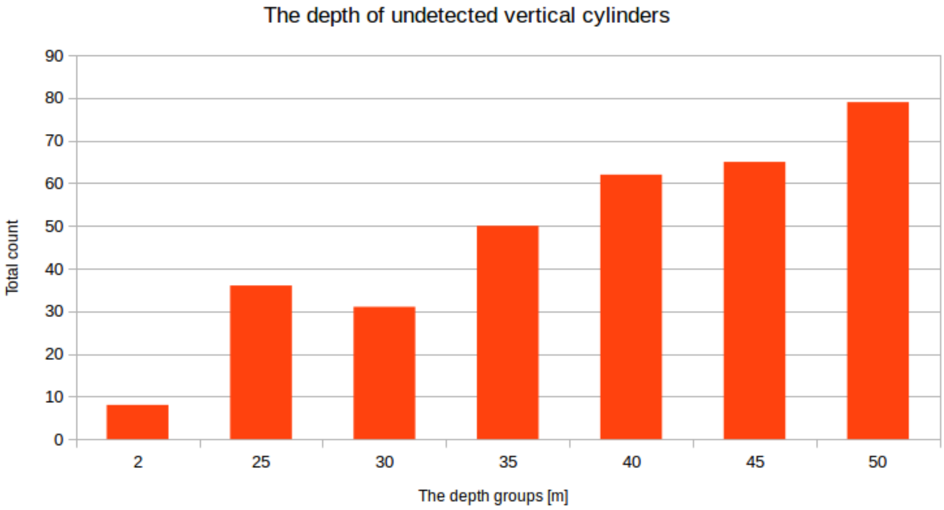


Figure 4.1: The correctly located unclassified vertical cylinder and the input depth of the anomaly.

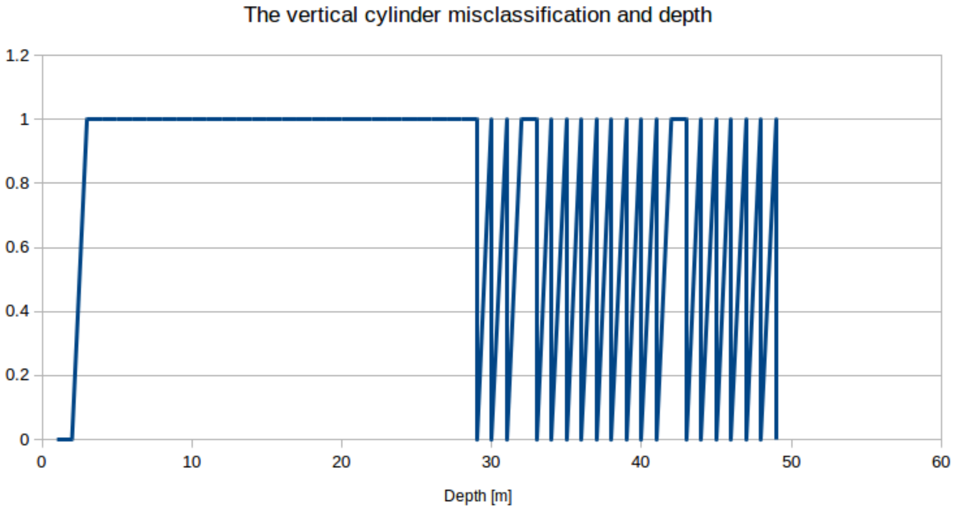


Figure 4.2: The relation between the successfully anomaly classification and anomaly depth for the vertical cylinder.

even if the SNR is quite high as it is demonstrated in the Table 4.3 for the SNR 40 dB. The detection of horizontal cylinders is still good, the reliability of sphere detection goes from 85 % to 75 % and 60 % of vertical cylinders remain unclassified. The Table 4.4 shows that when the noise level increases, the classification reliability falls dramatically down. The detection of the linear structures is still quite good, only 8 % of models are not classified as the horizontal cylinder, because no lines are detected in the data. In the case of spherical body, the reliability goes down from 85 % to 43 % and only 28 % of vertical cylinders are classified correctly.

The main reason is the uncertainty of the X and Y coordinates. Due to the noise the detected circular structures have bigger variance in the position of the central point and the circle radius. The vector of estimated values V_z at the borders of the circle have also bigger variation and the depth parameter is set with less accuracy. The estimated field at the end have biggest distance from the original field and the algorithm can find a model of vertical cylinder which is closer to the original data.

		Detected Type			
		HC	**S**	**VC**	**O**
Input Type	**HC**	945 (94.5 %)	0	0	55 (5.5 %)
	S	0	848 (84.8 %)	0	151 (15.1 %)
	VC	3 (0.3 %)	1 1 (0.01 %)	617 (61.7 %)	379 (37.9 %)

Table 4.3: Confusion Matrix, Test set 02, the noise was filtered using the Wiener filter.

		Detected Type			
		HC	**S**	**VC**	**O**
Input Type	**HC**	918 (91.8 %)	0	0	82 (8.2 %)
	S	0	437 (43.7 %)	137 (13.7 %)	425 (42.5 %)
	VC	0	0	280 (28.0 %)	720 (72.0 %)

Table 4.4: Confusion Matrix, Test set 03, the noise was filtered using the Wiener filter.

The confusion matrix in the Table 4.5 shows the classifier output on data set 02 without the noise filters to demonstrate the importance of the noise filtering.

Even if the input consists of smooth data or data with the additional noise, the algorithm detects the horizontal cylinder with high precision, worse results are obtained for spherical anomalies and the worst result the algorithm gives for the vertical cylinder anomaly. When the input data sets were analyzed, all the undetected horizontal cylinders were located so close to the border of the modeled area

		Detected Type			
		HC	**S**	**VC**	**O**
Input Type	**HC**	944 (94.4 %)	0	0	56 (5.6 %)
	S	0	748 (74.8 %)	0	252 (25.2 %)
	VC	10 (1.0 %)	0	399 (39.9 %)	600 (60.0 %)

Table 4.5: Confusion Matrix, Test set 02, without the noise filtering.

		Detected Type			
		HC	**S**	**VC**	**O**
Input Type	**HC**	899 (89.9 %)	0	0	101 (10.1 %)
	S	0	270 (27.0 %)	137 (13.7 %)	592 (59.2 %)
	VC	0	0	197 (19.7 %)	803 (83 %)

Table 4.6: Confusion Matrix, Test set 03, without the noise filtering.

that only one line structure remains in the data, as it was described in the Section 3.2.1, Figure 3.6.

4.1.2 The classifier and false data

All the above presented results were computed using the test sets 01, 02 and 03. Those sets contain only valid data – a mix of real models of anomalies. Therefore another data set was created. Random data were used to verify, that all the problems described at circle structure recognition and parallel line recognition were solved correctly (see the Figures 3.7, 3.9).

The cube model was used to have the model of the field with the peak, which sometimes creates a circle like structures in the slices, sometimes it is presented as pairs of parallel lines. Such model tests the precision of the circular structure detection as well as the precision of the parallel line detection.

If the data slices contain a concentric circles, the algorithm should always try to find a model of the vertical cylinder or a spherical anomaly. As the gravity interpretation is an ambiguous task, a model which fits into proposed data may be found anyway. Therefore the sets called False1 and False2 were created. As it was expected, the algorithm really tries to fit a sphere or a vertical cylinder to such type of input data. For the selected false data it is typical, that the proposed total contrast mass of the anomaly candidate is small (typically just 1-10 kg per cubic m).

So the algorithm has new optional parameter: a limit contrast mass which is acceptable for the model of sphere and cylinder. The user of the algorithm should set this parameter according to accepted density contrast in the searched area and according to the experience.

Table 4.7 presents the confusion matrix of data set 04, with the mass check parameter switched on. The number of samples of each anomaly type was 100, therefore only percents are written to the table, to omit duplicity values.

		Detected Type			
		HC	**S**	**VC**	**O**
Input Type	**HC**	96 %	0	0	5 %
	S	0	85 %	0	15 %
	VC	0	0	69 %	31 %
	N	0	0	0	100 %
	C	0	0	0	100 %
	F1	0	0	0	100 %
	F2	0	0	0	100 %

Table 4.7: Confusion matrix for data set 04, Fast scan algorithm.

4.1.3 Anomaly location estimation

When the input data is labeled as *S*, *VC* or *HC* its parameters may be estimated. At first the coordinates of the anomaly center ($[X, Y]$) or the central line (given by $[X1, Y1]$ and $[X2, Y2]$) are estimated. Secondary the depth d is estimated and last is the estimation of the total mass of the anomaly.

For the spherical body and vertical cylinder, the algorithm gives a set of central points of detected circles at the data slices. The final values for $[X, Y]$ are set as the median value of the vector of detected X and Y. The median works better than a mean value, because typically the first detected circle is shifted from the original position. The algorithm also checks, where is located the maximum value in the picture and if the coordinate of the maximum value is close or identical with the estimated $[X, Y]$. The coordinates of the maximum value in the original data can be used as the $[X, Y]$ only if there is no noise in the data. Otherwise, due to the noise, the maximum point can be shifted from the anomaly center.

Figures 4.3 and 4.4 illustrate the precision of the detection of the central point of the spherical anomaly for the smooth data without any noise. Both images show the histogram of the difference between the input value and the estimated value. The position of the anomaly is set with high precision, the bigger error appears for models located very close to the image borders.

Figures 4.5 and 4.6 show the histograms for the $XPos$ and $YPos$ estimation for the vertical cylinder.

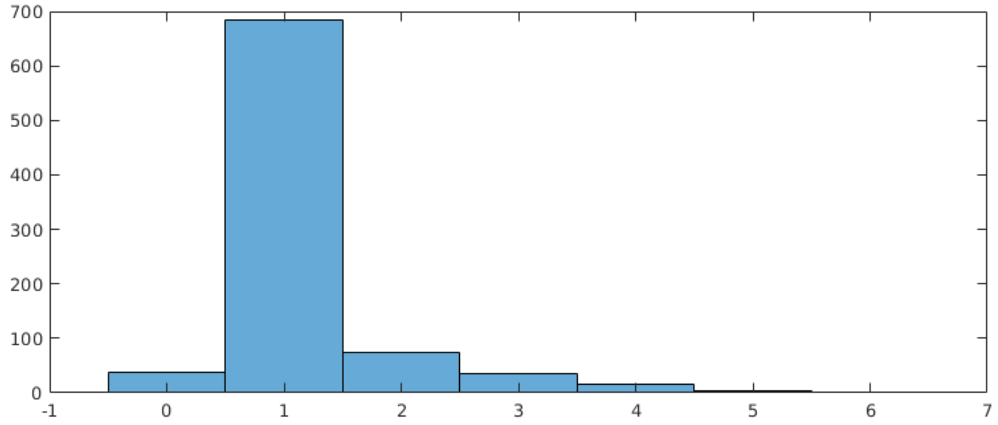


Figure 4.3: The histogram of the difference between the input XPos and the estimated value, the spheres.

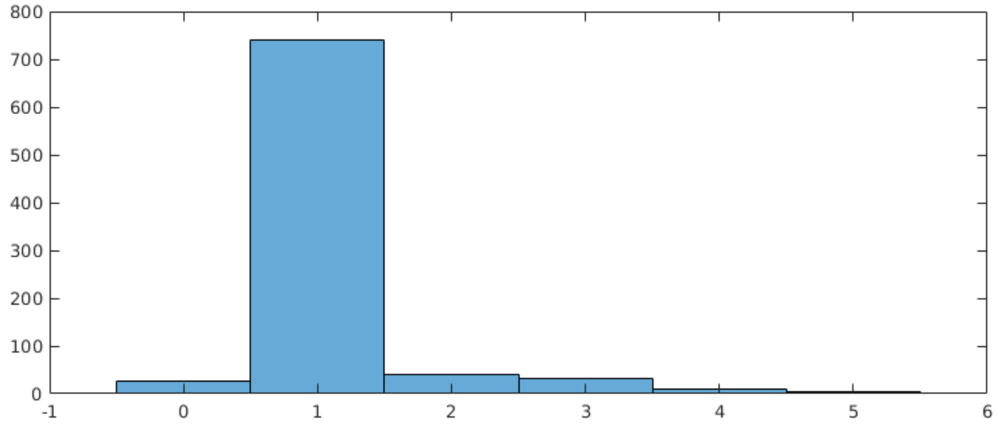


Figure 4.4: The histogram of the difference between the input YPos and the estimated value, the spheres.

What is interesting in all the Figures 4.3, 4.4, 4.5 and 4.6 is the fact that mostly the $XPos$ and $YPos$ are set 1 point higher than the input value. It is caused by the rounding prices during the $XPos$ and $YPos$ estimation.

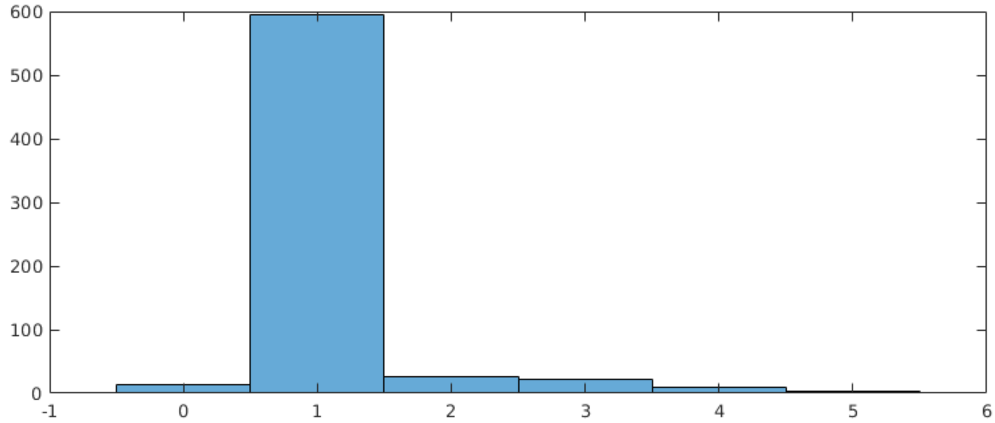


Figure 4.5: The X coordinate estimation error, vertical cylinder anomalies, set 01, the difference between the original and estimated value.

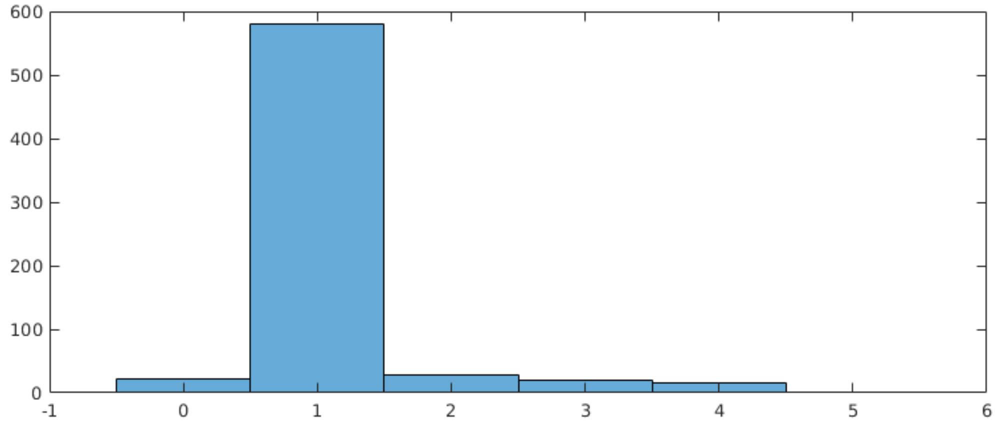


Figure 4.6: The Y coordinate estimation error, vertical cylinder anomalies, set 01, the difference between the original and estimated value.

The evaluation of the accuracy of the determination of the central line is a little bit more complicated. The input is given by 2 randomly selected points, the outputs comes in the form of pair $[rho, \theta]$. To first idea how to compare the position of both lines, was to calculate the intersections of the input and output line with each the axis x and y and compare it. For the input line, the $X0_{in}$ and $Y0_{in}$ was computed using the general line equation.

If the classifier finds a pair of parallel lines with the θ and $rho1$ and $rho2$ values, the rho value of the central line is given by:

$$rho = min(rho1, rho2) + \frac{|rho1 - rho2|}{2} \tag{4.3}$$

To set the $X0_{out}$ and the $Y0_{out}$, the Equation 3.10 of Hough transform was solved with x set to 0 to get $Y0_{out}$ and y set to 0 to get the $X0_{out}$.

The comparison of the accuracy of estimating the intersection points is a good idea, when the original line is not parallel or nearly parallel to the image border. In such case even when the main line angle to the image borders is estimated with good precision, the difference between the original and estimated intersection value can be really high as it is demonstrated in the Figure 4.7. On the left in the image is the standard situation with line not parallel to the border. On the right side of the image is presented the situation when a small inaccuracy of the line angle estimation leads to a big difference in the Y axis line intersections.

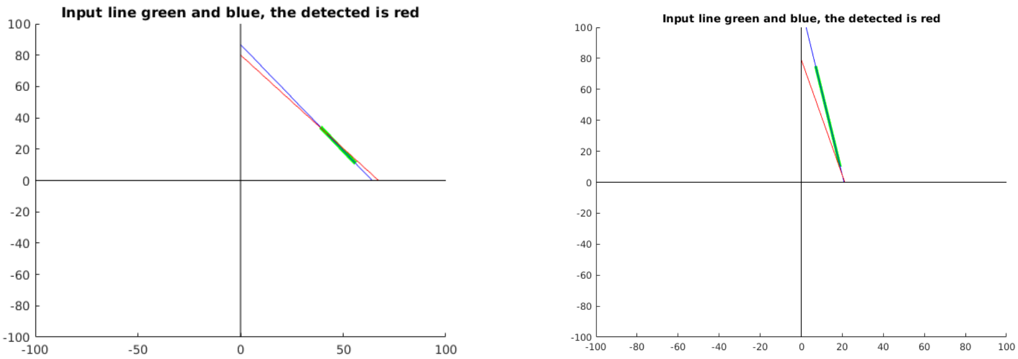


Figure 4.7: The similar precision of the angle detection, but differences in the intersection estimation. The left side presents the situation where the estimation error of the angle and intersection on the axis is similar. On the right side is depicted the situation where even small error in line angle detection results in big difference in original and detected intersection on the vertical axis.

Therefore if was finally implemented in reverse: for the input line the value of θ and rho were calculated using the calculated pair $[X0_{in}, Y0_{in}]$. The Equation 3.10 was used tto calcluate the θ_{in} and the rho_{in} using the substitution of intersection points of the input line $[X0_{in}, 0]$ and $[0, Y0_{in}]$ as it is demonstrated in the Equation 4.5 and the Equation 4.4.

$$\theta_{in} = arctg\left(\frac{X0_{in}}{Y0_{in}}\right) \tag{4.4}$$

$$rho_{in} = X0_{in}cos\left(arctg\left(\frac{X0_{in}}{Y0_{in}}\right)\right) \tag{4.5}$$

The precision of the θ and rho estimation is again illustrated using the difference of the original and estimated value related to the original value (see Figures 4.8 for the θ and 4.9 for the rho). As the rho value increases, the estimation error of θ increases as well. The original input data set was sorted at first by the input θ value and second by the rho. The Figure 4.8 shows that the error of the θ estimation is not related to the input theta value. When the parallel lines are detected, the θ is always estimated with acceptable precision with average relative error 1.21 %.

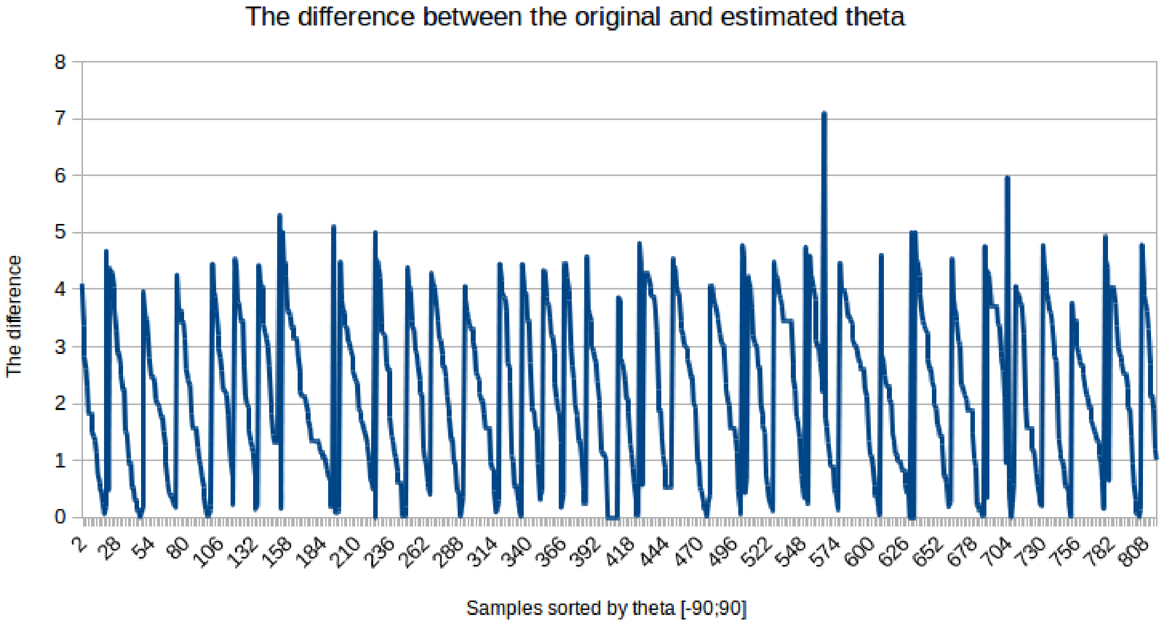


Figure 4.8: The difference of the input and the estimated θ value as a function of the input theta.

In the Figure 4.9 is the error of estimation distributed randomly. But if the output data set is sorted by the input anomaly depth, one can easily see that the error increases with the depth (Figure 4.10). The depth is not the only factor of the rho estimation precision – the second factor is the total mass of the anomaly. The worst results are obtained when the total mass is quite small and anomaly is located deep under the surface. The detected lines are very close to each other, the difference of estimated rho_1 and rho_2 is small and the total estimation error is high. Compared to the rho, the θ estimation accuracy is depth independent.

The average relative error of the rho estimation for the presented data set is 15 %. When data samples with depth greater than 45 m are removed from the data set, the average error is 11 %.

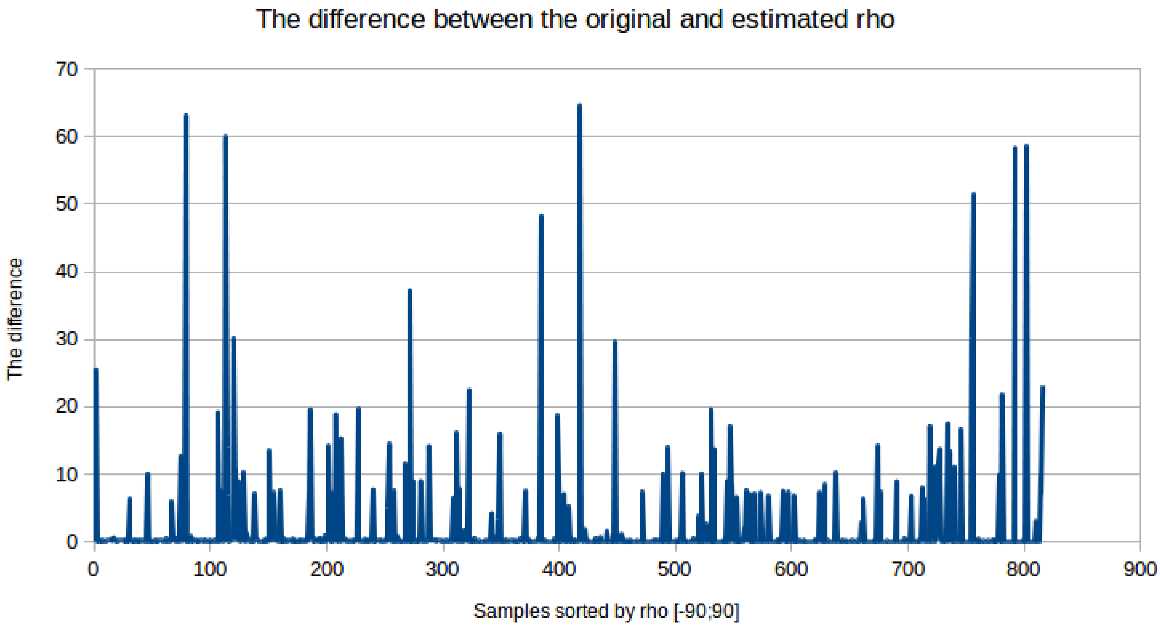


Figure 4.9: The difference of the input and the estimated *rho* value as a function of the input rho.

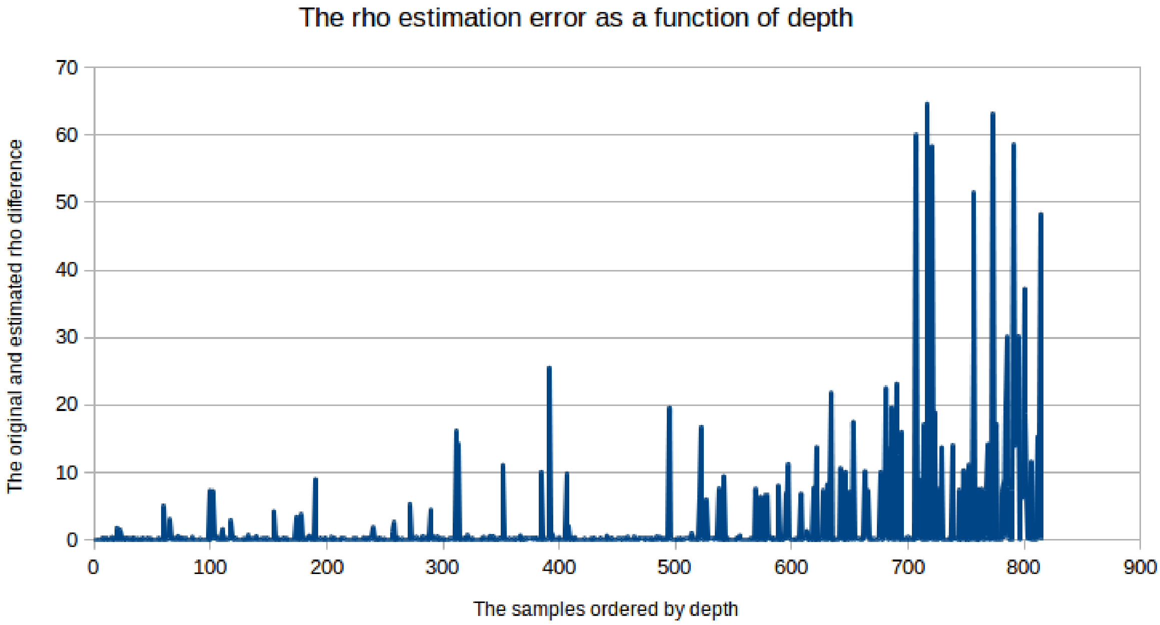


Figure 4.10: The difference of the input and estimated *rho* value estimation as a function of depth.

4.1.4 Anomaly depth estimation

Due to the rounding and the errors in the estimation of the $XPos$ and $YPos$ values the estimated depth is never precise as well as the values of the vector r_N. For all the used anomaly bodies, the depth is always estimated lower than was the original value. The table 4.8 summarizes the average relative error of the depth estimation.

Anomaly body	Depth relative error
Sphere	21 %
Vertical cylinder	30 %
Horizontal cylinder	21 %

Table 4.8: The average depth estimation relative error.

For the detailed information about the precision of the depth estimation following figures were inserted: Figure 4.11 shows the difference between the input depth and the estimated value for the sphere as a function of the input depth value. Figure 4.12 shows the relative error of the depth estimation as the function of the input depth value. Comparing the two figures one can easily see that the relative error of the depth estimation is not related to the original anomaly body depth. The input data models used the depth value from the interval of 1 m to 50 m.

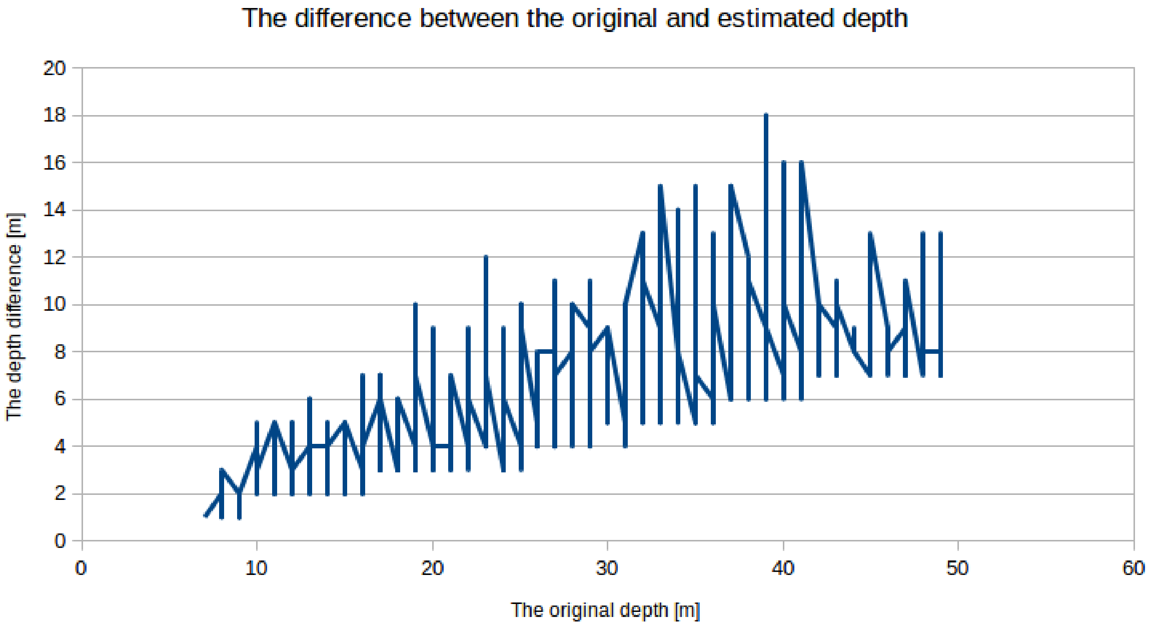


Figure 4.11: The depth estimation error, spherical anomalies, set 01, the difference between the original and estimated value.

The relative error of the depth estimation increases, if the anomaly body center is located close to the image border. When the Figures 4.11 and the 4.12 were created, the output data table was sorted first by the relative error value, second by the $XPos$ value and third by the $YPos$ value. The lowest relative error is obtained, when $XPos$ and $YPos$ are located in the interval $(15, 85)$, because in this case even increases the error of the estimation of the values of $XPos$, $YPos$ and the vector of r_N.

The described data sorting is the cause of the saw-like effect in the figures. The higher frequency of the curve fluctuations is caused by the $YPos$ values sorting, the lower frequency of the curve fluctuation is set by the $XPos$ sorting.

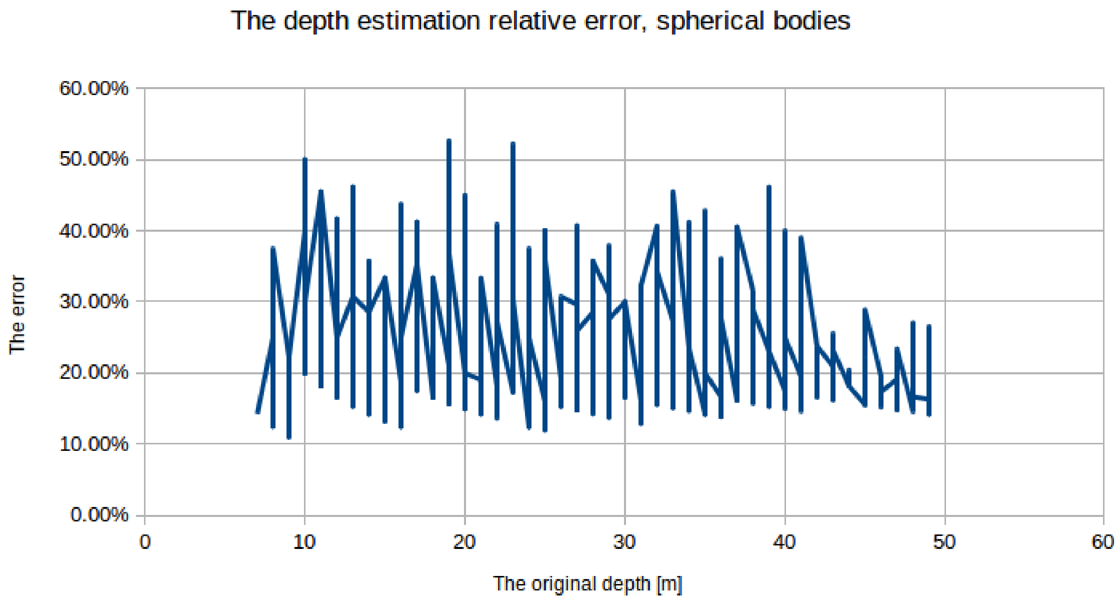


Figure 4.12: The depth estimation error, spherical anomalies, set 01, the relative difference between the original and estimated value.

The histogram presented in the Figure 4.13 shows the histogram of the relative error of the depth estimation of the sphere. The mean value of the error is 15 %. Mostly all the anomalies with depth relative error estimation higher than 30 % are located close to the original area border.

The depth estimation precision for the vertical cylinder is in general very similar as it is described for the spherical bodies. The only difference is that the relative error increases with the depth of the anomaly. The Figure 4.14 contains the difference between the original and estimated depth for the data set 01. The Figure 4.15 contains the graph for the relative error of the depth estimation. The saw-like effect is again presented in both the figures for the same reasons as it was described for spherical bodies. The relative error is quite high at the border of the area when the anomaly body depth is less than 20, and slightly increases as the depth is greater than 40.

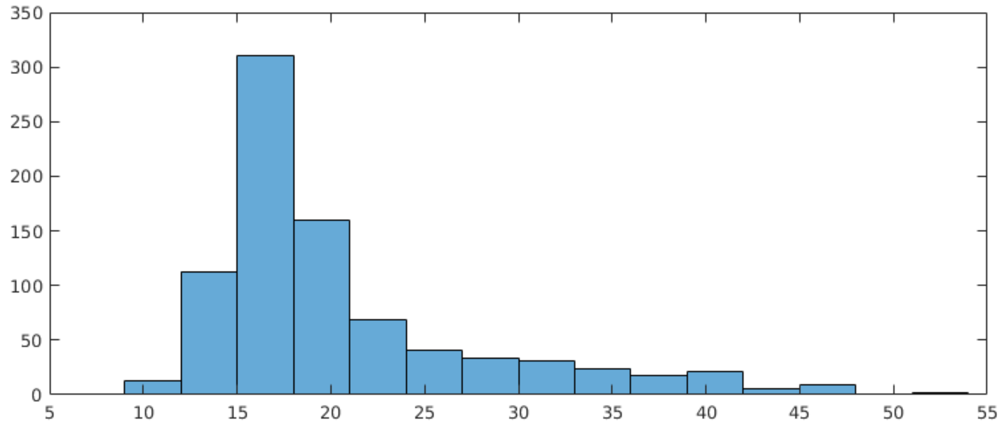


Figure 4.13: The histogram of the depth estimation relative error, spherical anomalies, set 01.

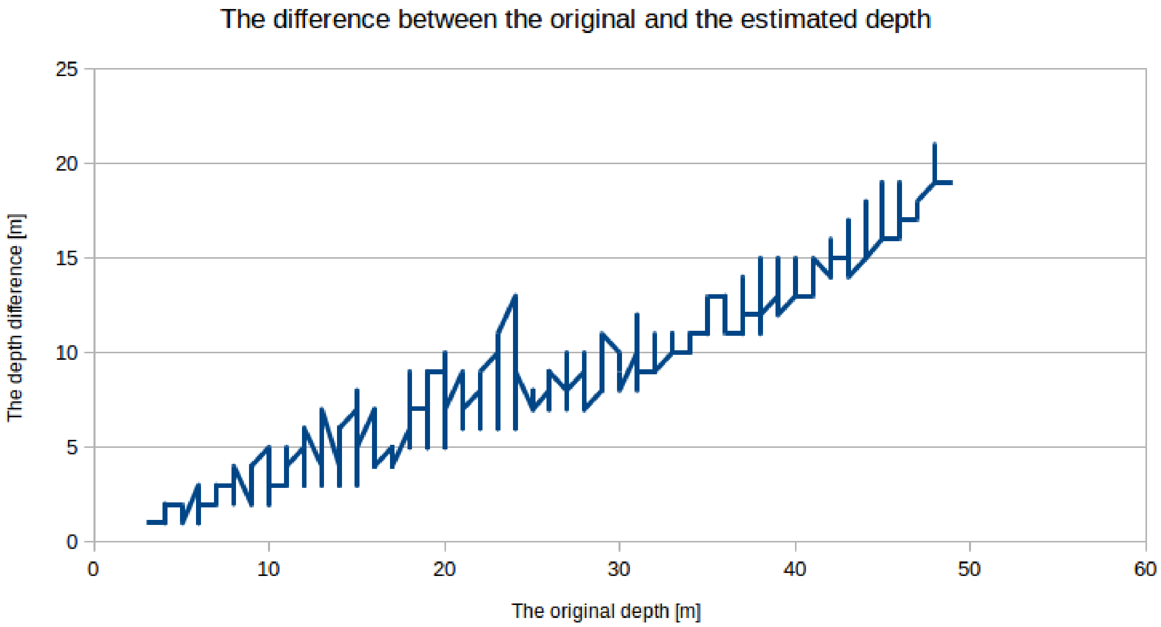


Figure 4.14: The depth estimation error, vertical cylinder anomalies, set 01, the difference between the original and estimated value.

In the Figure 4.15 it can be surprising that for the small depth the relative error have noticeable fluctuation for the depth under 30 m and for the deeper located bodies the relative error fluctuation is smaller. The accuracy of the depth estimation depends on the anomaly body location. As it was described, when the anomaly is located close to the area center the depth estimation have smaller relative error compared to the body located at the border of the area.

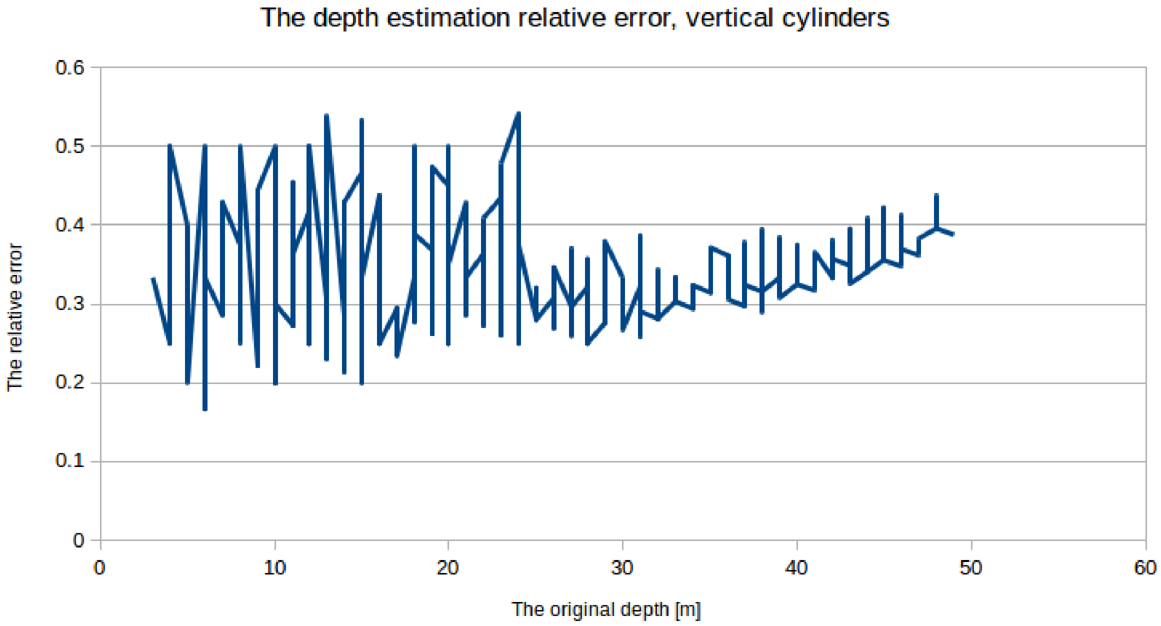


Figure 4.15: The depth estimation error, vertical cylinder anomalies, set 01, the relative difference between the original and estimated value.

The depth is estimated only when the input data set is labeled by the classifier as *S*, *VC* or *HC*. It was already described that for the depth bigger than 30 m, the reliability of the classifier is low and all the bodies located close to the area border remain unclassified and the depth is not estimated at all. This is the reason of the lower fluctuation of the relative depth estimation in the Figure 4.15 when the depth of the anomaly body is greater than 30 m.

The histogram of the relative errors of the depth estimation for vertical cylinder is presented in Figure 4.16. The median value of the relative error is very close to the average value of the relative error. For the models where the relative error of the depth estimation is greater than 40 % is typical the location close to the area border as it was described for the sphere.

The fact that the depth of the sphere is estimated with higher precision may be explained by the different nature of both anomaly bodies. The total mass of the sphere is located very close to the anomaly center and therefore it can be estimated with higher precision. The body of the vertical cylinder is modelled as semi infinite, the gravity field is flatter and the circle structures have bigger radius and therefore the circles are detected only at the sampling levels close to the maximum value. The

r_N vector contain less data and the error in the estimation of one of the values in r_N is therefore more significant than in the case of the spherical anomaly body.

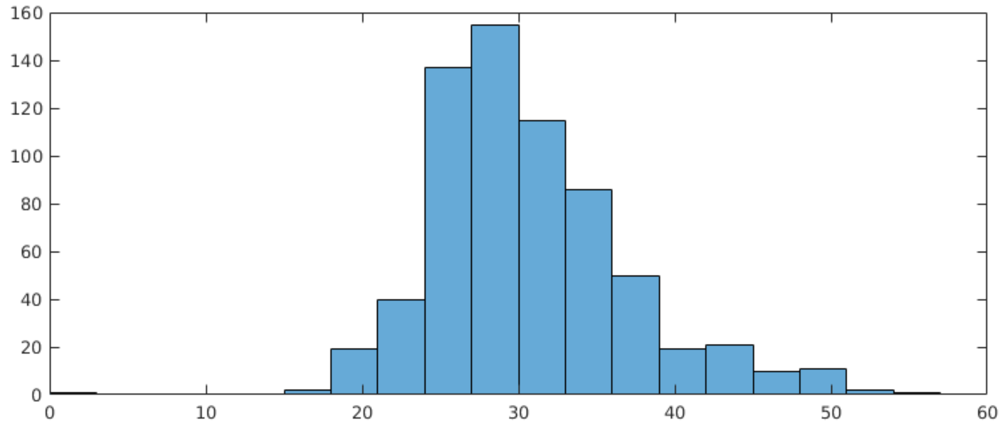


Figure 4.16: The histogram of the depth estimation relative error, vertical cylinders, set 01.

The average depth estimation relative error for the horizontal cylinders is 21 %. The value is close to the relative error of the depth estimation of the sphere. The shape of the field across the horizontally lying cylinder is similar as in the case of the sphere. Therefore the sensitivity to the estimation of the r_N values and the central line position is similar. The difference of the input depth and the estimated value is depicted in the Figure 4.17 and the relative error of the depth estimation is presented in the 4.18.

Compared to the previous cases the depth to relative error dependency is different. The relative error is greater when the body is located close to the surface or in higher depth. The histogram of the relative depth estimation error is in the Figure 4.19. The median value of the depth estimation error is close to the mean value.

The first estimated parameter is the depth of the anomaly. The total mass estimation precision depends on the precision of the depth estimation. For the spherical bodies the dependency is cubical and for the other two bodies it is linear. The estimated total mass using the formulas listed in the table 3.1 have the expected relative error as it is demonstrated in the Figure 4.20 for the vertical cylinders. In the case of other bodies, the results are similar.

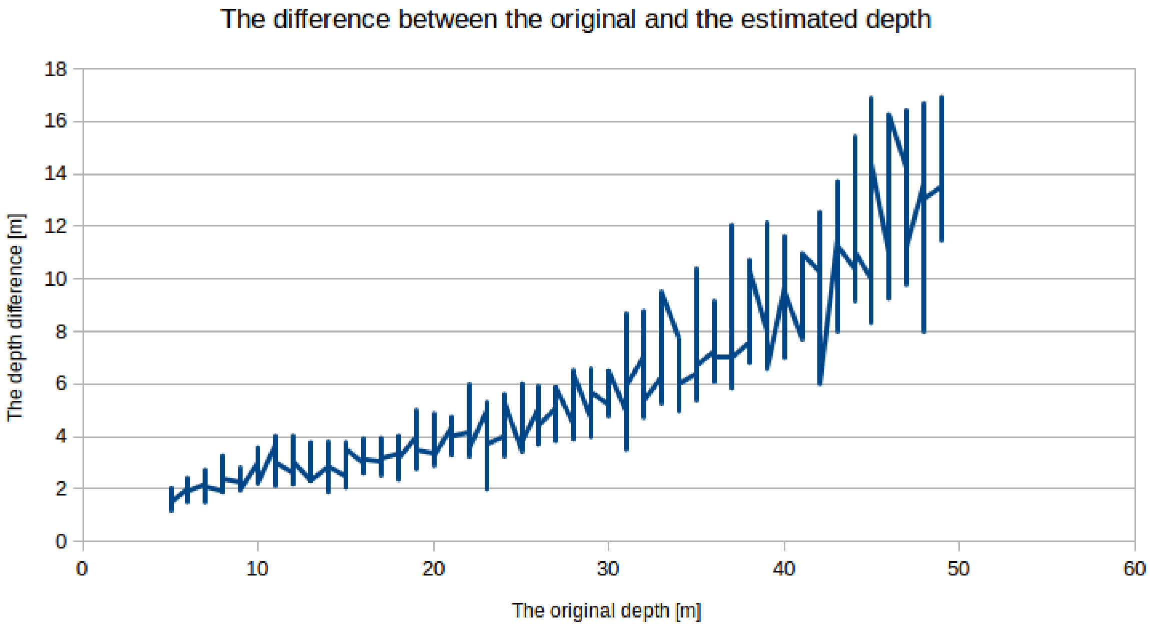


Figure 4.17: The depth estimation error, horizontal cylinder anomalies, set 01, the difference between the original and estimated value.

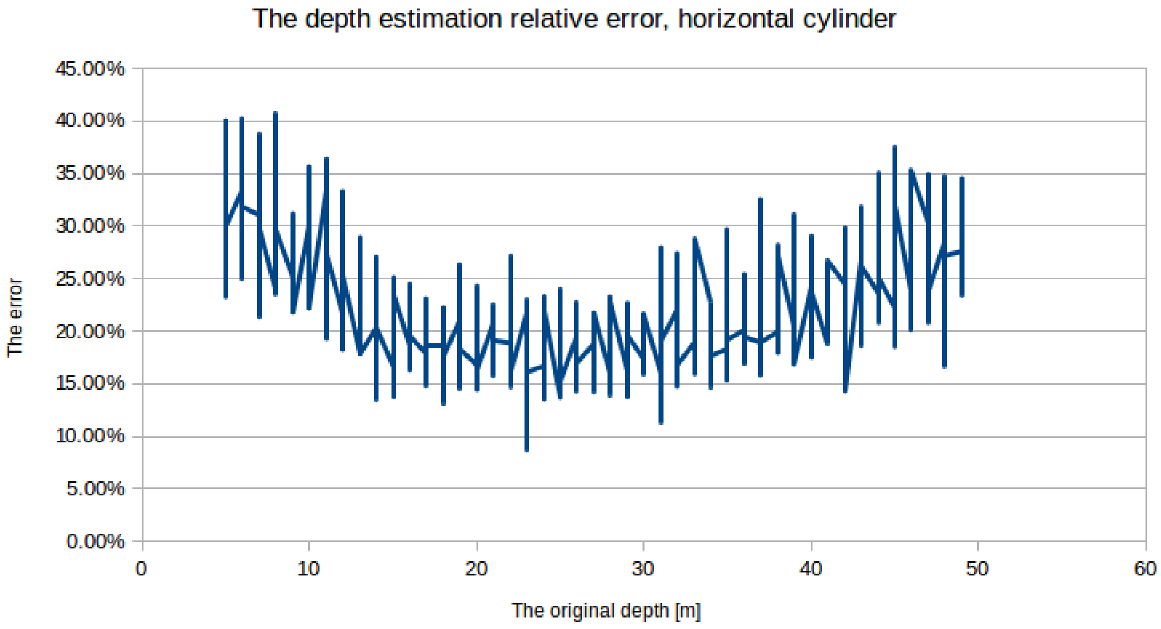


Figure 4.18: The depth estimation error, horizontal cylinder anomalies, set 01, the relative difference between the original and estimated value.

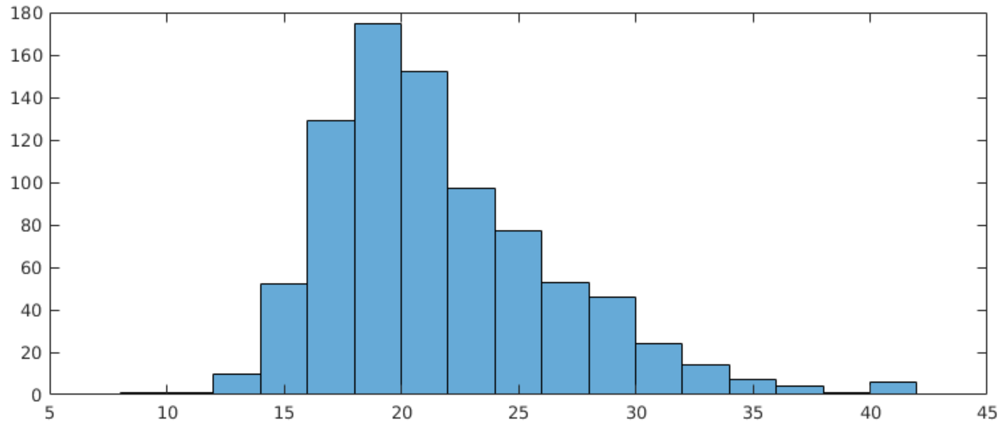


Figure 4.19: The histogram of the depth estimation relative error, horizontal cylinders, set 01.

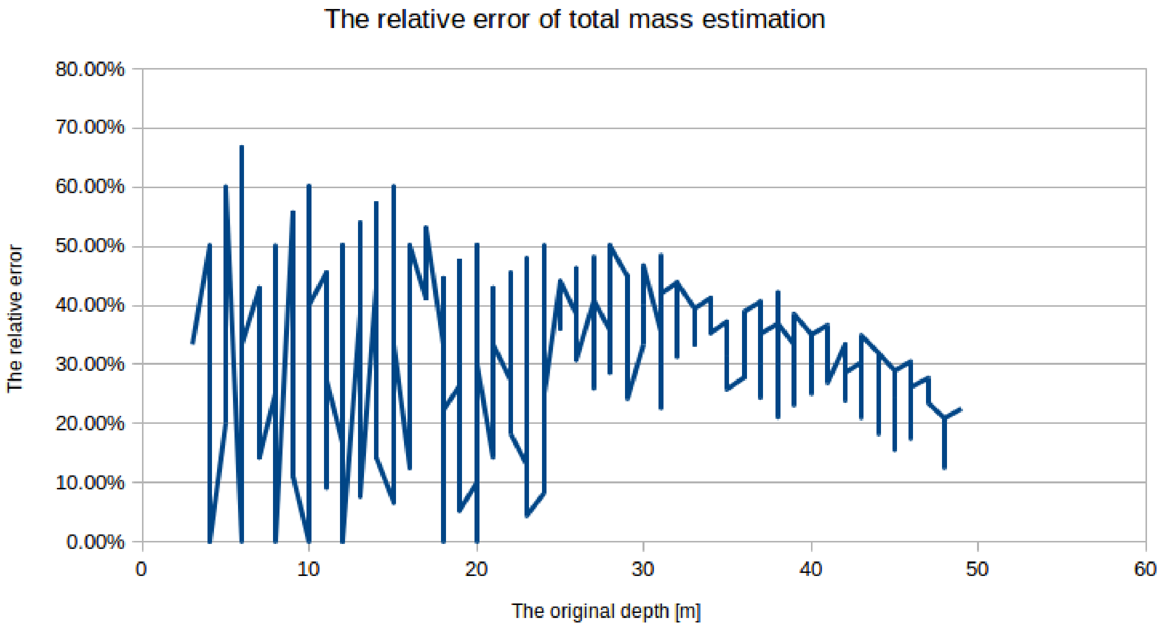


Figure 4.20: The mass estimation error, vertical cylinder anomalies, set 01, the difference between the original and estimated value.

4.2 Anomaly type classification with supervised machine learning

From the selected set of tested classifiers only a few of types were unsuccessful. The determinant based classifiers failed in training and gave no outputs. The SVM with the smallest Gauss kernel got the worst total reliability (49,1 %). Other types of classifiers were able to classify the data with more than 95 % reliability.

The Table 4.9 summarizes what types of classifiers were used in the tests and how the parameters of selected models were set.

General type	**Subtype**	**Parameters**
Tree	Complex	Maximum number of splits: 100, Gini index
	Medium	Maximum number of splits: 20, Gini index
	Simple	Maximum number of splits: 4, Gini index
SVM	Linear	Kernel function: linear, Kernel scale: automatic
	Quad	Kernel function: quadratic, Kernel scale: automatic
	Cubic	Kernel function: cubic, Kernel scale: automatic
	FineGauss	Kernel function: Gaussian, Kernel scale: 1.8
	MediumGauss	Kernel function: Gaussian, Kernel scale: 7.1
	CoarseGauss	Kernel function: Gaussian, Kernel scale: 29
KNN	Fine	Metrics: Euclidean, Neighbours: 1
	Medium	Metrics: Euclidean, Neighbours: 10
	Coarse	Metrics: Euclidean, Neighbours: 100
	Cosine	Metrics: Cosine, Neighbours: 10
	Cubic	Metrics: Cubic, Neighbours: 10

Table 4.9: The set of successfully trained classifiers.

The classifier outputs are labeled as S, HC and VC as each of the classifier models uses only the labels from the input data. The Table 4.10 summarizes all the outputs of the classifiers. The AVG column stands for the average value of reliability, the other columns contain reliability of the classification of corresponding labels.

Compared to the Fast scan classifier, the results of the standard machine learning classifiers are better. The disadvantage is that such type of classifier is not able to estimate the anomaly parameters. The robustness of the classification of all the models should be tested using the false data together with the real anomaly models.

Because all of the trained models have very similar reliability, the selection of the model should be done regarding the algorithm memory consumption and implementation complexity. The less memory consuming type of the classifier is the

Type	**Subtype**	**HC**	**S**	**VC**	**Average**
Tree	Complex	99.80 %	99.40 %	96.20 %	97.3 %
	Medium	99.80 %	97.00 %	95.40 %	97.4 %
	Simple	98.80 %	86.00 %	99.60 %	94.8 %
SVM	Linear	99.20 %	95.00 %	96.80 %	97.0 %
	Quad	99.20 %	95.80 %	96.20 %	97.1 %
	Cube	98.80 %	95.40 %	95.20 %	96.5 %
	FineGauss	97.00 %	16.00 %	34.20 %	49.1 %
	MedGauss	97.80 %	94.40 %	96.60 %	96.3 %
	CoarseGauss	98.60 %	89.20 %	99.20 %	95.7 %
KNN	Fine	99.60 %	96.00 %	93.40 %	96.3 %
	Medium	98.80 %	97.00 %	91.40 %	95.7 %
	Coarse	99.00 %	91.00 %	91.80 %	93.9 %
	Cosine	99.00 %	96.80 %	92.40 %	96.1 %
	Cubic	98.80 %	97.40 %	91.60 %	96.0 %

Table 4.10: The final accuracy obtained with data set 01.

decision tree, which is also easy to implement. As the Complex Tree and Medium Tree gives nearly the same result, the best is to select the Medium Decision tree.

The original fast scan classifier may be updated to combine the best of both approaches. The anomaly classification will be done by one of presented machine learning classifiers. When the anomaly type is set, the parameters shall be estimated using the methodology of the fast scan classifier.

The classification Learner application in Matlab was tested also using the data set 04 to see if the classifiers are able to distinguish the real model from the false models. The reliability of classification went down as it is demonstrated in the Table 4.11.

The results for the data set 04 are worse if compared with the accuracy of the Fast scan classifier. Partly it is caused by a small data set (only 100 of examples for each anomaly type), but mostly it shows that the posibilites of the Matlab Classification Learner application are limited. The results presented for the data set 04 open a hypothesis that in the case of real application, the Fast scan classifier can work with higher accuracy and with the posibility to estimate the anomaly parameters.

For the future research again it should be tested, how the classifier would work with the different output created only by the black and white thresholded images as it is described in the Section 3.3. The learning process will demand a lot of RAM and other computer resources, but final model can be very accurate and the implementation can run fast.

The output of the Matlab Classification Learner consist only of confusion matri-

Type	Subtype	HC	S	VC	0	Average
Tree	Complex	96 %	40 %	63 %	95.25 %	82.9 %
	Medium	95 %	36 %	62 %	92.5 %	80.4 %
	Simple	95 %	0 %	72 %	96 %	78.7 %
SVM	Linear	98 %	32 %	69 %	95.75 %	83.1 %
	Quad	97 %	48 %	81 %	96.5 %	87.4 %
	Cube	97 %	50 %	70 %	96 %	85.9 %
	FineGauss	74 %	15 %	21 %	99.5 %	72.6 %
	MedGauss	99 %	28 %	78 %	98 %	85.3 %
	CoarseGauss	98 %	0 %	67 %	98.25 %	79.7 %
KNN	Fine	89 %	47 %	69 %	94.5 %	83.3 %
	Medium	88 %	16 %	58 %	95.5 %	77.7 %
	Coarse	87 %	0 %	34 %	99 %	73.9 %
	Cosine	90 %	16 %	63 %	95.25 %	78.6 %
	Cubic	88 %	20 %	60 %	93.5 %	77.4 %

Table 4.11: The final accuracy obtained with data set 04.

ces and the parameters of the finally obtained classifiers parameters. Therefore no detailed study which types of anomalies were classified incorrectly regarding to their location or other parameters such as depth or mass. It would require to implement all the trained classifiers into the original Fast scan algorithm to get similar tables with results.

4.3 Fast scan with ANN

		Detected Type		
		HC	**S**	**VC**
Input Type	**HC**	310 (97 %)	0	10 (3%)
	S	0	360 (100.0 %)	0
	VC	0 (0.3 %)	10 (3 %)	310 (97 %)

Table 4.12: Adaptive neural network, Confusion Matrix, Test set 01.

The convolution neural network (CNN) described in the Section 3.4.1 was tested with the extended data set 01 as it is described in the Section 3.4.1. The neural network classifies the data into three groups: *HC*, *S* and *VC*. The obtained accuracy of the clasiffication is the best of all the tested classification methods – see the confusion matrix in the Table 4.12.

The ANN implementation was finished in the last phase of the presented research and only limited results are available as the CNN implementation was complex and required more time than it was intially planned. All the tests of the implemented CNN shows that the network gives the best classification results. The network will be tested also with other types of input data (noise corputed data and false data). In fact, the complexity of the CNN implementation is far from the original research idea to find a fast algorithm capable to be implemented on limited hardware.

5 The Summary

The inspiration for the presented research was the idea to speed up the disaster recovery operation with the geophisical survey. The research was to verify if computer vision and machine learning techniques can be applied in the field of semi automated processing of the geophysical data.

After an initial research of the current state of the art in the related research the Fast scan algorithm was proposed, implemented and tested using synthetical gravity data. The reference tests with the supervized machine learning and a set of classifiers were realized. To test the suitability of the application of the neural network, in cooperation with the research partner the convolutional neural network was implemented, trained and tested.

The research was mainly focused to detect a footprint of the anomaly in the synthetical gravity data. The research was later included into the Modern2020 project and the algorithm is now being rearranged to detect the abnormal situations in the seismical data.

The proposed fast scan algorithm is capable of anomaly classification and it can also estimate the anomaly parameters. It was deeply analyzed which type of anomaly is classified incorrectly and when and why the model parameters are not estimated with high precission. The best classification results were obtained with the convolutional neural network. The standard classifiers can also be trained to classify the anomaly type with acceptable precission. The second phase includes the other potential field methods.

It was tested that the reliability of the classification depends on the input data set. The proposed Fast scan classifier was tested if it can distinguish the searched anomaly model from the similar field with no real physical meaning.

The output of the research is the source code of the Fast scan classifier, the source code of the development GUI and all the input data.

In the conclusion it can be said that the main task of the thesis is filled. All the tests and developed software demonstrate that the geophysical data can be interpreted using the computer vision and machine learning. The future development of the Fast scan algorithm should focus in the first phase to the different anomaly types (the sloped cylinder, the cube) and to finalizing the preprocessing part responsible for the detection of multiple anomaly body.

The current version of the algorithm cannot be yet applied to the real application as it is fixed to the simple geometry bodies. For the real application the real accepted anomalies should be modelled and the classifier should be trained to the real application data as it is now done in the Modern2020 project.

6 List of related publications

The problems described and solved within presented thesis were also published. Following list gives a summary of related published work.

- Kosková Třísková L., Novák J. Bárta J: Rychlá detekce geofyzikálních anomálií pro potřeby havarijních situací, 2011, Sanační technologie XIV, poster and paper
- Kosková Třísková L.: Near surface geophysical anomaly modeling and detection in Matlab, 2012, Technical computing Bratislava, presentation and paper
- Kosková Třísková L., Novák J.: Geophysical decision support system for emergency rescue, 2012, IT for Geosciences, Dubna, presentation and paper
- Kosková Třísková L., Novák J.: Application of edge and line detection to detect the near surface anomalies in potential data, 2013, International conference on Pattern recognition Applications and Methods 2013, Barcelona, poster and paper
- Kosková Třísková L.: Machine learning and structure detection used to support the nuclear waste repository monitoring, 2017, EGRSE - International Journal of Exploration Geophysics, Remote Sensing and Environment Vol. 2

Bibliography

[1] Ayala-Cabrera D., Herrera M., Izquierdo J., Perez-Garcia R: Location of buried plastic pipes using multi-agent support based on GPR images, Journal of Applied Geophysics, 1998, Volume 10, Issue 24, Pages 933-951

[2] Aydogan D.: CNNEDGEPOT: CNN based edge detection of 2D near surface potential field data, Computers & Geosciences 2012, Vol. 46, Pages 1-8.

[3] Aydogan D.: Processing the Bouguer anomaly map of Biga and the surrounding area by the cellular neural network : application to the southwestern Marmara region, Earth Planets Space, 2007, Volume 59, Pages 201–208

[4] Beale R., Jackson T.: Neural Computing, 1st Edition, Institute of Physics Publishing, 1990, ISBN: 978-0852742624

[5] Blakely Richard J.: Potential theory in gravity and magnetic applications, Cambridge University Press 1996, ISBN 0521575478.

[6] Čmelík, M., Machonský, L., Šíma, Z.: Fyzikální tabulky. TU Liberec, 2001

[7] Davies E. R.: Machine Vision: Theory, Algorithms, Practicalities, 4th Edition, Academic Press, 2012, ISBN: 978-0123869081

[8] Deza M., Deza E: Encyclopedia of Distances, Springer, 2014, ISBN 9783662443415

[9] Dondurur D., Karsli H.: Swell Noise Suppression by Wiener Prediction Filter, Journal of Applied Geophysics, 2012, Volume 90, Pages 91-100

[10] A Complete Tutorial on Tree Based Modeling from Scratch (in R & Python), available at A Complete Tutorial on Decission Tree.

[11] Fitton N., Cox S.: Optimising the application of the Hough transform for automatic feature extraction from geoscientific images, Computers & Geosciences, 1998, Volume 24, Issue 10, Pages 933-951.

[12] Guo L., Meng X., Chen Z., Li S., Zheng Y.: Preferential filtering for gravity anomaly separation, Computers and Geosciences, 2013, Volume 51, Pages 247-254

[13] Gurbuz A, McClellan J, Scott Jr. W: Compressive sensing of underground structures using GPR, Digital signal processing, 2012, Volume 22, Pages 66-73

[14] Gurbuz A, McClellan J, Scott Jr. W: Detection of linear and planar structures in 3D subsurface images by iterative dimension reduction, Digital Signal Processing, 2012, Volume 22, Issue 1. Pages 66-73

[15] Kaftan, I., Salk, M., & Senol, Y.: Evaluation of gravity data by using artificial neural networks case study: Seferihisar geothermal area (Western Turkey), 2011, Journal of Applied Geophysics, 75(4), 711–718.

[16] Konate A. et all: Capability of self-organizing map neural network in geophysical log data classification: Case study from the CCSD-MH, Journal of Applied Geophysics, 2015, Volume 118, Pages 37-46

[17] Kotsiantis S. B.: Supervised Machine Learning: A Review of Classification Techniques, Informatica, 2007, Volume 31, 249-268.

[18] Krizhevsky A., Sutskever I., Hinton G. E.: ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25, 2012, Pages 1097-1105

[19] Kuroda, M. C., Vidal, A. C., de Carvalho, A. M. A.: Interpretation of seismic multiattributes using a neural network, 2012, Journal of Applied Geophysics, 85, 15–24

[20] Lary D., Alavi J., Amir H., Gandomi W., Annette L. Machine learning in geosciences and remote sensing, Geoscience Frontiers, 2001, Volume 7, Issue 1, Pages 3–10

[21] Maas C., Schmalzl J.: Using pattern recognition to automatically localize reflection hyperbolas in data from ground penetrating radar, Computers & Geosciences, 2013, Volume 58, Pages 116-125

[22] Manger, E. G.: Porosity and Bulk Density of Sedimentary Rocks, Geological Survey Bulletin, 1963, Volume 1144-E

[23] Manukyan E.: Seismic Monitoring and Elastic Full Waveform Inversion Investigations Applied To the Radioactive Waste Disposal, The PhD Thesis submitted to ETH Zurych, 2011

[24] Marelli S., Manukyan E., Maurer H., Greenhalgh S., Green A. G.: Appraisal of waveform repeatability for crosshole and hole-to-tunnel seismic monitoring of radioactive waste repositories, Geophysics, 2010, Volume 75, Issue 5, Pages Q21-Q34

[25] Mareš S.: Úvod do užité geofyziky. 2nd edition, SNTL, 1990. ISBN 8003004276.

[26] MathWorks: Classifier types, A software manual, available at Matlab documentation web page.

[27] Młynarczuk M., Górszczyk A., Ślipek B.: The application of pattern recognition in the automatic classification of microscopic rock images, Computers & Geosciences, 2013, Volume 60, Pages 126-133

[28] Geosoft: Oasis Montaj gridding, A software manual, available at Geosoft web page.

[29] Nuber, A., Manukyan, E. and Maurer, H.: Enhancement of near-surface elastic full waveform inversion results in regions of low sensitivities, Journal of Applied Geophysics, 2015, Volume 122, Pages 192–201.

[30] Al-Nuaimy W. et all: Automatic detection of buried utilities and solid objects with GPR using neural networks and pattern recognition, Journal of Applied Geophysics, 2000, Volume 43, Issue 2-3, Pages 157-165

[31] Patel A. K., Chatterjee S.: Computer vision-based limestone rock-type classification using probabilistic neural network, Geoscience Frontiers, 2016, Volume 7, Pages 53-60

[32] Reynolds J. M.: An Introduction to Applied and Environmental Geophysics, 2nd edition, John Willey and sons 2011, ISBN 978-0-471-48535-3

[33] Salem A.: Multi-deconvolution analysis of potential field data, Journal of Applied Geophysics, 2011, Vol. 74, Issue 2-3, Pages 151-156.

[34] Salem A., Gamey T. J., Ravat D., Ushijima K.: Automatic Detection of UXO from Airborne Magnetic Data Using a Neural Network, Subsurface Sensing Technologies and Applications, 2001, Vol. 2, Pages 191-213

[35] Sharmaa A., Liua X., Yangb X., Shic D: A patch-based convolutional neural network for remote sensing image classification, Neural Networks 2017

[36] Shih Frank Y.: Image processing and pattern recognition, Fundamentals and Techniques, John Willey and sons, 2010, ISBN 978-0-470-40461-4

[37] S class CS231n: Convolutional Neural Networks for Visual Recognition: A course notes, available online at http://cs231n.github.io/.

[38] Zyada Z., Matsuno T., Hasegawa Y., Sato S., Fukuda T: Advances in GPR-based landmine automatic detection, Journal of the Franklin Institute, 2011, Volume 348, Issue 1, Pages 66-78

7 Attachments

The attached DVD with source codes contains the electronic version of this document and all the sources used to get the published results. The content of the CD is organized in following folders:

- The folder **gravity_input_data** contains scripts used to get the gravity anomaly models. All the scripts were written using Python, the folder structure is described in the *readme* file. The main file is called *get_data.py*, the gravity anomaly defining functions are defined in the file *gravity.py*. The data sets used to test the algorithm are attached as well in this folder. The main *get_data.py* file has a lot of parameters for the target data set, the parameters are described in the *readme*. The folder also contains all the datasets described in the Chapter 4.

- The folder **gravity_classifier** contains scripts used to define the fast scan classifier, the folder structure is described in the *readme* file. The classifier was implemented in the Matlab 2017 R2 environment. The folder structure is described in the *readme* file located in the root folder.

- The folder **development_gui** contain the source code of the fast scan classifier development GUI. The folder structure is described in the *readme* file located in the folder.