



Assignment of master's thesis

Title:	Detection of archeological sites from aerial images
Student:	Bc. Anna Moudrá
Supervisor:	prof. Ing. Michal Haindl, DrSc.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2021/2022

Instructions

The work aims to design and implement an algorithm for automatic recognition of immovable archaeological monuments hidden under the soil surface from color aerial photographs and lidar data based on vegetation cover changes. The underground remains of buildings, such as the Celtic opida, medieval fortresses, or modern era fortifications, locally change the subsoil layer's chemical composition and structure and affect the vegetation growth in their upper layer. Automatic detection of such archaeological sites will significantly speed up the difficult and time-consuming visual detection of aerial photographs. In multichannel multimodal data sources, such detection is impossible, and their automatic classification is the only option. The work's successful results will significantly impact the cataloging of hitherto unknown archaeological sites in the Czech Republic and will enable relevant publications in the professional press.



**FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE**

Master's thesis

Detection of Archaeological Sites from Aerial Images

Bc. Anna Moudrá

Department of Applied Mathematics
Supervisor: prof. Ing. Michal Haindl, DrSc.

May 6, 2021

Acknowledgements

I would like to thank my supervisor prof. Ing. Michal Haindl, DrSc. for his advice and support during the process of implementing and writing this thesis. I would also like to thank my family and friends for their endless kindness and support throughout the years of my studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No.121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 6, 2021

.....

Czech Technical University in Prague
Faculty of Information Technology
© 2021 Anna Moudrá. All rights reserved.

This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).

Citation of this thesis

Moudrá, Anna. *Detection of Archaeological Sites from Aerial Images*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2021.

Abstrakt

Podpovrchová naleziště antropogenního původu jsou často viditelná z leteckých snímků ve formě změn vegetačního pokryvu. Tyto příznaky vznikají rozdílnými podmínkami pro růst vegetace v souvislosti se změnou lokální chemické skladby půdy, kterou podpovrchové objekty způsobují. Tato práce se věnuje návrhu a implementaci metody, která je schopna tyto vegetační příznaky detekovat ve volně dostupných leteckých snímcích, poskytovaných mapovými portály ve formě ortomap. Potenciál takovéto automatické detekce tkví ve výrazném urychlení procesu objevování a katalogizace archeologických nalezišť.

Práce obsahuje stručný úvod do problematiky letecké archeologie a rešerši state-of-the-art metod dálkového snímání a automatické detekce se zaměřením na postupy, u kterých je množství pozitivních příkladů podobně omezené jako v námi vytvořeném datasetu. Dále jsou popsány tři přístupy k automatické detekci, je vytvořen jejich návrh a následná implementace.

Na vytvořeném datasetu je provedeno experimentální testování navržených metod. Pomocí analýzy výsledků jsme odvodili omezení jednotlivých přístupů. První i druhá metoda jsou založeny na detekci rohů a lineárních segmentů a jejich vzájemných prostorových vztahů. Třetí metoda, založena na template matchingu vegetačních příznaků přibližně známých tvarů s využitím zobecněné Houghovy transformace, se ukázala jako nejperspektivnější pro budoucí využití.

Klíčová slova letecká archeologie, detekce vegetačních příznaků, zobecněná Houghova transformace, zpracování obrazu

Abstract

Subsurface deposits of anthropogenic origin are often visible from aerial photographs in the form of changes in vegetation cover. These crop marks arise from different conditions for vegetation growth due to the changes in the local chemical composition of the soil caused by subsurface objects. This thesis's objective is to design and implement a method that is able to detect these crop marks in publicly available aerial images, provided by mapping services in the form of orthomaps. Such a method could significantly speed up the process of cataloging archaeological sites.

The work offers a brief introduction to the concept of aerial archaeology and an overview of state-of-the-art methods of remote sensing and automatic detection with focus mainly on procedures where the number of positive examples is similarly limited as our dataset, constructed in this work. Three approaches to automatic detection are proposed, designed and implemented.

The proposed methods are experimentally tested on the assembled dataset. The results are analyzed and the limitations of each approach are deduced. Both the first and second approach is based on detection of corners and linear segments and their spatial relations. The third approach, based on the template matching of crop marks of approximately known shapes, utilizing the generalized Hough transform algorithm yields the best results in crop mark recognition and is most promising for future research.

Keywords aerial archaeology, crop mark detection, generalized Hough transform, image processing

Contents

Introduction	1
1 State of the Art	3
1.1 Recently used techniques for detection of archaeological remains from airborne data	3
1.2 Automatic site detection	4
1.3 Similar problems with applicable solutions	5
2 Data Acquisition	7
2.1 Area of study	7
2.2 Sources	8
2.2.1 High-resolution airborne imagery	8
2.2.2 Airborne laser scanning	10
2.2.3 Complementary data sources	11
2.3 Data acquisition	11
2.3.1 High-resolution aerial images	11
2.3.2 ALS data	12
2.3.3 Complementary data	13
2.4 Dataset description	14
2.4.1 Dimensions and dates	14
2.4.2 Irrelevant structures and negative examples	15
3 Methods	17
3.1 Image pre-processing	17
3.1.1 Noise removal	17
3.1.2 Thresholding	19
3.1.3 Edge detection	20
3.1.4 Principal component analysis	21
3.1.5 Morphological operators	22
3.2 Local features	23

3.2.1	Harris corner detector	23
3.3	Hough transform	25
3.3.1	Hough transform for lines and circles	26
3.3.2	Generalized Hough transform	27
3.4	Rectangular enclosure detection	28
3.4.1	Extraction of ridge features and candidate points	29
3.4.2	Graph representation and rectangularity measure	30
3.5	Classification and evaluation	31
3.5.1	Support Vector Machine	32
3.5.2	Random Forest	34
3.5.3	Cross-validation	34
3.5.4	Evaluation	35
4	Design	37
4.1	Corner based detection	38
4.1.1	Image pre-processing and corner detection	38
4.1.2	Detection with 4 corners	39
4.1.3	Detection with 3 corners	39
4.2	Linear segment based detection	41
4.2.1	Image pre-processing	41
4.2.2	Detection	43
4.2.2.1	Candidate point extraction	43
4.2.2.2	Linear segment detection	43
4.2.2.3	Configuration ranking	44
4.2.3	Classification	45
4.3	Template based detection	45
4.3.1	Image pre-processing	46
4.3.2	Templates	49
4.3.3	Detection	49
4.3.4	Classification	50
5	Implementation	53
5.1	Development tools and libraries	53
5.2	Image pre-processing	54
5.3	Corner based detection	54
5.4	Linear segment based detection	54
5.5	Template based detection	54
5.6	Other enclosed scripts	55
6	Experiments and results	57
6.1	Corner-based detection methods	57
6.1.1	Image pre-processing and corner detection	58
6.1.2	Detection with 4 corners	59
6.1.3	Detection with 3 corners	60

6.2	Linear segment based detection	60
6.2.1	Image pre-processing and valid configuration detection .	61
6.2.2	Classification	64
6.3	Template-based detection	65
6.3.1	Image pre-processing	65
6.3.2	Templates	67
6.3.3	Singlepass vs. region of interest detection	68
6.3.4	Hough accumulator threshold value	69
6.3.5	Classification	69
6.4	Discussion	71
6.4.1	Summary	71
6.4.2	Future improvements	72
6.4.3	Contribution of this work	73
	Conclusion	75
	Bibliography	77
	A Basic Notation and Conventions	83
	B Acronyms	85
	C Contents of enclosed flash disk	87
	D Algorithms	89
	E Sites	93
	F Experiments	99

List of Figures

1.1	Growth stages of small-grain cereals. Image sourced from [1].	4
2.1	Illustration of the scanning process. Image sourced from [2].	8
2.2	Orthorectification diagram. Image sourced from [3].	9
2.3	Comparison of RGB (left) and CIR (right) photography.	10
2.4	Comparison of ALS data and crop mark visibility, Ctinèves site. Data processed with QGIS.	13
2.5	Larger structures are clearly visible. Crop mark (N 50° 24' 12", E 14° 13' 41").	14
3.1	Edge-preserving bilateral filter, images sourced from [4].	18
3.2	Line detection via the Hough transform.	26
3.3	Generalized Hough transform: construction of the R table. Image obtained from [5].	27
3.4	Rectangular enclosures: Sampled candidate points, colors inverted.	30
3.5	Convexity constraint illustration. Image from [6].	30
3.6	SVM binary classification: (a) Example of optimal linear discrimi- nation, thick line denotes the discriminating hyper-plane, (b) case of linearly non-separable classes in 1D, (c) application of the kernel trick to create a linearly separable problem. Images sourced from [5].	33
3.7	Decision tree structure. Image from [5].	34
4.1	General data flow of the crop mark detection algorithm.	37
4.2	General data flow of the corner based detection algorithm.	38
4.3	HC: Image pre-processing applied to the red channel.	39
4.4	Detection with 4 corners.	40
4.5	Detection with 3 corners.	40
4.6	General data flow of the RE algorithm.	41
4.7	RE: (a) PC1, enhanced contrast, (b) white top hat, (c) white noise subtracted from (a).	42

4.8	RE: (a) black top hat and threshold, (b) morphological closing (c) histogram equalization.	42
4.9	RE: (a) detected and linked segments, (b) valid configurations. . .	44
4.10	Morphology of buried features in non-equivalent scale. Image sourced from [7].	45
4.11	General data flow of the GHT algorithm.	46
4.12	GHT: Image pre-processing.	46
4.13	GHT: (a) grayscale image, PC1, (b) enhanced contrast.	47
4.14	GHT: (a) white top hat, (b) white detail subtracted, (c) median filter.	47
4.15	GHT: (a) adaptive threshold, (b) median filter, colors inverted, (c) erosion	48
4.16	GHT: (a) LU map filtration, (b) CIR image filtration, (c) blob filtration.	49
4.17	Template examples. Dimensions ($w \times h$ or r , thickness) in pixels, colors inverted. (a) Rectangular template ($130 \times 78, 7$), (b) Circular template ($49, 5$).	50
6.1	Red channel Harris corner detection: (a) Černouček (b) Ledčice (d) Březno ^a	57
6.2	Harris corner detector: (a) results for LHC, (b) results for OHC. .	58
6.3	LHC: Cases of successful crop mark localization.	59
6.4	LHC: Most attempts ended unsuccessfully.	60
6.5	OHC: Cases of successful crop mark localization.	61
6.6	OHC: Corner orientation can be wrongly determined due to fragmentation in the original colour image (lower-left corner, orientation indicated by blue line).	62
6.7	OHC: Many sites were missed entirely due to undetected corners or wrongly determined corner orientation. In image (a) 3 corners are successfully detected, left upper corner remains undetected, while right upper corner's configuration is not suitable for the OHC algorithm. In image (b) upper left corner is missed by the corner detector and lower right corner coincides with the surroundings and is filtered out during the pre-processing, remaining corners and their orientations are detected correctly.	62
6.8	RE: site Ctiněves ^a PC1 (a) and RED (b) after segment enhancement. PC1 (d) and RED (e) after valid configuration detection. PC1 (c) after segment enhancement and (f) valid configuration detection with new parameters. Size of rectangle around each candidate point indicates size of rectangularity measure f_R	64
6.9	RE: Examples of ranked configurations, PC1 input.	64
6.10	Experiments with image pre-processing, colors inverted.	66
6.11	Red channel (a,d), PC1 (b,e) and VARI (c,f) before and after pre-processing.	67

6.12	(a) correctly filtered crop mark region, Ctiněves ^a (b) fragmented crop mark is divided into multiple regions, Březno ^a	69
E.1	Positive examples: part 1/3	94
E.2	Positive examples: part 2/3	95
E.3	Positive examples: part 3/3	96
E.4	Negative examples.	97
E.5	Unused sites.	98
F.1	Color extended Harris corner detector: (a,d) RGB, (b,e) ratio color space (corners projected onto RGB image), (c,f) opponent color space. Threshold for corner selection is set to 0.	99
F.2	RE classification results: PC1 input with class_weight parameter set to (a) .99 (b) .98 (c) .95, Part 1/3	100
F.3	RE classification results: PC1 input with class_weight parameter set to (a) .99 (b) .98 (c) .95, Part 2/3	101
F.4	RE classification results: PC1 input with class_weight parameter set to (a) .99 (b) .98 (c) .95, Part 3/3	102
F.5	RE classification results: RED input with class_weight parameter set to (a) .99 (b) .98 (c) .90, Part 1/3	103
F.6	RE classification results: RED input with class_weight parameter set to (a) .99 (b) .98 (c) .90, Part 2/3	104
F.7	RE classification results: RED input with class_weight parameter set to (a) .99 (b) .98 (c) .90, Part 3/3	105
F.8	GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	106
F.9	GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	107
F.10	GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	108
F.11	GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	109
F.12	GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	110
F.13	GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	111
F.14	GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	112
F.15	GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	113
F.16	GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	114
F.17	GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.	115

Introduction

Crop marks, soil marks, and unusual terrain break lines have been widely known to archaeologists as proxies for the identification of buried archaeological remains [8].

To identify these features from air and space, different methods of remote sensing have been explored ever since the end of World War II. It was then when technological advances in two fields - air travel and photography - made accurate documentation of the Earth's surface possible for the first time. Initially, the remotely sensed data were made for different purposes, such as strategic military and agricultural land planning, and were later re-purposed as valuable sources for archaeological research [9].

Nowadays, the rapid technological advances in remote sensing, such as space-borne and airborne high-resolution multispectral images and ground-penetrating LiDAR sensors, are candidly utilized by researchers for archaeological purposes with great success [10, 11, 12]. However, because of the high cost of specialized sensors, equipment, and flight-time, along with the time needed for large area mapping, the focus has recently shifted again to publicly available data [13, 14]. In recent years, commercial mapping services produce data in an increasingly higher resolution, making the discovery of even small archaeological sites and features possible. Given the lower cost of obtaining and processing publicly available data, this practice of co-opting open data for archaeological research will likely continue.

Motivation

Modern approaches to airborne archaeology rely on expensive surveying techniques in preordained areas of archaeological interest. In recent years, many mapping services started to provide satellite and aerial images in a resolution high enough to be useful for archaeological purposes. While this freely available data has many limitations, exploiting its full potential may lead to interesting results for archaeologists who currently must rely on either man-

ual processing of the available data or on their own costly surveillance. As mapping services periodically update their data, it is becoming an important source for the detection of sites that have been recently destroyed or covered by new structures.

Objectives

The overall objective of this thesis is to survey the publicly available data sources and subsequently design and implement a method for automatic recognition of available examples of crop marks, indicating subsurface remains of archaeological importance. Based on the research outlined in this work, we would like to generalize the properties of available examples of human-made crop marks and design and implement a method capable of automatic recognition of these positive examples, potentially expanding the method for detection of yet undiscovered sites. Were this work to succeed, it has the potential to significantly speed up the process of cataloging archaeological sites in the Czech Republic as current methods rely on manual discovery and classification.

Contribution

The research part of this thesis offers the reader an introduction to currently used aerial archaeology methods. We introduce publicly available data sources that are suitable for inspection and could be potentially valuable base or complementary inputs for the task of automatic site recognition. We analyze methods and algorithms applicable to our task and the available data, and we implement three such methods. The methods are tested on few positive examples and comparatively evaluated.

Structure

This thesis is structured as follows. In chapter 1, key concepts are introduced and the state of the art in the field of airborne archaeology is briefly summarized. Chapter 2 provides an insight into what data sources suitable for the task of automatic detection of archaeological features in the Czech Republic are publicly available. The examples of positive cases available to us are described. Chapter 3 offers a review of methods of image processing, feature extraction, and classification considered for our task. In chapter 4, the overview of each method is proposed and the tools, as well as the implemented scripts, are described in chapter 5. In chapter 6, we describe the conducted experiments and evaluate and discuss the results produced by our implemented methods.

State of the Art

This chapter is focused on up-to-date methods of systematic discovery of sub-surface archaeological remains.

1.1 Recently used techniques for detection of archaeological remains from airborne data

Research made in recent years by Lesaponara et al. [11], Agapiou et al. [8, 15] and Calleja et al. [12] and others [10] is predominantly focused on the exploitation of high-resolution multispectral and multitemporal satellite data. Apart from utilizing vegetation health indexes created for agricultural purposes, they research new ways to combine false-color composites from available multispectral satellite data to better identify crop and soil marks [8, 16, 10]. As concluded from the above-mentioned literature, the detection process consists of the following general steps:

1. Determination of the area of study, the scope is usually narrowed to a smaller area of known historical significance.
2. Exploitation of literature and collection of publicly available and historical surveillance data to gain contextual information and to construct a multitemporal dataset.
3. Re-screening of the area in key regions of interest.
4. Image enhancements, manual processing of the final dataset and discovery of relics.
5. Analysis of results, ground-truthing.

Because crop marks, when visible, are usually characterized by a very small spectral separability, the research is also focused on the effects different phenological stages of vegetation growth have on the quality of archaeological

research. Researchers note it is perhaps the most important factor to consider when collecting data. The research made by Agapiou et al. [8] suggests that the boot stage of the vegetation growth cycle, illustrated in Figure 1.1 as the 4th stage from the right, is the most ideal time period for crop mark detection.

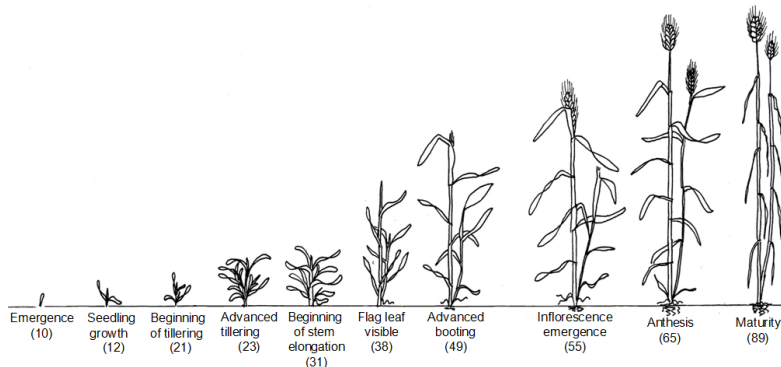


Figure 1.1: Growth stages of small-grain cereals. Image sourced from [1].

1.2 Automatic site detection

Research with the objective to assess the possibility of automatically extracting and mapping subtle archaeological features was conducted by Lasaponara et al. [11]. The study was conducted in Hierapolis, Turkey and consisted of two main steps: the unsupervised segmentation of high resolution multispectral satellite images to extract non-homogenous regions, and the subsequent classification of linear and rectangular features of archaeological interest. The outcomes of their analysis were evaluated by a field survey and resulted in the discovery of buried remains spanning from Imperial Roman to early Byzantine times.

In 2014, Lei Luo et al. [14] proposed an automatic method for the extraction of qanat¹ tops from high-resolution satellite images available in the Google Earth application. They used mathematical morphology for image processing and circular Hough transform. While their research was limited to only two sites, they reported a successful detection rates with their approach.

Kazimi et al. [17] recently proposed a hierarchical Convolutional Neural Network (CNN) to classify archaeological objects in airborne laser scanning data. The work focused on the detection of larger geological structures, such as lakes, tracks and streams, caused by historical mining activity in the region. In the future the researchers hope to extend the detection outcome to geometrically reconstruct the mining site in the region.

In 2019, Laundauer, Hesse [18] used a pre-trained CNN model and modified it via transfer learning to detect charcoal kilns in airborne laser scanning data

¹Type of underground aqueduct.

with 80% detection rate. However, their dataset was for publicly available data unusually detailed, with spatial resolution of 25 cm per pixel. Furthermore, the researchers had more than 29,000 of labeled, positive examples at their disposal.

In 2019, Stott et al. [19] used publicly available airborne laser scanning data to detect remnants of Viking ring fortresses across Denmark. Instead of deep learning, they used their domain knowledge and filtered circular structures from the data via the circular Hough transform. To filter out large numbers of false positives, a Random Forest model was trained with contextual information data such as historical maps and names of local landmarks dating to the 18th century. Eventually, they were able to identify several known archaeological sites, as well as multiple sites of geological interest such as glacial or meteorite craters.

1.3 Similar problems with applicable solutions

Aerial and satellite images are used for object detection across many applications. Most applications do not suffer from a lack of positive examples and as such are able to utilize deep learning. For computer vision tasks where large number of data is not available, the Canny edge detector, Hough transform and Harris corner detector are widely used methods for the detection of regular shapes, mostly large structures such as roads and buildings [20, 21, 5]. That being said, we came across to mostly older research as on-going research seems to be focused on the adaptability of small datasets for deep learning.

Particular is the work of Zingman et al. [6]. In 2016, they developed an approach for detecting the rectangular shapes of live-stock enclosure ruins in high-resolution remotely sensed images. While their objective is to search for structures and not crop marks, there are many similarities in the presentation of both objects. The ruins are usually composed of linear walls, of rectangular shape and oftentimes incomplete or with only fragmented sides. Additionally, the surrounding environment of the enclosure ruins is very similar to the complex terrain surrounding crop marks, containing irrelevant structures such as trees, trails and rocks. The authors also deal with the limiting issue of having only few positive examples. While this approach would be only applicable to detection of crop marks with strictly rectangular shapes, the authors are optimistic that their approach could have comparably good results when applied to other objects in rural areas.

Data Acquisition

The area of study is introduced in section 2.1. In section 2.2 we provide an explanation of key concepts of aerial surveying and data pre-processing. The list of publicly available data sources and the means of acquisition of data used in this work are described in section 2.3. Finally, the resulting dataset is defined in section 2.4.

2.1 Area of study

Our process of data acquisition began with several examples of known sites being provided to us by prof. PhDr. Martin Gojda, CSc., DSc. The initial set contained 5 examples of sites, additional 8 examples were provided later, during the process of implementation. The crop marks are clearly visible on several online mapping services such as Mapy.cz and Google Maps:

- Ledčice, district Mělník (N 50° 20' 0.02", E 14° 16' 42.21")
- Ctiněves, district Litoměřice (N 50° 22' 33.29", E 14° 18' 30.77")
- Černouček, district Litoměřice (N 50° 21' 24.59", E 14° 18' 2.88")
- Straškov^a, district Litoměřice (N 50° 21' 45.26", E 14° 15' 18.77")
- Straškov^b, district Litoměřice (N 50° 21' 55.86", E 14° 15' 26.54")
- Hořín, district Mělník (N 50° 20' 34.78", E 14° 27' 24.19")
- Rakovice, district Písek (N 49° 27' 18.57", E 14° 3' 38.51")
- Nechanice, district Hradec Králové (N 50° 14' 27.24", E 15° 37' 58.21")
- Hostinné, district Trutnov (N 50° 32' 16.21", E 15° 41' 59.73")
- Březno^a, district Mladá Boleslav (N 50° 21' 59.72", E 13° 45' 1.73")

2. DATA ACQUISITION

- Cítov, district Mělník (N 50° 22' 7.94", E 14° 24' 7.37")
- Třeboutice, district Litoměřice (N 50° 32' 6.03", E 14° 10' 55.69")
- Vražkov^a, district Litoměřice (N 50° 22' 46.08", E 14° 15' 7.65")

Two additional sites from the area were added during the process of data acquisition:

- Březno^b, district Mladá Boleslav (N 50° 22' 27.57", E 13° 44' 8.04")
- Vražkov^b, district Litoměřice (N 50° 22' 37.15", E 14° 15' 51.94")

2.2 Sources

2.2.1 High-resolution airborne imagery

Aerial photography or airborne imagery is the process of taking photographs (of a planet's surface) from an aircraft or another flying object such as unmanned aerial vehicles (UAVs) or, historically, carrier pigeons [22]. Aerial images are taken at lower altitudes and thus can, at a time, cover smaller areas than satellite images. For individual research, aerial photography is often more suitable as it is more accessible and offers higher resolution.

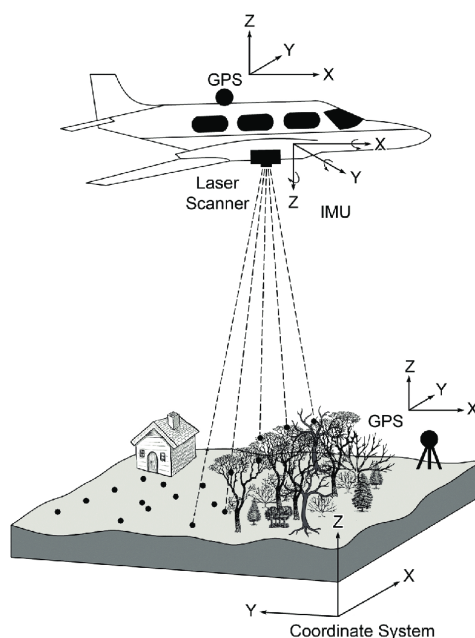


Figure 2.1: Illustration of the scanning process. Image sourced from [2].

Orthorectification

To remove the geometric and radiometric distortions in airborne images due to variances in viewing angles and distance during the survey, the images are transformed via a process called orthorectification [22]. The original images are created with a significant overlap to make the process of aligning and stitching of thousands of pictures into a seamless mosaic possible. To correct geometrical distortion caused by sudden changes in terrain height, the orthorectification process is enhanced by digital terrain model (DTM). If no such model is available, the process can be simplified to stitching of the edges of each image. This process is called false-orthorectification, as the taller structures, such as buildings or trees, retain a visible tilt, illustrated in Figure 2.2.

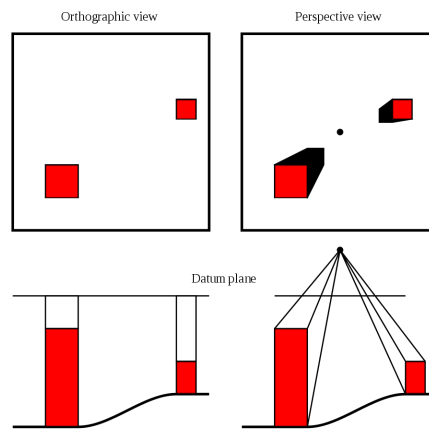


Figure 2.2: Orthorectification diagram. Image sourced from [3].

Color photography

The most common form of multispectral imagery is the visible spectrum color image, mostly in the RGB color space. The images are made with RGB sensors, sensitive to visible light, characterized by wavelengths between cca 400 and 700 nm. Each channel corresponds to one one-type sensor, resulting in the composite image with 3 channels, each respectively depicts the red, green a blue portions of the visible light spectrum [5]. The radiation with slightly longer wavelengths, between cca 750 and 1400 nm, is part of the near-infra red (NIR) waveband and is not visible to the human eye. However, because the NIR spectral band is able to penetrate deeper into the vegetation, near-infra red sensitive sensors are able to capture the state of vegetation health in much better detail than the RGB sensors. This relationship and it's significance for aerial archaeology is described in more detail in the work by G. J. Verhoeven [23].

2. DATA ACQUISITION

Since the optimum spectral region for positive crop mark detection in the NIR waveband is considered to be between 700 and 800, several approaches have been proposed to enhance the vegetation health in the form of pseudo-color composites [8]. The NIR channel is usually presented as part of the NIR-R-G false color composite picture, called the CIR image. As the NIR spectrum replaces the red channel, blue channel is missing and the other two channels are shifted, the CIR image looks characteristically red in regions of healthy vegetation, as shown in Figure 2.3.

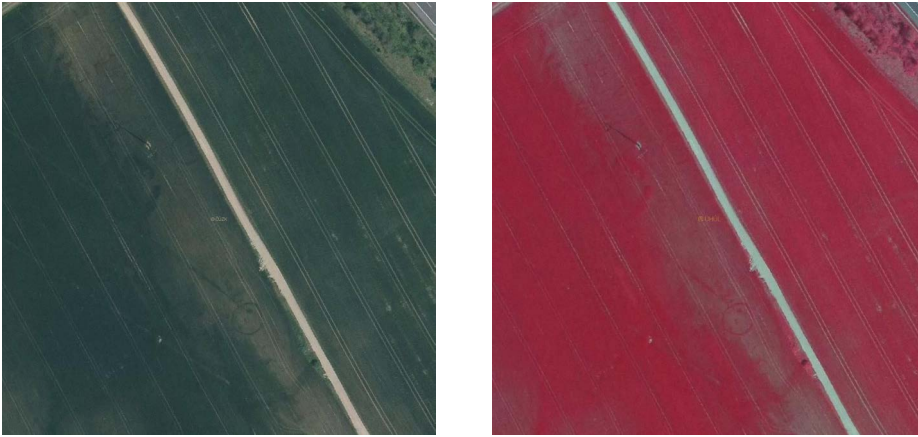


Figure 2.3: Comparison of RGB (left) and CIR (right) photography.

Widely used vegetation health indices are the Simple Ratio also called the Ratio Vegetation Index (SR or RVI) and the Normalized Difference Vegetation Index (NDVI) [8]. As ratios, they are invariant to differences in illumination conditions. For cases where near-infra red sensors are not available, the alternative Visible Atmospherically Resistant Index (VARI) has been used for plant health assessment in agriculture. The indexes, which result in grayscale false-color images, are presented in Equation 2.1.

$$\begin{aligned} RVI &= \frac{NIR}{R} \\ NDVI &= \frac{NIR - R}{NIR + R} \\ VARI &= \frac{G - R}{G + R - B} \end{aligned} \tag{2.1}$$

2.2.2 Airborne laser scanning

The technique of airborne laser scanning (ALS) is well used for digital surface model (DTM) generation. In recent years, light detection and ranging (LiDAR) technology made progress, due to its decreasing costs, across multiple

industries as laser pulses are able to penetrate vegetation cover and capture the true shape of the underlying relief. As such, it has the potential to uncover long-hidden anthropogenic structures [24]. The data of lower density, acquired by public service, can be lent to students by the Czech Office for Surveying, Mapping and Cadastre (ČÚZK) [13].

2.2.3 Complementary data sources

To offer more context, other complementary data may be sourced, such as regional maps, historical data and plans of land use. However, this approach may be in general more practical for more narrowed research, focused on just one or two sites. In this work we use the plans of land use (referred to as LU) provided by ČÚZK to filter out irrelevant regions such as urban areas and forests.

2.3 Data acquisition

In this work we considered these publicly available sources:

- Google Maps and Google Earth
- Mapy.cz
- The Czech Office for Surveying, Mapping and Cadastre

2.3.1 High-resolution aerial images

Google Maps and Google Earth services available online do not provide historical photography, nor they provide the actual date of screening, vital to determine the phenological stage of the vegetation cover. The desktop version of Google Earth Pro² provides historical airborne maps, but also does not specify the date of screening. Moreover, the imagery in non-urban areas is often in distorted form, obviously mixed from several image sources. For these reasons both applications proved to be too limiting sources, unsuitable for our task.

The images created by The Czech Office for Surveying, Mapping and Cadastre³ are available for viewing via their web client Geoprohlizec⁴. The images are already in orthorectified form and are updated every three, and recently every two, years. The specific date of each screening is available in the metadata for each map list of size 2×2.5 km. The spatial resolution of orthomaps made in between years 2017 and 2020 is 20 cm per pixel. Orthomaps made prior to 2017 have spatial resolution of 25 cm or 50 cm per pixel.

²<https://www.google.com/intl/cs/earth/>

³<https://www.cuzk.cz/>

⁴<https://ags.cuzk.cz/geoprohlizec/>

As the images provided by ČÚZK are mostly taken at times when crop marks are not well defined, the orthomaps available via Mapy.cz proved to be a more valuable source. The orthomaps publicly hosted at Mapy.cz were created by the company GEODIS Brno since 2003 until 2015. Since then the orthomaps have been provided by Topgis, s.r.o.⁵. Unfortunately, the specific dates of screening are available only for screening made by the latter company, and so until the year 2015 all dates are specified only in years. The most recent orthomaps, viewed under the highest zoom setting, have a spatial resolution of 12.5 cm per pixel, older maps were made with spatial resolution of 20 cm or 50 cm per pixel.

The image data were acquired through web scraping and via the Web Map Service standard. To ensure the RGB and CIR photography and the land use maps align the images needed to be sourced under an unified coordinate system. While the the WGS84 [25] is the standard used around the world and in GPS devices, the Czech standard used for all geodesy purposes is the JTSK-S [26] coordinate system. Therefore to prevent distortion, all coordinates were first converted from the WGS84 standard to JTSK-S. The WMS service was then used to download data from access points provided by ČÚZK⁶. Python libraries *pyproj*⁷ and *owslib*⁸ were used for this task.

As the color images available from Mapy.cz are not freely accessible through WMS, the data used in this work were acquired via the scraping of the mapping service website. The photography available at Mapy.cz is available for use under the condition of listing the mapping service as source⁹. The watermark remains intact in the original data and is only obscured by image preprocessing. As the acquired images are already in an orthorectified form, we manage to sidestep this otherwise crucial process.

2.3.2 ALS data

We were able to acquire 20 map lists of ALS data from the Czech State Administration of Land Surveying and Cadastre. The area contains most of our crop marks and so we could compare the resulting DTM to our images sourced from mapping services. The DTM 5th generation is the latest digital model of the Czech Republic, compiled from airborne laser scannings made in years 2009 to 2013 and completed in 2016. While other DTMs were available prior to 2016, the DTM 5G has the highest resolution yet. The prior model DTM 4G is available in regular raster 5×5 m, while DTM 5G is provided as triangulated irregular network (TIN) with resolution in places of up to 10

⁵<https://www.topgis.cz/cs/letecke-snimkovani/>

⁶Geoportal.cz - WMS access points

⁷<https://pypi.org/project/pyproj/>

⁸<https://geopython.github.io/OWSLib/>

⁹https://licence.mapy.cz/?doc=mapy_pu

meters/pixel and has been pointed out as a valuable source for archaeologists in recent years [13].

The data were lent to us as the original .XYZ files, containing cloud point coordinates for each map list, 2.5×2 km. The clouds were imported into a geographic information system software¹⁰ and processed with interpolation and TIN algorithms to create the final visualisation of the hill-shaded terrain model.

Upon the inspection of the imported point clouds and rasterized output, and comparison to our color images we concluded that the resolution of the data in place of our positive examples is insufficient. Therefore, we decided not to proceed with the data in our implementation. The rasterized DTM was further inspected for other known archaeological sites of larger dimensions, which were visible in the hill-shaded model and thus more suitable for detection by means of ALS. Examples of the data cloud, rasterized DTM and comparison to crop marks in the area are present in figure Figure 2.4 and Figure 2.5.

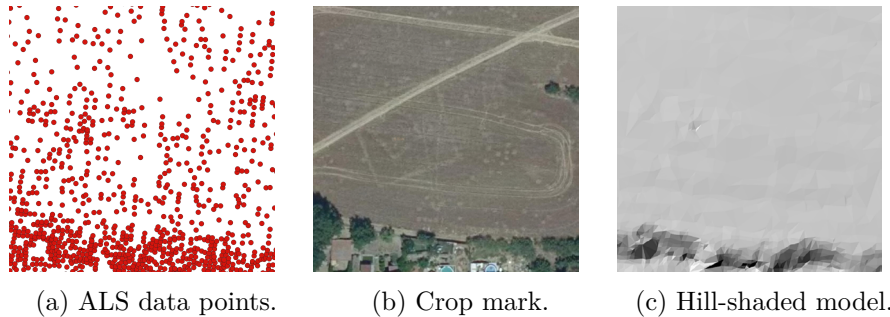


Figure 2.4: Comparison of ALS data and crop mark visibility, Ctiněves site. Data processed with QGIS.

2.3.3 Complementary data

The RGB orthomaps provided by ČÚZK were mostly created from images where crop marks were not very well visible. The images were taken mostly during late summer, when the crops were already fully grown. To utilize other resources provided by ČÚZK, for each site, the CIR orthomaps and the land use maps were also downloaded.

¹⁰QGIS, free and open-source system has been used for the purposes of this work. Available from <https://qgis.org/en/site/>.

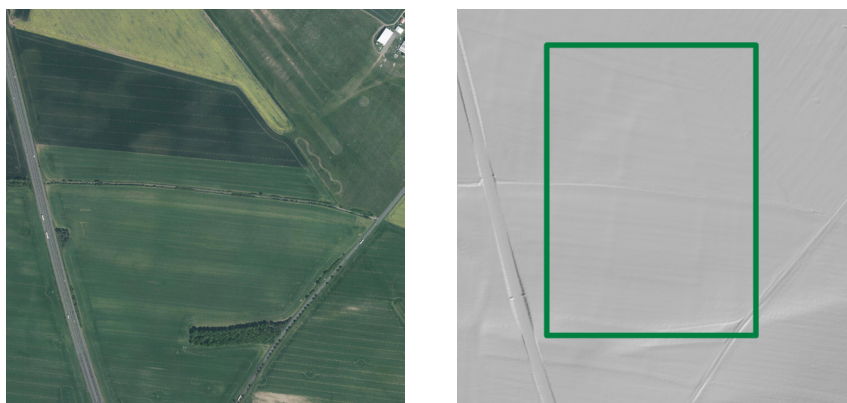


Figure 2.5: Larger structures are clearly visible. Crop mark (N 50° 24′ 12″, E 14° 13′ 41″).

2.4 Dataset description

The final set of positive examples consists of 10 sites. In this section, the data is described in terms of resolution and each site is described in terms of dimensions and the date of screening.

The full dataset has been acquired and analysed before implementation as full understanding of the data available and its properties has been crucial to the later design of each algorithm.

2.4.1 Dimensions and dates

Our task requires we use the images with the best crop mark visibility. Therefore, in addition to variable vegetation cover, most images in the final dataset were taken at different times, under different conditions and with different equipment. That resulted in inconsistent illumination, color and resolution. As higher resolution airborne photography became available only in recent years, some sites do not have the best resolution available for years when the crop marks were more visible due to more ideal phenological stage of the vegetation cover. For this reason we decided not to use the highest available zoom setting at Mapy.cz. Instead we used zoom 19, which is available at most sites since screening in 2005. The final resolution of images from Mapy.cz corresponds to circa 25 cm per pixel for sites screened in recent years by Topgis s.r.o, and 50 cm per pixel for sites screened by GEODIS Brno. The orthomap tiles available at Mapy.cz are sized 256×256 pixels. We decided to combine 16 tiles to create an image of size 1024×1024 pixels for each site. While we could have used lower or higher number of tiles, the resulting image size allows for multiple smaller marks to be present in one image (Černouček, Ctiněves) while larger crop marks remain intact (Ledčice). In addition, the resulting images are small enough so that the results of detection can be viewed com-

Location	Image ID	Resolution [m]	Mapy.cz RGB	ČÚZK CIR
Černouček	1	0.5	2004-2006	19.5.2017
Ctiněves	3	0.25	22.5.2016	18.5.2017
Ctiněves	6	0.5	2004-2006	18.5.2017
Straškov ^a	4	0.25	22.5.2016	19.5.2017
Ledčice	7	0.25	6.6.2015	19.5.2017
Březno ^a	8	0.5	2004-2006	19.5.2017
Cítov	9	0.25	6.6.2015	18.5.2017
Straškov ^b	12	0.25	22.5.2016	19.5.2017
Vražkov ^a	14	0.25	22.5.2016	19.5.2017
Březno ^b	16	0.5	2004-2006	19.5.2017
Vražkov ^b	19	0.25	22.5.2016	19.5.2017

Table 2.1: Dates of screening of each location in dataset.

fortably without excessive zooming. The dataset can be viewed in Figure E.1, Figure E.2 and Figure E.3 in Appendix E, and is supplied in full resolution on the medium enclosed to this work.

The resulting crop mark dimensions are anywhere from 54 to 279 pixels in width and the wall thickness ranges from 4 to 12 pixels in images of size 1024×1024 pixels. The exact dimensions of each crop mark of interest are shown in folder *data/dimensions* on the enclosed medium. The phenological stages of the vegetation cover differ based on the chosen year and location. The screening dates for each image in the data is listed in Table 2.1, the CIR images screened in 2017 were chosen as they are the most recent and have the highest resolution. Because of the final image dimensions and zoom that we chose, we decided to not use examples near Třeboutice, Rakovice, Nechanice and Hořín shown in figure in Figure E.5 in Appendix E, as their much larger dimensions make unified enhancements of crop marks in image pre-processing impossible. Additionally, example near Hostinné was not visible in any aerial images provided by free mapping services. To further narrow the scope of this work, we eventually incorporated only crop marks with approximately rectangular and circular shapes.

2.4.2 Irrelevant structures and negative examples

The final dataset of positive examples contain a large amount of noise from large structures as roads and forests, harvester rails, transmission towers and naturally caused crop marks. To test whether our algorithms are able to correctly discriminate between valid archaeological sites and natural variations in vegetation cover, the final dataset was supplemented with negative examples of crop marks of presumably natural origin. The negative examples are shown in Figure E.4 in Appendix E.

Methods

This chapter offers an overview of the methods considered, implemented, and used to conduct experiments on the dataset.

3.1 Image pre-processing

Our collected data contain large amounts of noise and variation and as such, image processing methods and techniques can be used to suppress the influence of noise and irrelevant structures and also to balance the differences between each location, resulting in a more homogenous dataset.

Throughout this work, a 2-dimensional gray-scale image is denoted as scalar function

$$I[x, y] = i \tag{3.1}$$

where i is the intensity of pixel with x, y coordinates. Multispectral images in n -dimensional color space are denoted as

$$I_c[x, y] = [c_1, c_2, \dots, c_n]^T . \tag{3.2}$$

3.1.1 Noise removal

Image smoothing is a technique that aims to suppress noise or high frequencies in the input image. Noise can be introduced to the image in form of image degradation through processing, i.e., artifacts caused during capture or by compression. Alternatively, natural parts of the captured scene, such as irrelevant structures or texture changes, can be considered noise as well, and appropriate concealing of such features is desirable for better solving of a computer vision task. As image smoothing employs the redundancy of data, the technique is only helpful when the spatial resolution of the input image is adequate.

One of the simplest and effective methods of noise filtration is a convolution of the input image with a smoothing kernel. Operator $*$ denotes convolution,

3. METHODS

for 2D image I and convolution kernel G defined in its discrete form as:

$$(I * G)[x, y] = \sum_m \sum_n I[m, n] * G[x - m, y - n]. \quad (3.3)$$

Gaussian filter

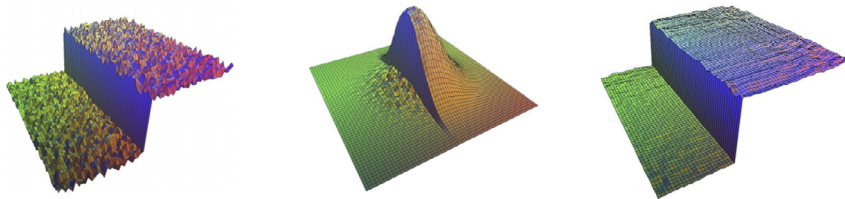
The Gaussian smoothing kernel is widely used for random noise removal and detail blending. The operation computes the value of each pixel as the weighted sum of its neighbors, with direct neighbors having the most influence. The parameter σ , the standard deviation of the Gaussian distribution, in combination with the kernel size regulates the strength of the final blur. The Gaussian kernel is computed by approximating the 2D Gaussian function to a discrete version [27]. Presented is the Gaussian function and an example of a Gaussian kernel of size 3×3 normalized to have a unit sum:

$$Gauss(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.4)$$

$$H_G = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.5)$$

Median filter

The median filter, in the form of a sliding window, replaces each pixel with the median intensity value of its neighbors. This non-linear technique, in comparison to the Gaussian smoothing, reduces the blurring of edges but is more computationally demanding. The median value is not affected by individual noise spikes in the region and as such is ideal for salt and pepper noise removal. The blur can be applied in iterations to strengthen the smoothing effect while preserving the window size [5].



(a) Input image I (b) Weighted kernel G_b (c) Filtered image

Figure 3.1: Edge-preserving bilateral filter, images sourced from [4].

Bilateral filter

The bilateral filter is another edge-preserving non-linear method of image smoothing [4]. The filter employs properties of Gaussian smoothing in combination with a function of pixel difference. For each pixel, the spatial Gaussian smoothing kernel is weighted by the difference in intensities of the central pixel and its neighbors. The weighting function returns low values for large differences in intensities and vice versa, effectively suppressing the smoothing in areas of presumed edges. The common function of pixel differences is the Gaussian function. The process is illustrated in figure Figure 3.1. The filtering operation can be defined as:

$$(I * G_b)[x, y] = W \sum_m \sum_n I[m, n] \cdot G[x - m, y - n] \cdot b(F[m, n] - F[x, y]) \quad (3.6)$$

where W is the normalization factor:

$$W = \sum_m \sum_n G[x - m, y - n] \cdot b(F[m, n] - F[x, y]) \quad (3.7)$$

and function b is the pixel similarity function. The Gaussian function is often used. The impact can be set by tuning its standard deviation parameter σ_b :

$$b(s) = \frac{1}{2\pi\sigma_b^2} \cdot e^{-\frac{s^2}{2\sigma_b^2}} \quad (3.8)$$

The computational complexity of this operation is higher than that of regular convolution, as the function of pixel intensity difference must be computed separately for each pixel, and thus rendering common improvement techniques, such as employing kernel separability or transformation to Fourier domain space, non-applicable [4].

3.1.2 Thresholding

Thresholding is a simple segmentation technique that is used to separate regions of fixed intensity from background by setting an intensity threshold τ . The most basic form of thresholding is a global threshold applied to a grayscale image and results in a binary output image:

$$I_{thr}[x, y] = \begin{cases} 1, & \text{if } I[x, y] > \tau \\ 0, & \text{otherwise} \end{cases} \quad (3.9)$$

Correct threshold selection is crucial for successful segmentation. However, even in simple scenes, there are likely to be levels of intensities shared between background and objects, leading to undesirable regions of misclassification. Moreover, in many applications images are captured under sub-optimal conditions resulting in non-uniform lighting. In cases as such, adaptive thresholding is often used to compensate for different lighting conditions in different

areas of an image. The threshold value is determined locally in a region of a predetermined size, while the global level is controlled with parameter c . The operation is shown in Equation 3.10, where \star denotes cross-correlation and G is a Gaussian kernel.

$$I_{thr}[x, y] = \begin{cases} 1, & \text{if } I[x, y] > (\tau(x, y) - c) \\ 0, & \text{otherwise} \end{cases} \quad (3.10)$$

$$\tau(x, y) = (I \star G)[m, n] = \sum_x \sum_y I[x, y] \cdot G[m + x, n + y]$$

3.1.3 Edge detection

Edges are defined as regions where the intensity of a grayscale image abruptly changes. Edge accentuation or detection is often used to improve image interpretation, as dominant contours are cleared of often less informative features such as depth and color.

The Sobel operator belongs to a group of edge detecting operators based on approximation of the first derivative of the image function. The operator combines a noise suppressing smoothing operation with an estimation of the first derivative for each pixel of the input image. The Sobel kernels sized 3×3 are presented in Equation 3.11; larger sizes are possible.

$$H_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, H_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.11)$$

Often, only the vertical and horizontal kernels are used to detect horizontal and vertical lines in an image. The resulting response image is defined as $S_n = I \star H_n$ and combined gradient magnitude of the image can be computed as $I(x, y) = \sqrt{S_x^2(x, y) + S_y^2(x, y)}$ or $I(x, y) = |S_x(x, y)| + |S_y(x, y)|$ for performance sake [5].

The Canny edge detector is one of the most used and reliable edge detectors [28]. It is based on zero-crossings of the image function second derivative. It is robust against noise and returns a binary image with detected edges' widths of 1 pixel, regardless of the original dimension. The Canny edge detection algorithm consists of multiple steps:

1. **Noise reduction** performed by smoothing the image with a Gaussian kernel.
2. **Estimate of local gradient magnitude and direction** for each pixel. Gradient direction is always perpendicular to the detected edge. This

step may be performed by Sobel or alternative operator H_n in various directions n .

3. **Non-maximal suppression** is performed to eliminate redundant edge pixels, resulting in a thinned image. A true edge location is determined by finding the local maximum by zero-crossing. The original image I is convoluted with the operator H_n differentiated¹¹ in the direction n , resulting in the second derivative of the input image. The local maximum is where the second derivative is 0.
4. **Hysteresis** thresholding using minimum and maximum threshold parameters. The lower (soft) threshold suppresses edges caused by noise while the upper (hard) threshold is the lowest value of magnitude for edge pixel to be considered correct. Edge pixels with magnitudes between thresholds are filtered by their adjacency to a true edge.

The quality of results acquired by the Canny detector is dependent on the suitable determination of thresholds, as well as the scale of the derivation kernel.

3.1.4 Principal component analysis

The principal component analysis, or PCA, is one of the most widely used linear techniques used for dimensionality reduction in statistics and also in image processing. The PCA operation transforms possibly correlated variables into a new set of uncorrelated variables called *principal components*. The first principal component accounts for as much variability in the original data as possible, and each subsequent component accounts for as much of the remaining variability [5]. This property is used for two main purposes - for transforming variables into uncorrelated space, or approximating dataset to a lower-dimensional space while retaining key information. In the processing of aerial images specifically for archaeological purposes, PCA is often used to transform CIR or other multichannel space to grayscale with component mixing to highlight specific conditions such as soil and crop marks [8].

The PCA is applied to normalized image data. Let $X \in \mathbb{R}^{N,M}$ be a representation of image I_c , where N is the number of columns (image channels) and M is the number of pixels in each channel. The empirical mean of each channel is subtracted from the corresponding column of X and the covariance matrix Σ is computed.

$$E[X_{\bullet,n}] = \frac{1}{M} \sum_{m=1}^M X_{m,n}, \text{ for } n = 1, \dots, N \quad (3.12)$$

¹¹Only the kernel needs to be differentiated again, in lieu of the image I , given the differentiation property of convolution: $(I * G)' = I' * G = I * G'$ [29].

$$\Sigma = (X - E[X])^T(X - E[X]) \quad (3.13)$$

The covariance matrix $\Sigma \in \mathbb{R}^{N,N}$ is symmetric and positive definite and as such is guaranteed to have real eigenvalues [5]. The associated eigenvectors v_1, v_2, \dots, v_N in matrix form are the basis vectors of the new vector space of principal components. The original data X are then transformed from N -dimensional vector space to a new N -dimensional vector space by multiplication:

$$P = X \cdot (v_1, v_2, \dots, v_N)^T \quad (3.14)$$

To acquire the transformed data from matrix P to image representation, the pixels of each utile principal component column must be ordered to a grid of the original image I_c dimensions. The drawback of this method, as is the case for any linear non-local transform, is that because each channel of the processed image is flattened into a single row of pixels, the spatial information and its importance is lost. The relations between neighboring pixels are not taken into account, which means that significant outliers may affect the entire resulting image [5].

3.1.5 Morphological operators

Mathematical morphology is an area of image analysis that operates on objects in image as point sets and employs the properties of their connectivity and shape. Its is widely used in applications where shape preservation is crucial, such as microbiology or criminology. A binary input image I is represented as a point set $X = \{(x, y) | \forall x, y \in I[x, y] = 1\}$. The binary input image is analyzed by a binary image of a certain shape, often rectangle, circle or linear segment, called the *structuring element* (SE) and denoted B . This process is called a *morphological transformation* and is denoted Ψ . Morphological transformations are dual, meaning for each transformation $\Psi(X)$ there is a transformation $\Psi'(X)$ such that:

$$\Psi(X) = (\Psi'(X^C))^C \quad (3.15)$$

Dilation and erosion

Dilation and erosion are the primary morphological operators from which more complex transformations are derived. Dilation operator \oplus combines two sets using vector addition and erosion is the dual transformation denoted \ominus using vector subtraction:

$$\begin{aligned} X \oplus B &= \{p \in \mathcal{E}^2 : p = x + b, x \in X, b \in B\} \\ X \ominus B &= \{p \in \mathcal{E}^2 : p + b \in X \text{ for every } b \in B\} \end{aligned} \quad (3.16)$$

Opening and closing

While dilation and erosion transformations are dual, they are not inverse, meaning, that dilation of an eroded picture with the same structuring element does not yield the original picture. Instead, erosion followed by dilation creates morphological transformation called opening, denoted \circ , and dilation followed by erosion creates transformation called closing, denoted \bullet .

$$\begin{aligned} X \circ B &= (X \ominus B) \oplus B \\ X \bullet B &= (X \oplus B) \ominus B \end{aligned} \tag{3.17}$$

Top hat

Top hat transformation is the result of the residue of morphological opening as compared to the original image X :

$$\Psi_{tophat}(X) = X \setminus (X \circ B) \tag{3.18}$$

The operation can be imagined as extraction of intensity peaks from the original image, with their spatial scale determined by the morphological opening. White top hat retrieves light objects on a dark background, a black top hat does the opposite [5].

3.2 Local features

The process of finding local features, i.e. interest points, is the first stage of many computer vision applications [30]. For rectangular object shapes, locating corners and subsequent interest point analysis are often used in object recognition and image matching. Corners can be located by using local feature detectors. In this work, we analyze the possible contribution corner location could have to the task of crop mark detection.

A corner in an image can be rather vaguely described as a region, where two different dominant edge (gradient) directions meet. Edge detectors alone cannot be used to determine corners in an image, as the gradient at the tip of a corner is ambiguous. This property is employed by corner detectors [5].

3.2.1 Harris corner detector

The Harris corner detector is based on differences in intensity shifts in vertical and horizontal directions. For grayscale image I , let I_x, I_y be image derivatives in x, y directions. The *cornerness measure*, indicating the presence of a corner, is measured for an image window w . Usually, an isotropic window, such as

3. METHODS

Gaussian kernel is selected [5]. The local second-moment matrix M , the same size as w , is computed for each pixel of input image I :

$$M_{x,y} = \sum_{x,y} w(x,y) \cdot \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (3.19)$$

Matrix M is symmetric and positive semi-definite, meaning it has two real eigenvalues λ_1, λ_2 , in which the main modes of variations are reflected. [5] Three cases can occur:

1. Both eigenvalues are small, which indicates a flat region.
2. One eigenvalue is large and the other small, indicating a ridge in one direction.
3. Both eigenvalues are large, i.e. shift in any direction causes change, which indicates a corner.

The properties can be summarized into a response function R , the corner-ness measure, to differentiate corners from non-corners by setting an appropriate threshold. When eigenvalues λ_i are explicitly known, few alternative measures R can be calculated:

$$\begin{aligned} R_1 &= \lambda_2 \\ R_2 &= \frac{\lambda_1 \cdot \lambda_2}{\lambda_1 + \lambda_2} \\ R_3 &= \lambda_1 \cdot \lambda_2 - \lambda_1 + \lambda_2 \\ R_4 &= \lambda_1 \cdot \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 = \det(M) - k \cdot (\text{trace}(M))^2 \end{aligned} \quad [31] \quad (3.20)$$

The recommended value for parameter k differs across literature but is usually set in interval $[0.02, 0.2]$ [5]. The Harris detector shows to be robust to noise and lighting variations and is shift and rotation invariant [32].

The classic Harris corner detector is intensity based but can be extended to multispectral images. In their works Sirisha, Sandhya and Sebe, Weijer conclude, that while most salient points are found using just intensity, the added color information significantly improves corner distinctiveness and may be used as a key descriptor in image matching algorithms. However, the introduction of color can also lead to less stable results, as the detector becomes less invariant to changes in illumination [31, 33].

The authors of [31] extend the Harris detector on RGB channel image, while in [33] authors implement and test the detector for the opponent color space and for the m-color ratio space. The opponent color space is defined as:

$$\begin{aligned} O_1 &= L \text{ the luminance component} \\ O_2 &= G - R \text{ as the red-green channel} \\ O_3 &= B - (G + R) \text{ as the blue-yellow channel} \end{aligned} \quad [34] \quad (3.21)$$

The color ratio system already incorporates the image derivatives and can be used directly as input in the Harris cornerness matrix [33]. For standard RGB image, the conversion is defined for channel pairs RG, RB and GB. The ratio is computed for each direction x, y separately. Presented is the equation for one resulting ratio channel in one direction, where x_1, x_2 denote two neighboring image pixels:

$$O_{RG} = \frac{R_{x_1} G_{x_2}}{R_{x_2} G_{x_1}} \quad [33] \quad (3.22)$$

As corner detectors are, by definition, not very robust, large redundancies are to be expected [5].

Non-maximum suppression In salient point and object detection algorithms often classify multiple overlapping regions in place of a presumed point or object discovery. To reduce the number of overlapping entities, a method of non-maximum suppression is used to discard redundant entities. The exact implementation differs in respects to the task at hand, but in principle the method is as follows:

1. Each of presumed positives returned by a detection algorithm is evaluated by appropriate quality function.
2. For each presumed positive inspect all adjacent or overlapping positives.
3. If the quality measure of any of the adjacent positives exceeds that of the one under inspection, the inspected positive is marked for deletion.
4. When all overlapping regions have been inspected, all positives marked for deletion are removed from the list of positives [5].

3.3 Hough transform

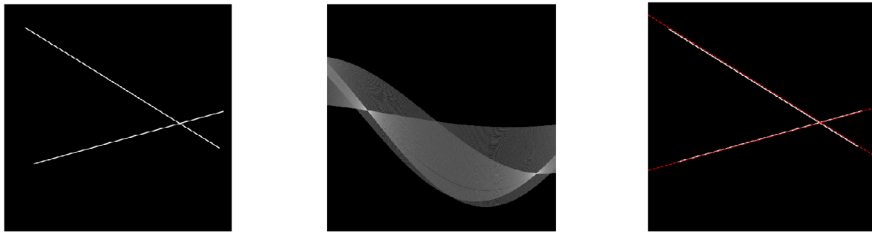
The Hough transform [35] (HT) is a widely used method of object detection and location based on known shape and size [36]. In its original form, HT was designed to detect lines and circles expressed in their parametric forms, and later extended to detect any 2D shape given its full description in a form of a template. It is robust in the presence of noise and additional structures and is capable of recognizing even partial or slightly deformed shapes [5].

The method is based on transformation of the image I into a polar parameter space, i.e. an accumulator space or Hough space. The location of an object in image I is determined by a voting mechanism, where each edge pixel of image I corresponding with the given equation, is one vote in the accumulator array. The final detection takes place by detecting local maxima in the parameter space, as illustrated in Figure 3.3. Because the parameter space usually has duplicate local maxima per shape, it is recommended to smooth the accumulator array with a Gaussian kernel [37].

3. METHODS

Pre-requisites of the Hough transform algorithm:

- The input image I is to be pre-processed with an edge detector to extract the outer contours of captured objects. Canny edge detector or Sobel filter are often used for this task.
- The shape and size of a detected object is known and expressed as one or multiple parametric equations.



(a) Input image I .

(b) Hough space.

(c) Detected lines.

Figure 3.2: Line detection via the Hough transform.

3.3.1 Hough transform for lines and circles

The parametric equations for a line and circle respectively are:

$$\begin{aligned}x \cdot \cos \theta + y \cdot \sin \theta &= r \\(x - a)^2 + (y - b)^2 &= r^2\end{aligned}\tag{3.23}$$

For each edge pixel (x_i, y_i) in input image I , the polar coordinates r, θ are derived and plotted to the Hough parameter space. In polar space, each input point of I becomes a sinusoid curve. In case of a line detection, collinear points in the input image I result in intersecting sinusoid curves in the parameter space. Therefore, peaks in the accumulator array correspond to lines in the original image and the intensity of the peak directly corresponds to the line length. In case of circles, where the parameters a, b are the coordinates of the circle center and r is the radius, the accumulator space becomes 3 dimensional. Because with each added parameter, the accumulator space gains an extra dimension, the original Hough transform becomes computationally unsustainable for more complex shapes.

3.3.2 Generalized Hough transform

To detect arbitrary shapes, the Hough transform was extended for boundaries for which a parametric representation does not exist. The generalized Hough transform [37] offers a solution as it incorporates a learning stage. Assuming the shape, size and orientation of the object are known, a mask, or template can be constructed.

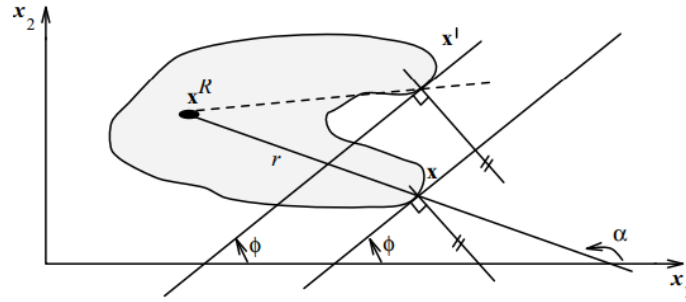


Figure 3.3: Generalized Hough transform: construction of the R table. Image obtained from [5].

During the learning stage, a reference table R which contains description of the object is constructed. From a given reference point \mathbf{x}^R inside the depicted shape, a line is constructed to every border point \mathbf{x} of the object shape. All border points (r, α) , given by the angle α of the line $(\mathbf{x}^R, \mathbf{x})$, and the distance r of the border point from the reference point \mathbf{x}^R , are stored in the R table as a function of the border direction $\phi(\mathbf{x})$. R table construction is illustrated in Figure 3.3.2. Each border point \mathbf{x} that has a significant edge in the direction $\phi(\mathbf{x})$ has coordinates of a potential reference points $\{x_1 + r(\phi) \cos(\alpha(\phi)), x_2 + r(\phi) \sin(\alpha(\phi))\}$ [37]. Depending on the shape of the boundary, there may be more than one (r, α) pair for every $\phi(\mathbf{x})$ and so the R table format is:

$$\begin{array}{l} \phi_1 \quad (r_1^1, \alpha_1^1), (r_1^2, \alpha_1^2), \dots, (r_1^{n_1}, \alpha_1^{n_1}) \\ \phi_2 \quad (r_2^1, \alpha_2^1), (r_2^2, \alpha_2^2), \dots, (r_2^{n_2}, \alpha_2^{n_2}) \\ \dots \\ \phi_k \quad (r_k^1, \alpha_k^1), (r_k^2, \alpha_k^2), \dots, (r_k^{n_k}, \alpha_k^{n_k}) \end{array}$$

If the scale and rotation of the object vary, the number of parameters increases from two (x_1^R, x_2^R) to four with the additional orientation τ and scale S parameters.

With the constructed R table, the generalized Hough transform algorithm is as follows:

1. Initialize an accumulator array $A(x_1, x_2, \tau, S)$ to zero.
2. For each pixel (x_1, x_2) from input image I , determine the edge direction $\phi(\mathbf{x})$. For all potential reference points \mathbf{x}^R , increase accumulator for all possible values of rotation and size change:

$$\begin{aligned} \mathbf{x}^R &= \{x_1 + r(\phi + \tau)S \cos(\alpha(\phi + \tau)), x_2 + r(\phi + \tau)S \sin(\alpha(\phi + \tau))\} \\ A(\mathbf{x}^R, \tau, S) &= A(\mathbf{x}^R, \tau, S) + 1 \end{aligned} \tag{3.24}$$

3. The locations of detected shapes are given by local maxima in the smoothed accumulator array A .

The Hough transform, albeit computationally expensive for brute force implementation, can search for multiple object occurrences in a single process, and also is easily parallelizable, making it widely applied even in real-time applications.

3.4 Rectangular enclosure detection

The detection of livestock enclosure remnants in high-resolution aerial images, as proposed by Zingman, Lambert in [6]. Their approach is based on extensive pre-processing of the input image, line segment detection via the Hough transform, and subsequent grouping and evaluation of the resulting geometric properties of nearby line configurations. As the process is described in great detail in [38, 6], only a synopsis for guidance is provided in this text and more details regarding implementation are offered in chapter 4 and chapter 5. The algorithm (RE) was implemented and tested because it handles a task much similar to the detection of crop marks. Like rectangular crop marks, the ruins of livestock enclosures are composed of linear wall, often heavily fragmented, the angles between walls may due to design or deterioration deviate from 90 or 180 degrees and one or more sides may be missing completely. The authors work with satellite and aerial images of 0.5m resolution per pixel, where the width of the linear segment of a ruin does not exceed 2 pixels. As we have significantly higher resolution at our disposal, often needed to even manually distinguish crop marks in vegetation, this approach may prove to be impractical for our task. Nonetheless, the authors overcame some shared complications with their approach, such as an amount of irrelevant structures with much higher contrast than of the enclosures and very limited number of positive examples.

3.4.1 Extraction of ridge features and candidate points

The RE detection algorithm is based on finding approximately rectangular shapes around selected candidate points in extensively processed image. First, the authors use top hat morphological operator with a square SE to process the original grayscale aerial photography. To enhance linear segments the authors use their own Morphological Feature Contrast transformation (MFC) [39] with a square SE. They follow it with ridge extraction by selecting a point-wise maximum from morphological openings with linear SE of a set length at different orientations. This step enhances linear features in the image and suppresses non-linear structures. The steps are described in algorithm 1. The MFC operator for input set X and two SE of sizes r_1, r_2 is defined as:

$$\begin{aligned}\Psi_{MFC}^+(X) &= X - (X \bullet B_{r_1}) \circ B_{r_2} \\ \Psi_{MFC}^-(X) &= (X \circ B_{r_1}) \bullet B_{r_2} - X \\ \Psi_{MFC} &= \Psi_{MFC}^+ + \Psi_{MFC}^-\end{aligned}\tag{3.25}$$

Algorithm 1: Ridge extraction

Input: Grayscale image I , sizes of SE and list of angles for opening.

Output: Grayscale image I - binary map of ridge features.

$I \leftarrow \Psi_{tophat}(I)$

$I \leftarrow \Psi_{MFC}(I)$

$openings \leftarrow \emptyset$

for a **in** $angles$ **do**

$openings \leftarrow openings \cup \{I \circ B_a\};$

end

for (x, y) **in** I **do**

$I(x, y) = \max\{o(x, y) \mid \forall o \in openings\};$

end

return I

The authors propose using a local window placed at sparsely sampled candidate points, to help reduce the time and memory complexity of processing all image data with Hough transform. The location of candidate points and the size of the window is determined adaptively, such that in sparse areas only couple points are located and the size the window size gets progressively bigger. Vice versa, in areas with lot of ridge features, the candidate points are denser. The sampling is performed on thinning of the ridge image complement and the point density is derived from the distance to the closest ridge. Example is shown in Figure 3.4.

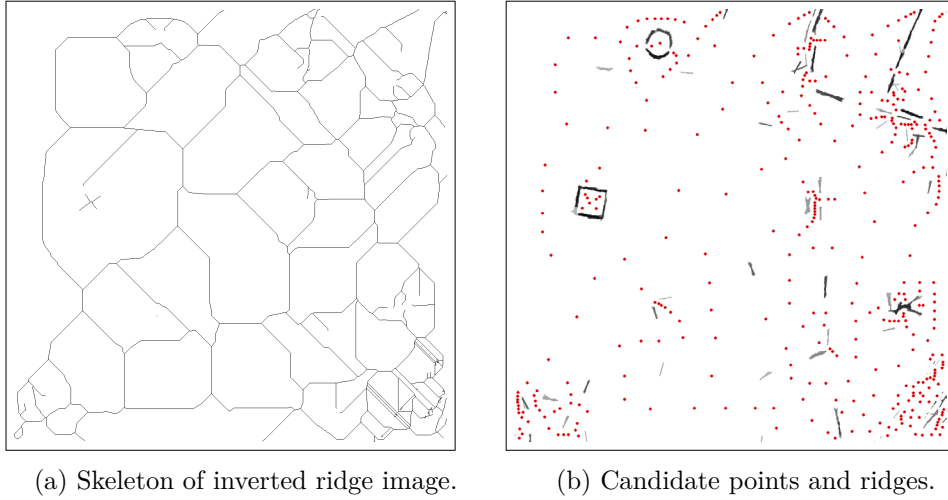


Figure 3.4: Rectangular enclosures: Sampled candidate points, colors inverted.

3.4.2 Graph representation and rectangularity measure

For window centered at every candidate point, line segments are extracted by Hough transform, followed by linking and line approximation to merge potentially fragmented ridge features. The authors encode linear segments and their spatial relations into a graph G , where each node v is a linear segment $S(r, \theta, l)$ and an existing edge between two nodes $e(u, v)$ denotes a valid connection between segments. A valid connection is defined as:

- Linear segments S_k, S_j are either approximately perpendicular or approximately parallel to each other.
- Extension of all linear segments in configuration forms nearly convex contour.

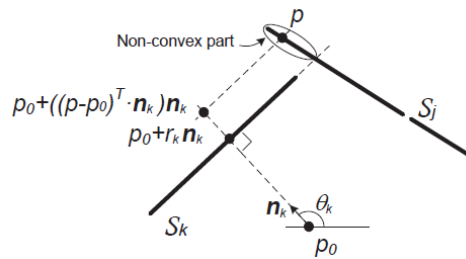


Figure 3.5: Convexity constraint illustration. Image from [6].

To inspect the convexity constraint for each pair of segments, illustrated in Figure 3.5, the authors propose pair-wise convexity function τ measuring

the overlap of two segments in regard to the candidate point p_0 :

$$\begin{aligned}\tilde{\tau}_{k,j} &= \frac{1}{l_j} \sum \mathbb{1}((p - p_0)^T \cdot n_k - r_k) \\ \tau_{k,j} &= \max(\tilde{\tau}_{k,j}, \tilde{\tau}_{j,k})\end{aligned}\tag{3.26}$$

Each clique G^c of a graph G corresponds to a valid configuration of linear segments [6].

Valid configurations are evaluated by a rectangularity measure $f_R(G^c)$ and the clique with the highest value is selected as optimal. Additional feature $f_S(G^c)$ is proposed to quantify the configuration size [6]. During our testing, we implemented an additional feature $f_D(G^c)$ meant to quantify the *compactness* of the configuration, based on the idea that the sum of line segments should not be much smaller than the lengths of presumed diagonals. Measures are presented in Equation 3.27, points p_1, p_2 are the endpoint coordinates of a segment and $dist()$ denotes the distance between two points in Euclidean space. Variable $\beta_{k,j}$ is the angle between two line segments and f_{cv}, f_{90}, f_{180} are mode functions described in [6].

$$\begin{aligned}f_R(G) &= \left(\sum_{k,j}^{E(G)} l_k l_j f_{90}(\beta_{k,j}) f_{cv}(\tau_{k,j}) \cdot \sum_{k,j}^{E(G)} l_k l_j f_{180}(\beta_{k,j}) f_{cv}(\tau_{k,j}) \right)^{\frac{1}{4}} \\ f_S(G) &= \frac{\sum_j l_j r_j}{\sum_j l_j} \\ f_D(G) &= \frac{2 \sum_j l_j}{\sum_j dist(p_{0j}, p_{1j}) + dist(p_{0j}, p_{2j})}\end{aligned}\tag{3.27}$$

The authors of the paper deduce the adjusted rectangularity feature \hat{f}_R , referred to in this work as f_{RS} as the optimal linear combination of the rectangularity and size feature. Resulting configurations, each evaluated by feature vector (f_R, f_S, f_D) , are used as data points for a chosen classification model, to help discriminate detected rectangular features from irrelevant structures and clutter [6].

3.5 Classification and evaluation

Two of the mentioned methods, RE and GHT, return one or more measures, called features, that indicate the perceived quality of the detected region of interest (ROI). Because we expect a large amount of false positives to be extracted, to determine whether the ROI contains a crop mark of archaeological importance, a classifier is trained to help discriminate crop marks from noise. This type of problem is known as a binary classification. This process of utilizing classifier as the last step of the image processing approach is known as object or pattern recognition [5].

The role of a classifier is to assign an object to a class, based on the object's formal description in the form of a feature vector. The choice of the most appropriate classifier for each task is dependent on the number of classes and the amount and type of available features. Even so, various classifiers have many properties in common. Most classifiers are utilized in two phases: *learning* and *classification*. During the learning phase, the classifier is supplied with a *training* set, subset of the complete dataset, with a known class for each data point. During the learning phase the classifier optimizes its parameters to be able to predict the true class for each data point with as low error rate as possible. In the subsequent classification phase, the classifier is supplied data without known class. The learned classifier proceeds to predict the class for each data point. The approach to training utilized in this work and described above is known as supervised learning. The classification models selected for the purposes of this work are the Support Vector Machine (SVM) model and Random Forest (RF) model.

3.5.1 Support Vector Machine

Support Vector Machine [40] is a popular choice of classifiers for dichotomic linear classification problems. The classifier aims to maximize the margin between two classes, by dividing them with an optimally placed discrimination surface - a hyperplane in the n -dimensional feature space. The data points from each class closest to the discriminating hyperplane are called support vectors. We selected the SVM classifier as it aims to reduce the risk of overfitting, which is crucial for our small dataset. The SVM can be used when features are continuous and standardized in interval $[0, 1]$.

For a binary problem, let each feature vector be denoted \mathbf{x} and the class identifiers be denoted $\omega \in \{-1, 1\}$. Let the discriminating hyper-plane and the two parallel hyperplanes, maximizing the margin, be defined as:

$$\begin{aligned}\mathbf{w} \cdot \mathbf{x} + b &= 0 \\ \mathbf{w} \cdot \mathbf{x} + b &= 1, \mathbf{w} \cdot \mathbf{x} + b = -1\end{aligned}\tag{3.28}$$

The constraint that guarantees that no data points are present between the margins must be true for each data point \mathbf{x}_i :

$$\omega_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1.\tag{3.29}$$

To maximize the margin, that is to find the optimal solution, we need to minimize $\|\mathbf{w}\|$ which leads to an optimization problem.

In most cases, the two classes cannot be linearly separable to which a solution is to either tolerate some minimal misclassification rate or to employ the so-called kernel trick. The idea behind the kernel trick is to transform the feature vectors into a space in which a linear discrimination hyper-plane can be determined. Example is provided in Figure 3.6. The resulting SVM learning and classification stages are the following:

1. Training: An appropriate non-linear kernel function $k(x_i, x_j)$ must be selected.
2. Training: The $\|\mathbf{w}\|$ subject to the constraint given in Equation 3.29 is minimized by maximizing the kernel-modified Lagrange function with respects to α_i , with constraints $\sum_i^N \alpha_i \omega_i = 0$ and $\alpha_i \geq 0$:

$$L(\mathbf{w}, b, \alpha) = \sum_i^N \alpha_i - \frac{1}{2} \sum_{i,j}^N \alpha_i \alpha_j \omega_i \omega_j k(\mathbf{x}_i, \mathbf{x}_j) \quad (3.30)$$

3. Training: The non-zero weights α_i and corresponding support vectors x_i are stored.
4. Classification: For each data point \mathbf{x} , compute the discrimination function $f(\mathbf{x})$, using the support vectors $\mathbf{x}_i \in SV$ and their corresponding weights α_i :

$$f(\mathbf{x}) = \sum_{i \in SV} \alpha_i \omega_i k(\mathbf{x}_i, \mathbf{x}) + b \quad (3.31)$$

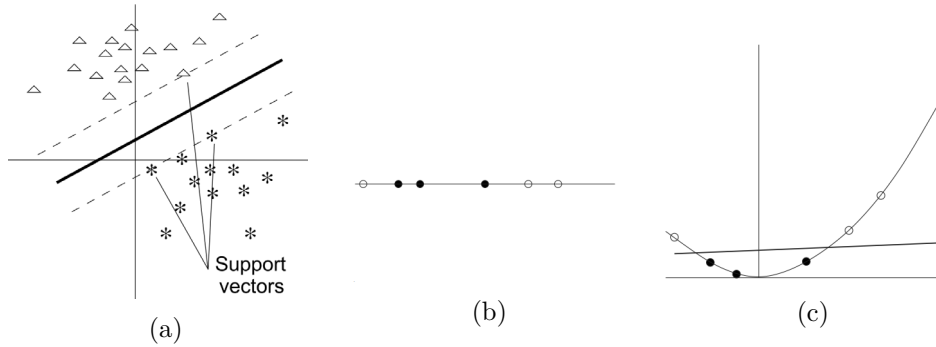


Figure 3.6: SVM binary classification: (a) Example of optimal linear discrimination, thick line denotes the discriminating hyper-plane, (b) case of linearly non-separable classes in 1D, (c) application of the kernel trick to create a linearly separable problem. Images sourced from [5].

Popular kernel functions are:

Polynomial kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d \quad (3.32)$$

Gaussian kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \quad (3.33)$$

Gaussian radial basis (RBF) kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2} \quad (3.34)$$

3.5.2 Random Forest

Random forest [41] is an ensemble classifier, consisting of a combination of decision tree classifiers. Each decision tree in the ensemble is trained independently on a randomly sampled subset of the training data. The prediction result for each data point is determined by a majority vote. The random sampling employed in the construction of the random forest results in better robustness compared to a single decision tree classifier, which has the inclination to over-fit. As such the RF classifier is, with the right parameters, robust to outliers and can be used with a mixture of categorical and numerical features.

Decision tree is a simple and transparent classifier with a very well interpretable structure. A decision tree classifier consists of internal split nodes, corresponding to a decision rule, and leaf nodes, corresponding to a predicted class. The tree structure is illustrated in Figure 3.7. The decision rule for each node is determined for a single feature to maximize the information gain. The process of building the decision tree is also its learning stage. Widely used decision tree generating algorithm is ID3 [5].

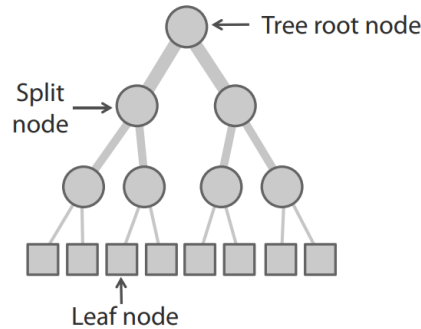


Figure 3.7: Decision tree structure. Image from [5].

3.5.3 Cross-validation

Regardless which classification model is used, our resulting dataset must be split into a training set, to train the model, and a validation set, for which the classes will be predicted and will serve as our final outcome. The method of cross-validation, or k-fold validation, ensures that prediction is made for all

data points in dataset. Because our dataset is very limited, we use the leave-one-out cross-validation method. The leave-one-out cross-validation refers to a case when $k = 1$; one site is selected for classification, and the rest of the dataset is used to train the model.

3.5.4 Evaluation

Given, how our research is limited by the low number of positive examples, it is expected that commonly used methods of statistical evaluation will not be applicable. On the other hand, the small size of our dataset makes manual evaluation possible. Therefore, we will use the classifier of choice to possibly remove falsely detected regions, and will evaluate the resulting outcome site by site.

Design

This chapter presents the algorithms deduced from each method outlined in chapter 3. We describe the general approach to the problem and outline the process, step by step, for each algorithm. The parameters of each algorithm and the problems we encountered with each method are further described in chapter 6.

There are three general approaches we explore: corner based detection, linear segment based detection and template based detection. For each approach, we discuss the inherent problems that may arise and present the methods of validation and evaluation of the final results.

The general pipeline applied to each detection approach consists of three steps: image pre-processing, detection phase and classification of detected regions as valid or false alarms. The outcome of each phase can be saved independently of the process, making each stage modular. We utilized this quality in our implementation and experiments, as we test multiple approaches to pre-processing, detection and classification.

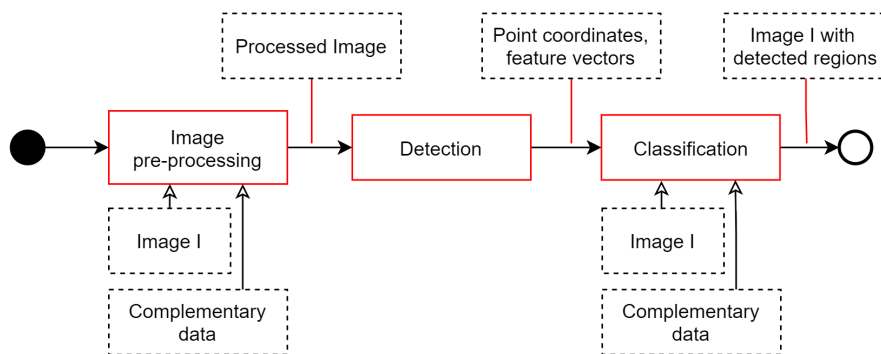


Figure 4.1: General data flow of the crop mark detection algorithm.

4.1 Corner based detection

The corner based detection is the most straightforward method presented in this work. By design, it is only applicable to crop marks with defined corners, which accounts for only half of our data set. We propose two approaches to classification of the detected corners as regions of interest. The general pipeline is illustrated in Figure 4.2.

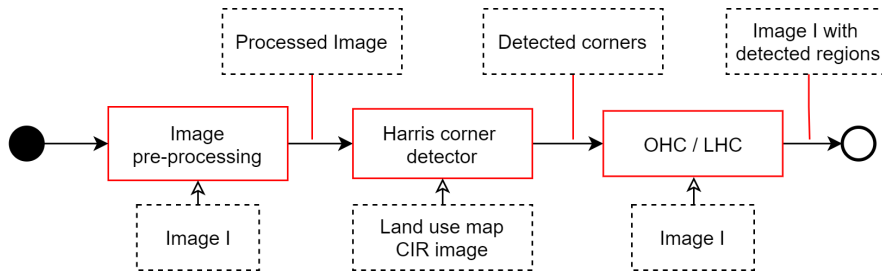


Figure 4.2: General data flow of the corner based detection algorithm.

4.1.1 Image pre-processing and corner detection

The original Harris corner detector was developed for grayscale images. Our input RGB image is converted to grayscale either by means of PCA, or by utilising only the red channel of the original image. The red channel, referred to as RED, was deemed suitable because pure green shades appear the darkest and the yellow areas appear the lightest in the red channel. The topic is discussed in more depth in a study by Gojda, O. et al. [42], focused on the pre-processing of crop mark images. The grayscale imaged resulting from PCA corresponds to the first principal component and is from now on referred to as PC1. The differences in performance of each approach are discussed in chapter 6.

For the color extended Harris corner detectors, the image was either left in the original RGB form or converted to the opponent color space or the m-color ratio space. The image pre-processing is focused on enhancement of contrast, to bring out the often very faint outlines of crop marks. Additionally, bilateral filter is used to smooth the areas between dominant edges. The pre-processing is illustrated in Figure 4.3.

Harris corner detector

The resulting pre-processed images are used as input to either the grayscale or color extended Harris corner detector.

A filtration of irrelevant regions is performed on the resulting corner set. The map of land use is used as a binary mask to the input image, only allowing areas designated to agricultural use. In addition, the CIR image is used to



Figure 4.3: HC: Image pre-processing applied to the red channel.

filter out additional areas that do not contain vegetation. This step was found helpful in case the site image contains newly build structures, not yet present in the map of land use, or trails and unofficial roads. This step is performed prior to the non-maximal suppression. The resulting set of corners is provided as input to the two algorithms described in the following sections.

4.1.2 Detection with 4 corners

The detector described in this section presumes that rectangular crop marks have all four corners intact. The detection is based on considering all possible combination of 4 corner configurations and gradually filtering out non-valid combinations. Each valid structure must meet several conditions:

- Each pair of corners must have at least partial edge between them.
- Each opposing lines must be roughly parallel and of similar lengths.
- Adjacent lines share one corner and must be roughly perpendicular.

The algorithm is referred to as LHC in further text and is proposed in pseudo-code in algorithm 3 in Appendix D, the process is illustrated in Figure 4.4

4.1.3 Detection with 3 corners

In an ideal case, the corner detector would detect all 4 corners of a rectangular crop mark. However, because we already know that many of our examples

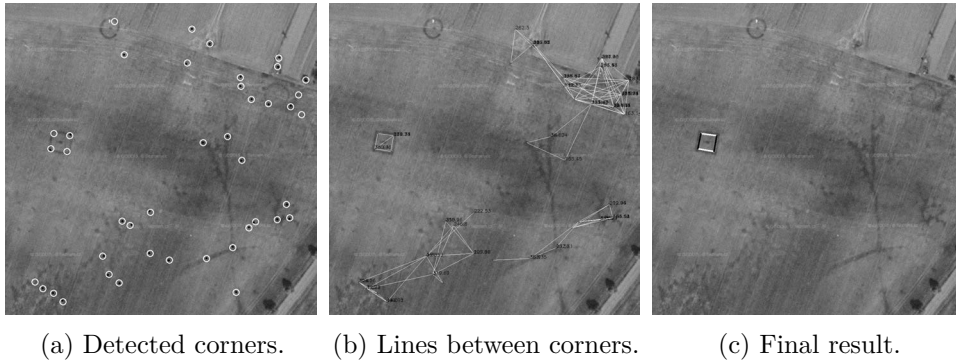


Figure 4.4: Detection with 4 corners.

do not have all four corners intact¹², we propose an alternative method that requires only three corners to be detected. Because without a fourth corner we cannot employ the parallelism constraint from the previous algorithm, we propose an additional constraint in the form of corner orientation. This constraint ensures, that neighboring corner orientations differ by roughly 90° , and that the angles of corners forming a diagonal are roughly opposite. In addition to the corner detection performed by Harris corner detector, the orientation of each detected corner must be defined. The corner orientation is the angle of the vector from the corner coordinate to the intensity centroid in the local neighborhood around the corner calculated using first order central moment.

The algorithm is referred to as OHC in further text and is proposed in pseudo-code in algorithm 4 in Appendix D, the process is illustrated in Figure 4.5

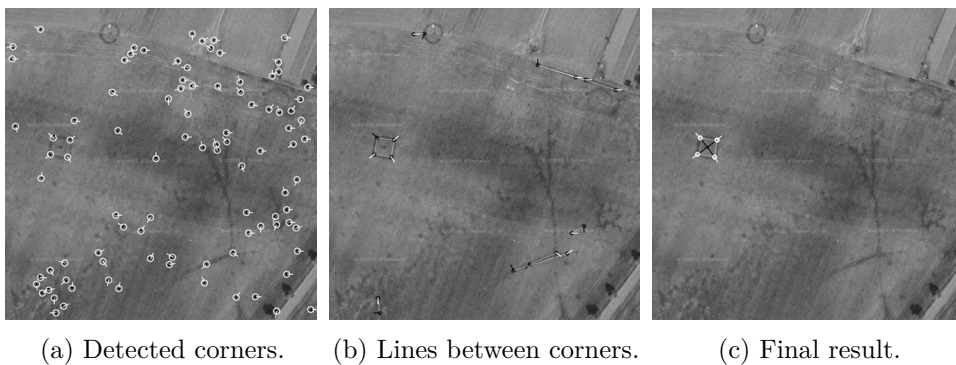


Figure 4.5: Detection with 3 corners.

¹²Which might happen for variety of reasons. According to our dataset it can be attributed either to disintegration of the subsurface remains, uneven vegetation planting or additional noise in the area of the image.

4.2 Linear segment based detection

The line segment based detection, referred to as the RE detection algorithm, is adapted from the work of Zingman et al., initially presented in [38] and later improved upon and reintroduced in [6]. The approach can be summarized in 5 steps, illustrated in Figure 4.6, where the detection phase is divided into sequence of three steps.

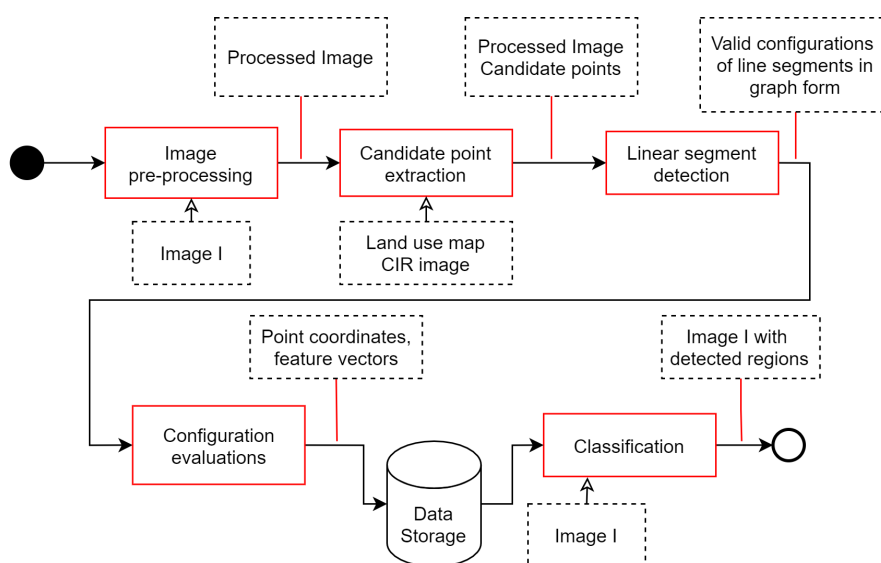


Figure 4.6: General data flow of the RE algorithm.

4.2.1 Image pre-processing

The algorithm is applied to a grayscale image. Again, both the red channel and the first principal component are compared in our experimental section in chapter 6. The image pre-processing methods are used in the following order:

Filtering with LU binary mask and CIR image is applied as the first step, so that irrelevant features do not seep into the rest of the image during the pre-processing. Contrast enhancement with parameters $\alpha = 2.5, \beta = -170$ is applied to the grayscale image and then several morphological transformations are applied.

First, a white top hat operator is used to extract small detail from the image. This extracted image containing small white details, is subtracted from the original image, which results in suppression of noise present in the

form of watermarks and other bright regions like plough furrows. Process is illustrated in Figure 4.7.

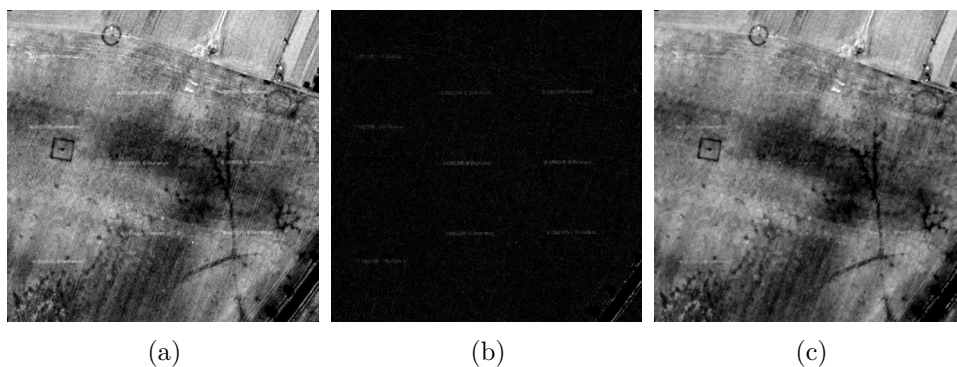


Figure 4.7: RE: (a) PC1, enhanced contrast, (b) white top hat, (c) white noise subtracted from (a).

After, a black top hat transformation is applied to extract dark regions corresponding to (positive) crop marks in the image and other features present. The image is then thresholded to filter out faint debris and closing operator is applied to better define the resulting shapes. Finally, histogram equalization is applied to intensify the effects of the filtration. Process is illustrated in Figure 4.8.

In the next step, line segment enhancement is applied as means to extract line-like features from the image. The process is already described in algorithm 1 in chapter 3. The minimal line length should be determined by this point, as shorter segments are filtered out from the image.

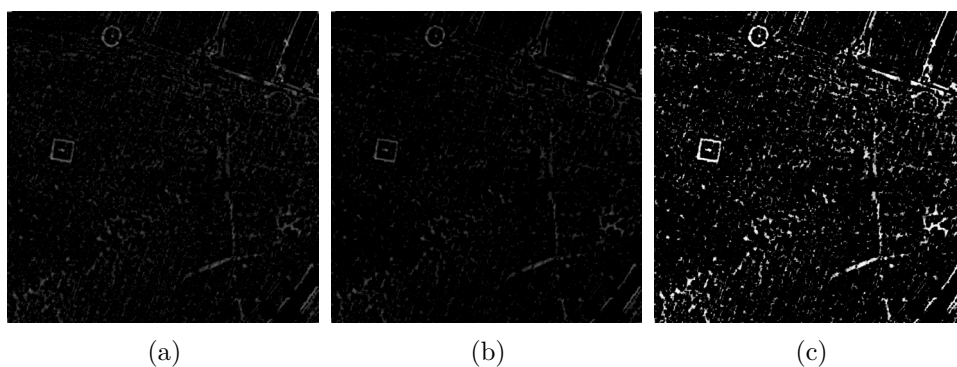


Figure 4.8: RE: (a) black top hat and threshold, (b) morphological closing (c) histogram equalization.

4.2.2 Detection

The detection stage of the RE algorithm is divided into three steps, each explained in the following sections.

4.2.2.1 Candidate point extraction

If we were to analyze the configurations of linear segments in a sliding window around each pixel in the image, the result would be very time consuming. Because neighboring pixels would share the same line segments, the candidate points can be sampled as a way to describe each relevant region of the image without duplicates. The resulting analyzing window for each candidate point is determined adaptively and is tunable with the sampling rate n_{sp} and a parameter *window_multiplier*. The procedure to extract such sparse points is based on the distances between the linear segments extracted in the pre-processing stage:

Distance map is computed from the pre-processed image via the euclidean distance transform. For each pixel the distance value is denoted D .

Skeletonization: The distance map is thinned, resulting in binary image, where every point on the skeleton is the furthest distance away from the closest segment.

Candidate point sampling is applied to the skeleton image in a row-wise order. For each point in the skeleton a sampling distance Δd is computed as:

$$\Delta d = \frac{2 * window_multiplier * D}{n_{sp}} \quad (4.1)$$

The resulting window size for each sampled candidate point can be further restricted by setting thresholds on the distance value D . Additional structures and regions are filtered from the image prior to line segment detection. Candidate points in those areas are also discarded. The outcome of this process is illustrated in Figure 3.4 in chapter 3.

4.2.2.2 Linear segment detection

The linear segments are detected in a window around each candidate point.

Line detection via the Hough transform is applied to a window of size $D * window_multiplier$ around each candidate point, and is followed by segment linking to concatenate short lines of uniform orientation into longer segments.

Graph construction is performed as described in chapter 3 by considering each detected segment a node in a graph G . Only configurations with at least 3 sides, forming \square – like shapes, are considered valid.

Example of line detection and resulting valid configurations is depicted in Figure 4.9.

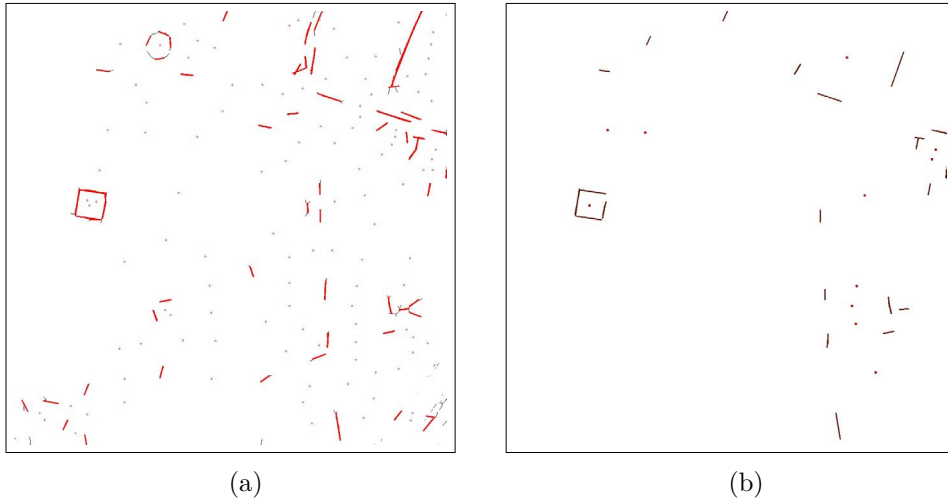


Figure 4.9: RE: (a) detected and linked segments, (b) valid configurations.

4.2.2.3 Configuration ranking

In the last step of the detection phase, each valid configuration around each candidate point is ranked with a rectangularity measure f_R . Only the maximal cliques of each graph G , are kept. The resulting structure size measure f_S and compactness measure f_D are also computed.

The resulting *feature vector* for each valid configuration contains:

- The image coordinates x, y of the candidate point,
- rectangularity measure f_R ,
- structure size measure f_S ,
- compactness measure f_D ,
- *size* - number of segments in the configuration.

The resulting data is saved for later classification.

4.2.3 Classification

Classification is applied to the resulting dataset, when all available site images are processed. For the purposes of evaluation we prepare a *ground truth* variable for each point in our dataset. With a prepared dataset, we can employ supervised learning, to predict a correct class for each datapoint by means of cross-validation. SVM classifier is used for this task.

4.3 Template based detection

Template matching approach has previously been suggested as feasible by Gojda O. et al. in [42] as the variation of crop mark shapes is limited. The most common crop mark shapes in the area were described in work by [7] and are illustrated in Figure 4.10. The most common shapes in our dataset belong to groups A.1.1 and A.1.2.
















































Group	LINEAR FEATURES (lines and enclosures) and MACULAE										
A.1.1.	 A.11t1	 A.11t2	 A.11t3	 A.11t4	 A.11t5	 A.11t6	 A.11t7	 A.11t8	 A.11t9	 A.11t10	 A.11t11
A.1.2.	 A.12t1	 A.12t2	 A.12t3	 A.12t4	 A.12t5	 A.12t6	 A.12t7	 A.12t8	 A.12t9	 A.12t10	 A.12t11
A.1.3.	 A.13t1	 A.13t2									
A.2.1.	 A.211a	 A.211b	 A.212a	 A.212b	 A.213						
A.2.2.	 A.222.1	 A.222.2	 A.2211a	 A.2211b	 A.2212						
B.	 B.11	 B.12.1	 B.12.2								
C.	 C.t1	 C.t2.1	 C.t2.2	 C.2							
I.-III.	 I.1	 II.1	 I.2	 II.2	 III.1	 III.2					

Figure 4.10: Morphology of buried features in non-equivalent scale. Image sourced from [7].

The template based detection utilizes the generalized Hough transform to detect crop marks of approximately circular and rectangular shapes. The algorithm, referred to as GHT, consists of extensive image pre-processing, a detection phase, where a template is matched against the processed image and multiple instances of the object can be retrieved, and a classification step, that helps to filter out falsely detected shapes. The process is illustrated in Figure 4.11.

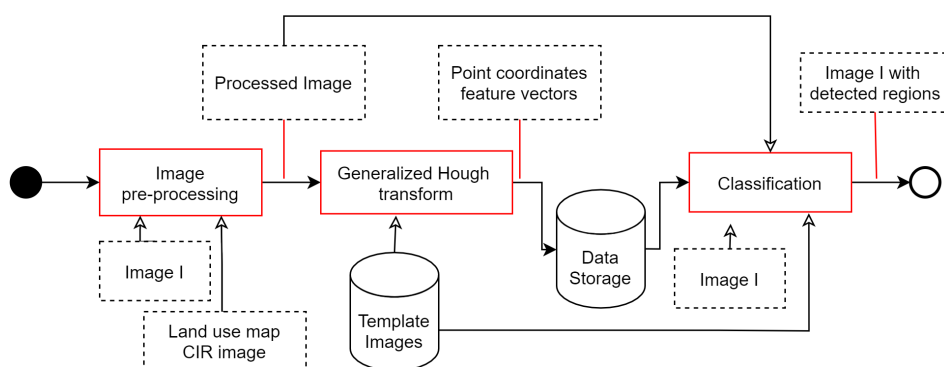
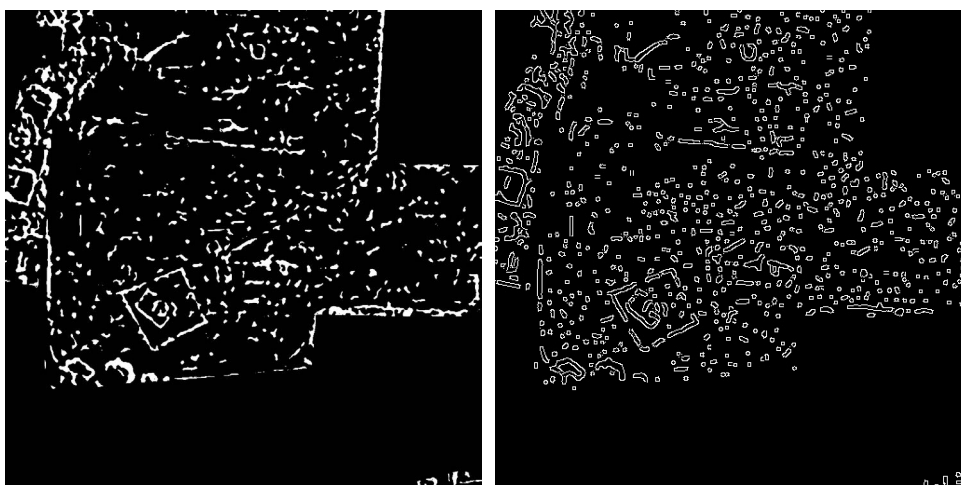


Figure 4.11: General data flow of the GHT algorithm.

4.3.1 Image pre-processing

The image pre-processing consists of several steps. The result is a binary image prepared to be used as input for the generalized Hough transform algorithm. The image is illustrated before and after the application of the Canny edge detector in Figure 4.12.



(a) GHT: Processed image before edge detection. (b) GHT: Processed image after Canny edge detection.

Figure 4.12: GHT: Image pre-processing.

The original RGB image I is converted to grayscale and once again, both the red channel and the first principal component are compared in our experimental section in chapter 6.

If different parameter values are used for each input type, the values are enclosed in Table 4.1. The first step of the pre-processing is contrast enhancement, followed by the filtration of white noise. The white noise is extracted by white top hat operation with square SE of size 4×4 pixels. Median filter is applied to smooth out the artifacts caused by the previous operation.

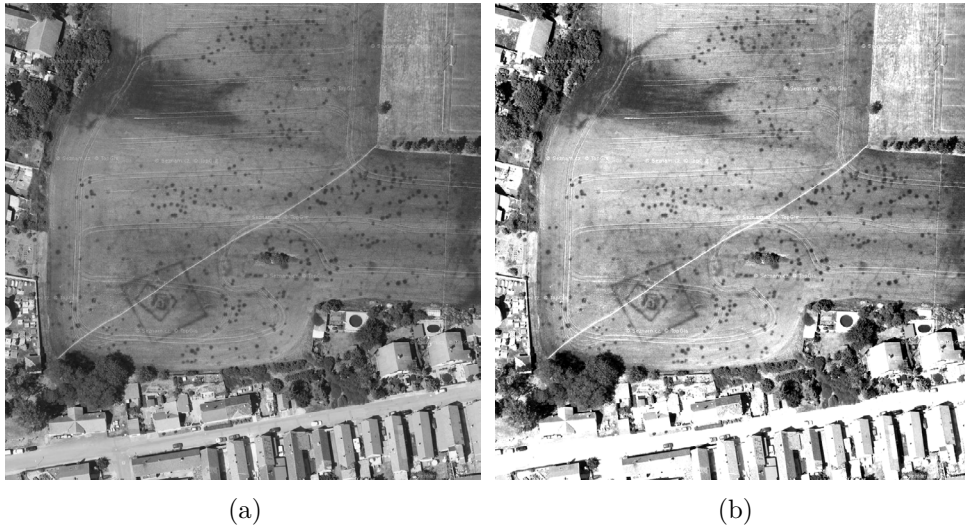


Figure 4.13: GHT: (a) grayscale image, PC1, (b) enhanced contrast.

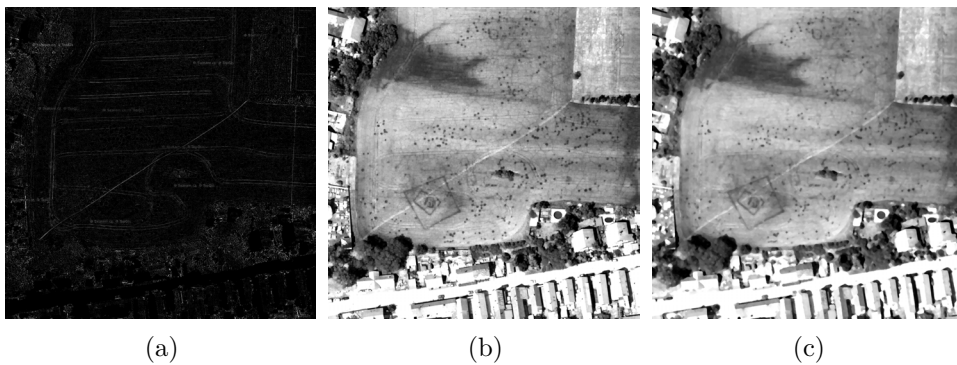


Figure 4.14: GHT: (a) white top hat, (b) white detail subtracted, (c) median filter.

Adaptive threshold of size 37×37 pixels and constant $C = 7$ is applied to the image. The edges are smoothed with one more pass of median filter

	RED	PC1
contrast α	2.2	1.8
contrast β	-150	-50
erosion SE size [px]	3	2

Table 4.1: Different parameter values for RED channel and PC1.

smoothing and the image is inverted. To preserve the original spatial resolution, erosion with square SE of size 3×3 pixels is applied.

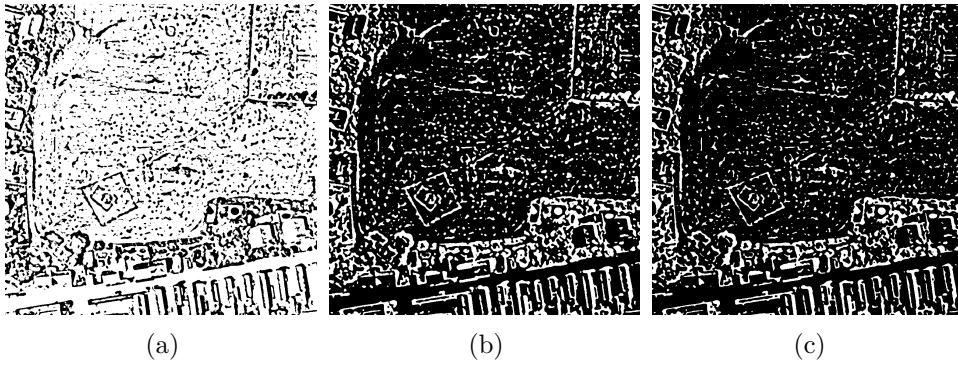


Figure 4.15: GHT: (a) adaptive threshold, (b) median filter, colors inverted, (c) erosion .

Eventually, the image is filtered with a binary mask of land use, so that only agricultural land is retrieved. As can be seen in Figure 4.16, the method is not foolproof as certain areas are marked incorrectly in the map of land use. Some additional noise is removed with information from the CIR imagery and small blob-like features with area smaller than 50 pixels are detected and removed from the image in the final step of pre-processing. This version of the processed image is denoted P . The resulting image is resized to half-size from 1024×1024 to 512×512 pixels. According to our tests this step did not effect the detection rate in a negative way and significantly sped up the process.

Before an image can be processed by the generalized Hough transform, the image P is smoothed by Gaussian blur of size 9×9 pixels and $\sigma = 1.6$. This step smooths out the edges of the wall ridges and the subsequent edge detection results in more bar-like features, which leads to better detection performance. The Canny edge detector is applied to the image, with the following parameters: Sobel operator size is set to 3×3 pixels and the lower and upper hysteresis thresholds are set to 40 and 70.

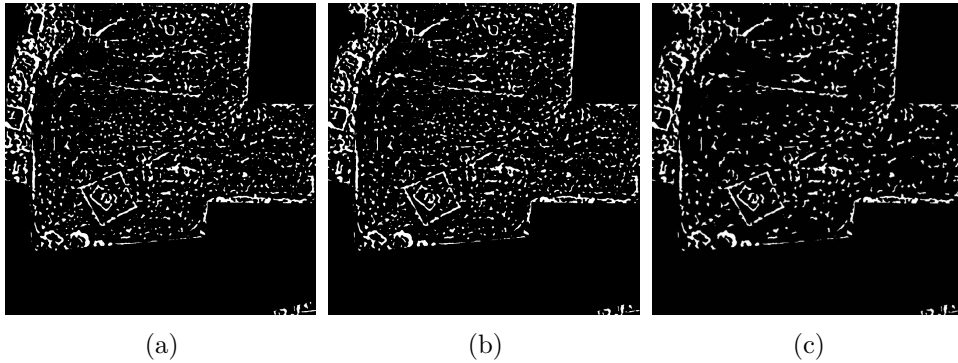


Figure 4.16: GHT: (a) LU map filtration, (b) CIR image filtration, (c) blob filtration.

Rect. sides ratio	Max. width	Max. height	Min. width	Min. height
1.0	260	260	75	75
0.9	260	234	75	67
0.8	260	208	75	60
0.7	260	182	75	52
0.6	260	156	75	45
circle (diameter)	260	–	61	–

Table 4.2: Template sizes in pixels.

4.3.2 Templates

To keep the templates as generalized as possible, we used the perfect rectangular and circular shapes of variable sizes and wall thickness.

The template sizes are disclosed in table Table 4.2. The dimensions are halved accordingly when the processing image is resized. Values enclosed in the table correspond to the original image resolution of 1024×1024 pixels.

We experimented with single edged templates and skeletonized image as is described in more detail in the final chapter, but double edged templates proved to be much more robust when the crop mark shape deviates from the ideal circle or rectangle.

4.3.3 Detection

To save time, the detection was not run for each combination of size and rotation. Instead the circular shape was not rotated at all, and the rectangular template was rotated by a step size $\Delta 3^\circ$ in range $[0^\circ, 90^\circ)$ in case of a square and in interval $[0^\circ, 180^\circ)$ otherwise.

Because the wall thickness does not change in direct proportion to the crop mark size, we propose a solution to resize the template but keep the wall

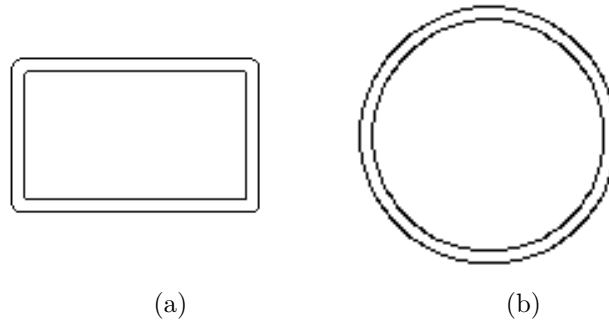


Figure 4.17: Template examples. Dimensions ($w \times h$ or r , thickness) in pixels, colors inverted. (a) Rectangular template (130×78 , 7), (b) Circular template (49 , 5).

thickness relatively unchanged. Creating a new template for each permissible scale would be very time consuming as we could no longer utilize the parallelized version of the generalized Hough transform to its full potential. It would also mean more memory must be reserved to accommodate the extra template files. As a compromise, we run the general Hough Transform in phases, as shown in pseudo-code in algorithm 2 in Appendix D.

A threshold τ needs to be set on the accumulator value, to determine the retrieval rate of the algorithm. The value of this threshold is further discussed in chapter 6. The retrieved matches were saved to a JSON file, for subsequent classification. The following data are saved for each match:

- Image coordinates (x, y) of the center of the template in the input image,
- Hough accumulator value,
- template type (rectangle or circle),
- template scale,
- template rotation,
- exact template dimensions.

4.3.4 Classification

The final process of object recognition is performed by binary classification with a Random Forest classifier. Before the process of additional feature extraction, non-maximum suppression is applied on the resulting set of retrieved ROIs, with the accumulator value used as the deciding factor.

The following features are further extracted for each datapoint:

- *accum_scale*: scale adjusted accumulation value.

- *white*: the number of white pixels in the retrieved region in the processed image P
- *overlap*: the number of pixels in the image P overlapping with the template image with the ridge filled.
- *noise*: the complement to overlap in the region. The number of white pixels in P not covered by the filled in template.
- *overlap_noise*: overlap to noise ratio.
- *noise_scale*: adjusted noise to scale ratio. smaller templates are expected to have higher noise statistics and vice versa.

The ground truth is determined for each datapoint, before the classifier can be trained. The ground truth is determined by a binary mask prepared in advance for each site. If the matched template overlaps the mask from at least 60% the match is considered correct.

Implementation

This chapter describes which tools and libraries were used for realisation of the proposed methods and briefly presents the implemented solutions. The created or modified source files are enclosed to this work and located in folder *src/code* as presented in Appendix C. The remainder of this chapter aspires to introduce the source codes, at least to the extent, to enable possible future attempts at experiment reproduction.

5.1 Development tools and libraries

The majority of this work was implemented in Python, version 3.9.1, using the PyCharm IDE. Most of the since discarded early experiments were conducted in JupyterLab by Project Jupyter¹³, as it offers interactive development environment that allows for fast prototyping. The remainder of this work was implemented in C++, mostly in the form of making modifications to existing source codes. CMake, version 3.20, and Visual Studio 15 2017 were used to compile and build all *.cpp programs.

Available image processing libraries *OpenCV*¹⁴, version 4.0, and *scikit-image*¹⁵ were gladly used in the process of implementation. *Pandas*¹⁶ library has been used to handle datasets in the classification phases of our methods. Finally, the machine learning library *scikit-learn*¹⁷ was used to deploy and train the classification models.

In addition to the mentioned, libraries *NumPy*¹⁸, *SciPy*¹⁹, *Matplotlib*²⁰ and

¹³<https://jupyter.org/>

¹⁴<https://opencv.org/>

¹⁵<https://scikit-image.org/>

¹⁶<https://pandas.pydata.org/>

¹⁷<https://scikit-learn.org/stable/>

¹⁸<https://numpy.org/>

¹⁹<https://www.scipy.org/>

²⁰<https://matplotlib.org/>

*Seaborn*²¹ were used throughout most of the implementation for both basic and high-level mathematical operations, multi-dimensional array operations and data analysis. The Python library *os* is used to read, write or execute files from a script.

For our limited dataset, the storage, needed to handle data between programs in an organized fashion, was realised by utilizing the JSON file format.

5.2 Image pre-processing

Most of the image pre-processing for each method is handled in module *Input*, implemented in script *Input.py* in folder *code/Other*. The script utilizes libraries *NumPy*, *OpenCV*, *scikit-image* and *os*.

5.3 Corner based detection

The algorithms OHC and OHL as well as the Harris corner detectors are implemented in *HarrisCornerDetection.py* and the supplementary file *Line.py*. Image pre-processing and the detection is executed from the file *hc_loader.py*.

5.4 Linear segment based detection

The applicable source files are in the *src/RectangularEnclosures* folder. The folder contains a file *re_loader.py* used to execute the entire process. Candidate point sampling, and segment detection is handled by the class *RECandidatePoints*, graph representation of the detected configurations and graph operations are handled in the *REGraph* class. The classification phase is implemented in the *REClassifier* class. All classes are saved under the corresponding names in the *RectangularEnclosures* folder.

5.5 Template based detection

The applicable source files are in the *code/GHT* folder. The detection phase of the algorithm is implemented in the module *HoughTransform* implemented in *HoughTransform.py*. After the original RGB images are processed by *Input.py*, the detection is run from the file *Matcher.py*.

Initially, for better integration into the project, and also as means to understand the method fully, the generalized Hough transform was implemented from scratch in Python²². As such the solution is included in the source codes

²¹<https://seaborn.pydata.org/>

²²To the author's knowledge a readily available solution does not exist in *scikit-image*, *OpenCV* or any other widely used image processing library.

under the folder *code/GHT*, but unfortunately, even after partial parallelization of the algorithm, the implementation proved to be much too slow to conduct our experiments in an effective way. Therefore, we came back to utilizing the parallelized implementation in C++ written by Matěj Sedláček²³, initially written for the task of tortoise recognition. The implementation must have been modified and extended by several features:

- All input parameters are loaded from a JSON file, prepared by *HoughTransform.py* module.
- Instead of just the best match, the program returns more instances of successful template localization, based on the threshold parameter τ .
- Program saves each successful localization to a JSON file which is further processed by *Loader.ipynb*.

The code is also included in the folder and credited accordingly. The classification phase has been handled in a Jupyter Notebook files *Loader.ipynb* and *Classifier.ipynb*.

5.6 Other enclosed scripts

File *Other/Spectra.py* contains code to convert CIR and RGB photography into false-color composites NDVI and RVI, however as the CIR data acquired from ČÚZK were not taken during a more appropriate season for our task, the scripts ended up unused. Script *Other/WMSClient.py* was used to acquire current and archive orthomaps in RGB and CIR form as well as the maps of land use.

²³sedlacek@utia.cas.cz

Experiments and results

The overall goal of this chapter is to present the experiments conducted with each method on our dataset and to describe the process of evaluation. For each method, the exact parameters for the algorithm are enclosed and subsequently the process of experimental application to our dataset is described. The results are analyzed, after which the limitations of each approach are deduced.

Examples of each experiment outcome are provided only for illustration in the text, as the figures need to be large and are very space-consuming. The full results are provided in folder *data* on the enclosed medium. The final section of this chapter is dedicated to the discussion of the general results presented in this work.

6.1 Corner-based detection methods

The experiments conducted with the two corner-based methods, algorithms LHC and OHC, are described in this section.

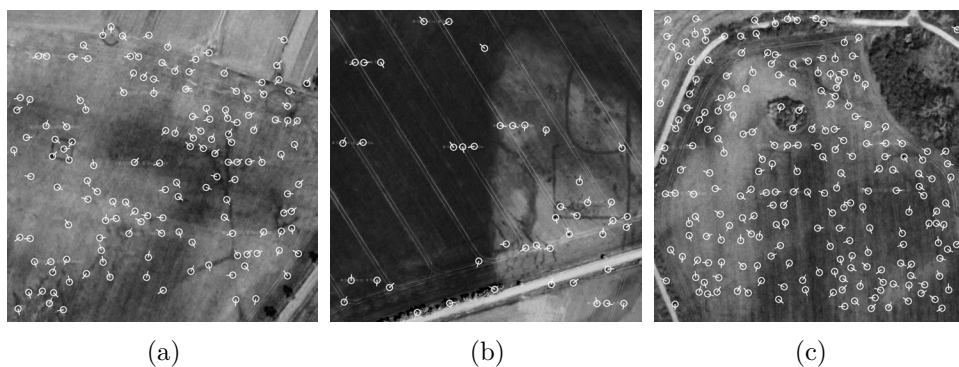


Figure 6.1: Red channel Harris corner detection: (a) Černouček (b) Ledčice (d) Březno^a.

6.1.1 Image pre-processing and corner detection

The algorithms were initially tested on the grayscale images in the forms of the red channel RED and the first principal component PC1 (as were the rest of the methods). Because the results of corner detection on the grayscale images were sub-optimal, as shown in Figure 6.1, the color extended Harris detectors for RGB, opponent color, and ratio color spaces were implemented.

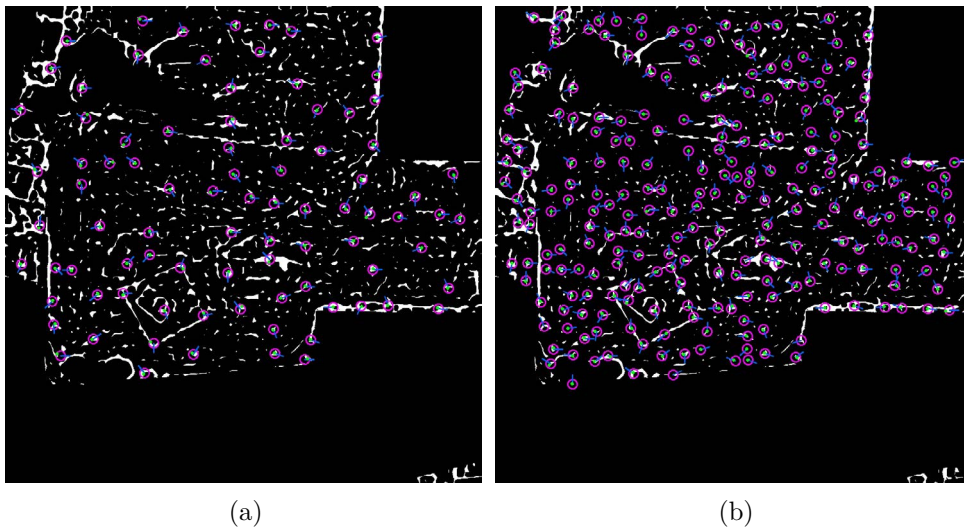


Figure 6.2: Harris corner detector: (a) results for LHC, (b) results for OHC.

The results for the color extended corner detector were generally better, in terms of corner retrieval, than that of the grayscale input, shown in Figure F.1 in Appendix F. Even so, many crop mark corners were still left undetected, especially if a non-zero threshold was applied to the cornerness measure (or measures) in hopes of retaining only the more dominant corners and suppressing the amount of irrelevant points.

In another experiment, we tried to apply the original intensity-based Harris corner detector on the pre-processed images retrieved from the template matching method described in section 4.3 of chapter 4. The motivation behind this decision was that the pre-processed image is in binary form with sharp edges, therefore weak gradient response is not an issue. Additionally, the pre-processing is designed to suppress noise while retaining the original crop mark shape, and thus the total amount of detected corners would be lower, making the subsequent detection faster and possibly more precise. Examples of the resulting corner detection are shown in Figure 6.2.

Apart from the cornerness threshold, a parameter for the non-maximum suppression must be set to define minimum allowed distance between two peaks. Ideally, this parameter would be set to the expected minimum distance value between two corners of a crop mark. In practice, this decision allows for *true* corners to remain undetected, when in close proximity of *false* corner caused by unfiltered noise that yields a higher cornerness measure value. This may not pose such a problem for the LHC algorithm, but the false corners most likely also have an orientation not corresponding with the true corner of a nearby crop mark. In such a case the OHC algorithm is unusable. Upon several experiments with different values of cornerness measure threshold, it was determined the threshold set to 0 returned the best results.

6.1.2 Detection with 4 corners

The detection of crop marks by the LHC algorithm was largely unsuccessful. As was noted before, the detection of all four corners, with a general setting applied to our entire dataset, proved to be a very challenging task. In the end only two or three crop marks were consistently detected by the algorithm with one parameter setting, shown in Figure 6.3. Any attempts made at including more crop marks resulted in large numbers of false positives, as shown in Figure 6.4, concluding the method is very sensitive to noise and parameter tuning. For the best run of the LHC algorithm, as determined by our experiments, the parameters were set to the following values. The minimal and maximal distance between corners was set to 30 and 240 pixels. The maximum allowed deviation from line parallelism was set at 25° and at 15° for deviation from line perpendicularity. At least 60 percent of an edge between two corners must be intact for a line to be considered valid.

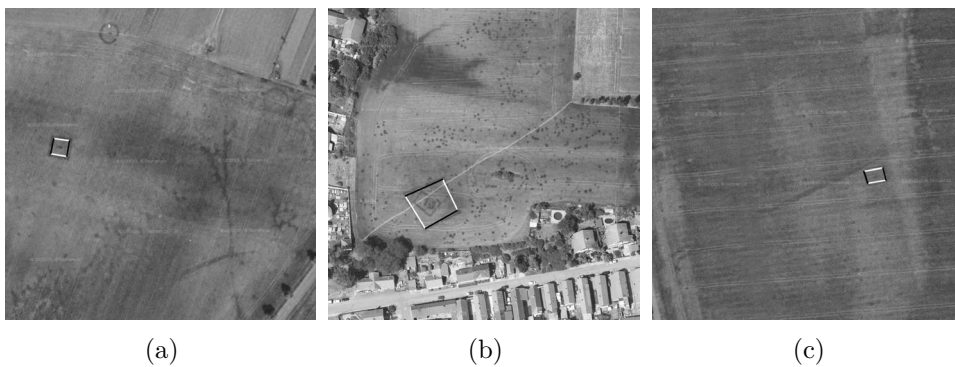


Figure 6.3: LHC: Cases of successful crop mark localization.

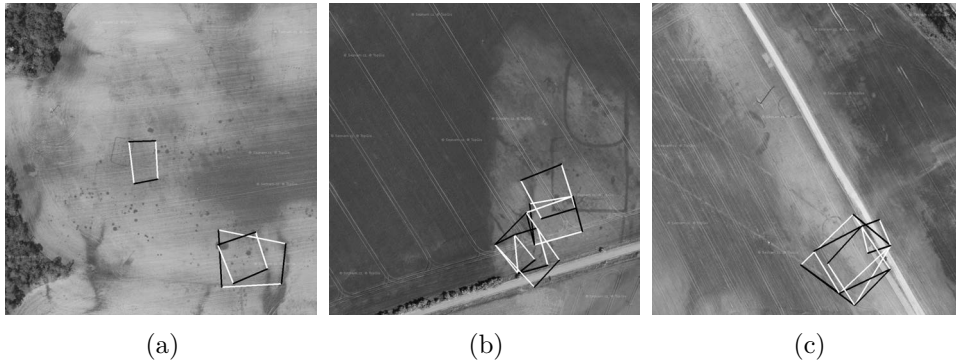


Figure 6.4: LHC: Most attempts ended unsuccessfully.

6.1.3 Detection with 3 corners

Algorithm OHC, which requires only 3 corners, did not improve the detection rate by a large margin, as only 3 or 4 crop marks were consistently retrieved, as shown in Figure 6.5. The advantage that was gained by allowing one of four corners to be missed by the detection phase was, as per our experiments, effectively offset by the additional corner orientation constraint. As can be seen in Figure 6.6, even when the corresponding corner is detected, it is not guaranteed the original shape has not already been degraded in the original site image to such an extent that the detected corner orientation is skewed. Examples of problematic sites are shown in Figure 6.7.

The minimal and maximal distance between corners was set to 30 and 270 pixels. The maximum allowed deviation of corner orientation from orientation of the presumed diagonal was set at 30° , as it is understood that more elongated rectangles will deviate more. The maximum allowed deviation for line perpendicularity was set at 20° . At least 85 percent of an edge between two corners must be intact. From our testings, the constraint placed on the completeness of the two adjacent edges was very effective in the reduction of false positives.

6.2 Linear segment based detection

Out of the three proposed methods, the RE algorithm was the least experimented with, as it was almost completely applicable to our task in its original form. The following section describes the parameter settings that were used to achieve our best results with the method.

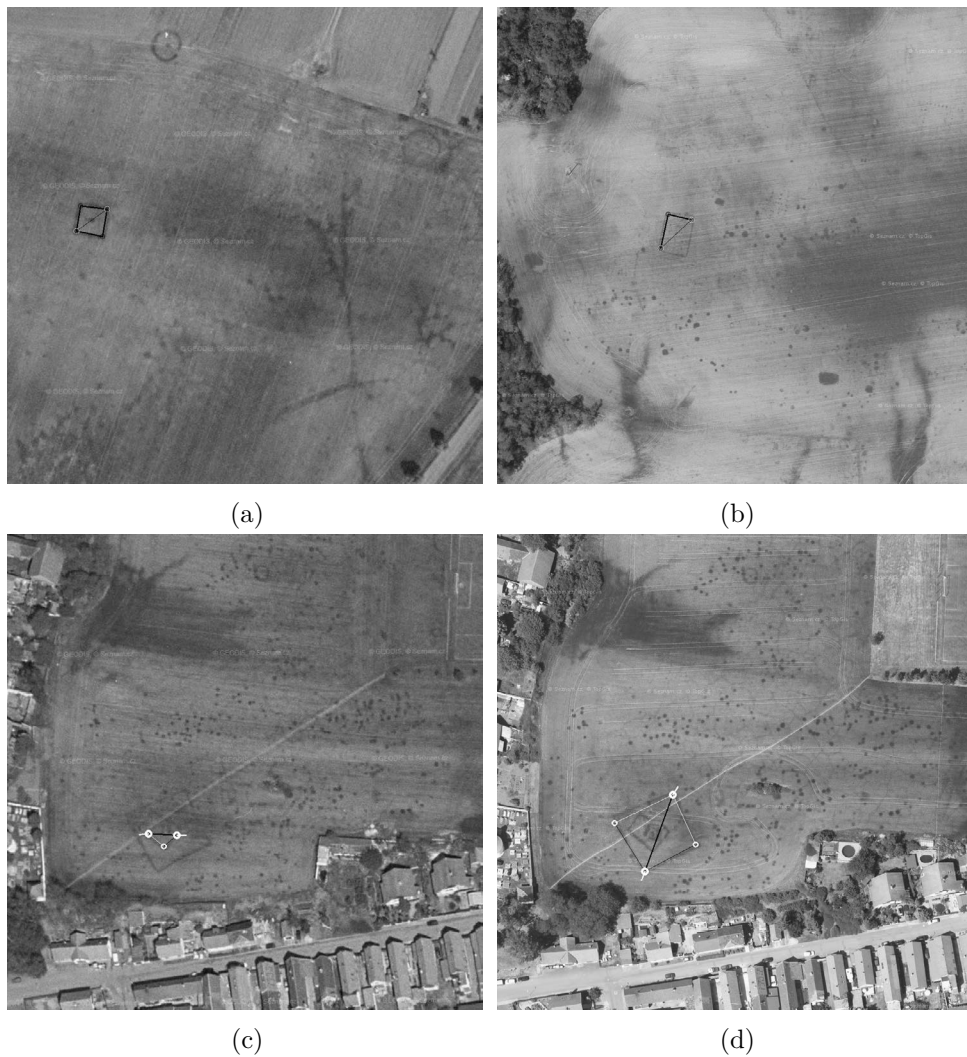


Figure 6.5: OHC: Cases of successful crop mark localization.

6.2.1 Image pre-processing and valid configuration detection

The parameters as described in this section were experimentally selected by manual evaluation to maximize the number of detected crop marks. The following parameters were set during the pre-processing stage. The white top hat square SE size was set to 3×3 pixels. The following black top hat was applied with square SE of size 15×15 pixels. The pixels with intensity lower than 20 are suppressed to 0 with thresholding. The sizes of the morphological closing square SEs were eventually set to 1 pixel for dilation, effectively leaving the image unchanged, and 3×3 pixels for erosion. The MFC operator is applied with structuring elements of size 3×3 and 10×10 pixels. The linear segments were enhanced by the composite opening with linear SE set to 30

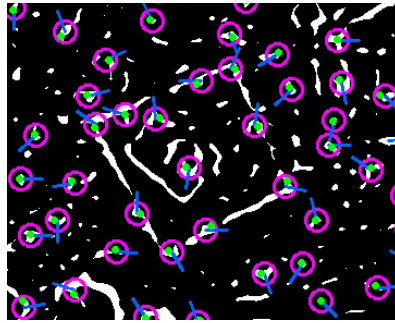


Figure 6.6: OHC: Corner orientation can be wrongly determined due to fragmentation in the original colour image (lower-left corner, orientation indicated by blue line).

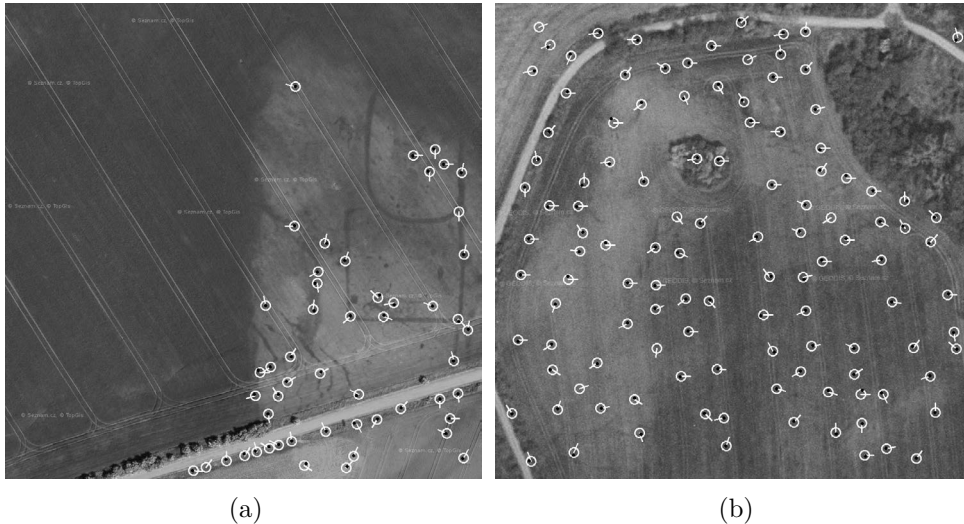


Figure 6.7: OHC: Many sites were missed entirely due to undetected corners or wrongly determined corner orientation. In image (a) 3 corners are successfully detected, left upper corner remains undetected, while right upper corner's configuration is not suitable for the OHC algorithm. In image (b) upper left corner is missed by the corner detector and lower right corner coincides with the surroundings and is filtered out during the pre-processing, remaining corners and their orientations are detected correctly.

pixels and rotated in increments of 5° . The proposed thresholding after the segment enhancing algorithm is not applied, thus the threshold is set to 0.

Following the pre-processing stage, the candidate points were sampled with the following parameters: The *window_multiplier* parameter is set to 1.8 with the sampling rate parameter n_{sp} set to 5. All candidate points having a distance smaller than 10 or greater than 170 pixels were discarded, which

limits the distances between opposing walls of the detected structure.

The segment detection was conducted with the Hough transform, and all lines with minimum length of 20 pixels were detected. The often fragmented segments were linked together if the gap between two endpoints was less than 30 pixels and the angles differed by less than 10 degrees.

The angle constraint $\beta_{k,j}$ was set to 15° and the convexity measure threshold applied to $\tau_{k,j}$ was set to 0.4.

After the application of the detection on the red channel image RED and the first principal component composite PC1, it was determined the algorithm returned better results for the red channel. Initially, as shown in Figure 6.8, the pre-processing applied to the RED image tended to detect less linear ridges in the line segment enhancement step described in algorithm 1, which in turn leads to less segments that are potentially detrimental to the next step in the algorithm – the candidate point sampling which is strongly dependent on the successful suppression of noise.

After the following parameters were changed for the pre-processing of PC1 image, the results became comparable to those of the red channel, shown also in Figure 6.8. The white top hat square SE size was changed from 3 pixels to 4, the dilation SE size in the morphological closing operation was also increased from one to two pixels and new threshold on segment enhancement was set to 60, instead of 0.

The number of retrieved points for each input is shown in Table 6.1.

	Retrieved	N_{FP}	N_{TP}
PC1	439	421	18
RED	293	271	22

Table 6.1: RE: Retrieval rates.

Compactness measure Many of the detected configurations consisted of small segments placed far apart. Because many of such configurations were attributed to the remains of irrelevant structures such as plough furrows, we implemented the compactness measure f_D defined in Equation 3.27 in hopes of better discrimination of these sparse configurations from true crop marks, that tended to be more compact. Examples of the ranked configurations are shown in Figure 6.9, where the intensity of the red channel intensity of each valid configuration indicating square is set to $200f_D$.

Overall, we were able to retrieve 7 unique rectangular crop marks with the PC1 dataset and 6 using the red channel.

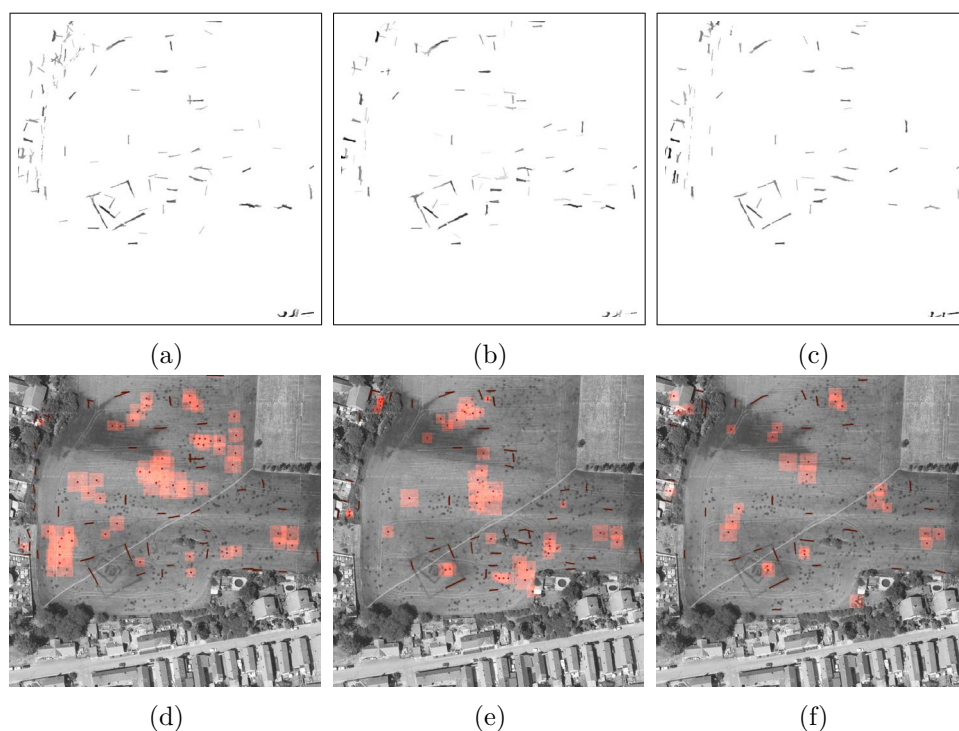


Figure 6.8: RE: site Ctiněves^a PC1 (a) and RED (b) after segment enhancement. PC1 (d) and RED (e) after valid configuration detection. PC1 (c) after segment enhancement and (f) valid configuration detection with new parameters. Size of rectangle around each candidate point indicates size of rectangularity measure f_R .

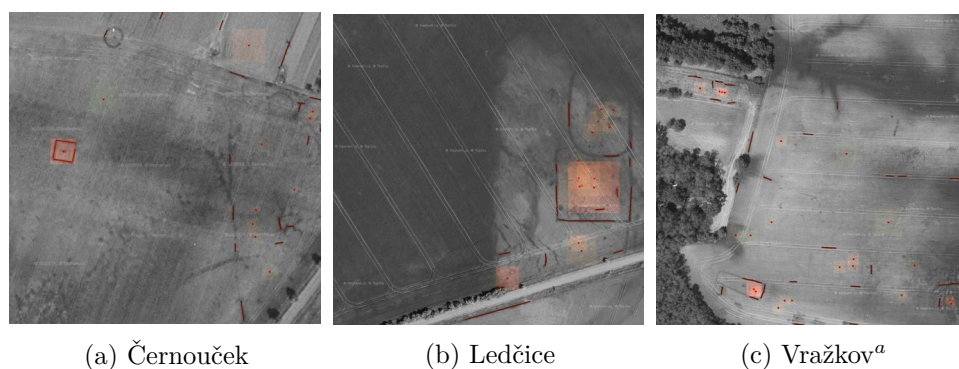


Figure 6.9: RE: Examples of ranked configurations, PC1 input.

6.2.2 Classification

Before the classification, the ground truth was determined for each data point. Because of how limited is our dataset, the training and evaluation of the SVM

model was conducted as leave-one-out cross-validation. The regularisation parameter C of the classifier was set to 1.0 and the default RBF kernel was used. The model validation on our small dataset did not result in any statistically significant metrics for reporting. Similar to the work of Zingman et al. in [6], the method was evaluated based on the feature’s ability to discriminate crop marks from irrelevant structures. We conducted this experiment by setting the parameter *class_weight* on the classifier. Therefore, we set the parameter *class_weight* to such a high value for the true class, that none of our retrieved true positives were misclassified, and focused on the number of remaining false positives N_{FP} . N_{FP} and N_{TP} are, respectively, the numbers of false positives and true positives.

As can be seen from Table 6.2, the classification helped to significantly reduce the false positive rate and it can be expected that with the inclusion of additional data, the results would be further improved. Furthermore, the compactness measure f_D proposed in this work improved the classification results, to those with only the original two features, as the rate of false positives was significantly reduced.

dataset	class_weight	(f_R, f_S, f_D)		(f_R, f_S)	
		N_{FP}	N_{TP}	N_{FP}	N_{TP}
PC1	0.99	16	18	30	18
PC1	0.98	1	16	15	15
PC1	0.95	0	10	4	13
RED	0.99	37	22	50	22
RED	0.98	29	22	42	22
RED	0.95	9	22	30	22
RED	0.90	9	22	23	22
RED	0.85	7	21	22	22

Table 6.2: RE: Retrieval rates with all features included.

The experimental results of the classification are presented in Figure F.2, Figure F.3 and Figure F.4 for the PC1 input and in Figure F.5, Figure F.6 and Figure F.7 for the RED input in Appendix F.

6.3 Template-based detection

The experiments conducted with the template-based detection method are described in this chapter.

6.3.1 Image pre-processing

In this section, we describe what led to the final form of pre-processing of grayscale images in the forms of the red channel RED and the first principal

component PC1. The complete process of the image pre-processing, including parameters, was already described in chapter 4. Therefore, this chapter focuses on several experiments that were conducted before arriving at the final solution.

The process was initially influenced by the pre-processing applied in the work by Zingman et al. [38] and used in the RE algorithm. In first stages, the generalized Hough transform was applied to the ridge image acquired from RE and processed by the Canny edge detector. The problem of this approach was two-fold. First, that the pre-processing used in RE detection heavily alters the original wall thickness. Secondly, the pre-processing understandably does not enhance circular shapes. It became obvious that the same level of abstraction, applied in the RE algorithm, will not work in the GHT algorithm. Therefore more gentle approach focused on very gradual changes to the image was developed. The best sequence we have found is already described in section 4.3 of chapter 4, however, few other approaches were also considered, as is outlined below.

Skeletonization We briefly experimented with application of the GHT algorithm to image that resulted from the skeletonization of the image P . The idea behind the step was to minimize the amount of noise. While most residual noise in the image was in the form of blobs or fragmented speckles and resulted in tiny dots and lines after the skeletonization, the crop mark skeletonization resulted in comparably long lines. Unfortunately, the double ridge, which we found to be a valuable feature, was lost during this process. After the application of the detection algorithm, it became clear that the results will be inferior to the initial process. The skeletonized image P is shown as example (b) in Figure 6.10.

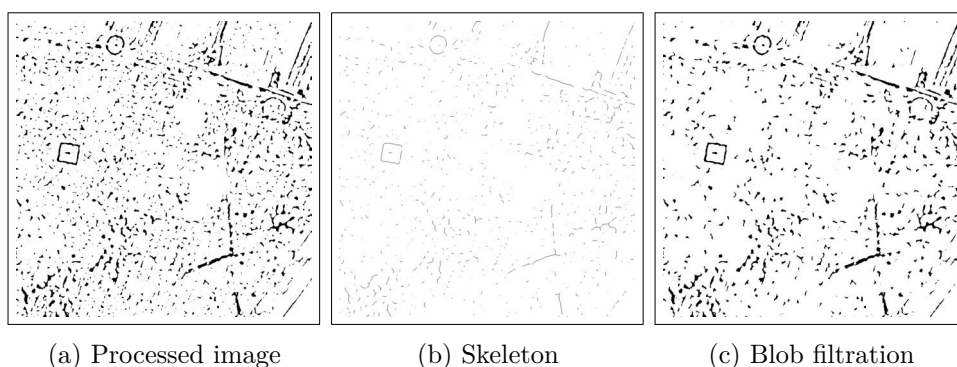


Figure 6.10: Experiments with image pre-processing, colors inverted.

Blob filtration The blob filtration described as the last step of the pre-processing was the most recent addition to the GHT algorithm. It did not significantly change the generalized Hough transform retrieval rate, but slightly improved the final detection result, as many of the features used in the classification are derived from the pre-processed image P .

VARI The VARI index from Equation 2.1 in chapter 3 was not eventually used. While the results seemed interesting at first, the ratio yielded less stable results during the pre-processing phase. The VARI ratio retained some crop marks better than the red channel, but it performed worse when it came to filtration of lines caused by furrow ploughs and in many cases missed the entire patch containing a crop mark. The comparison of RED, PC1 and the VARI images is presented in Figure 6.11.

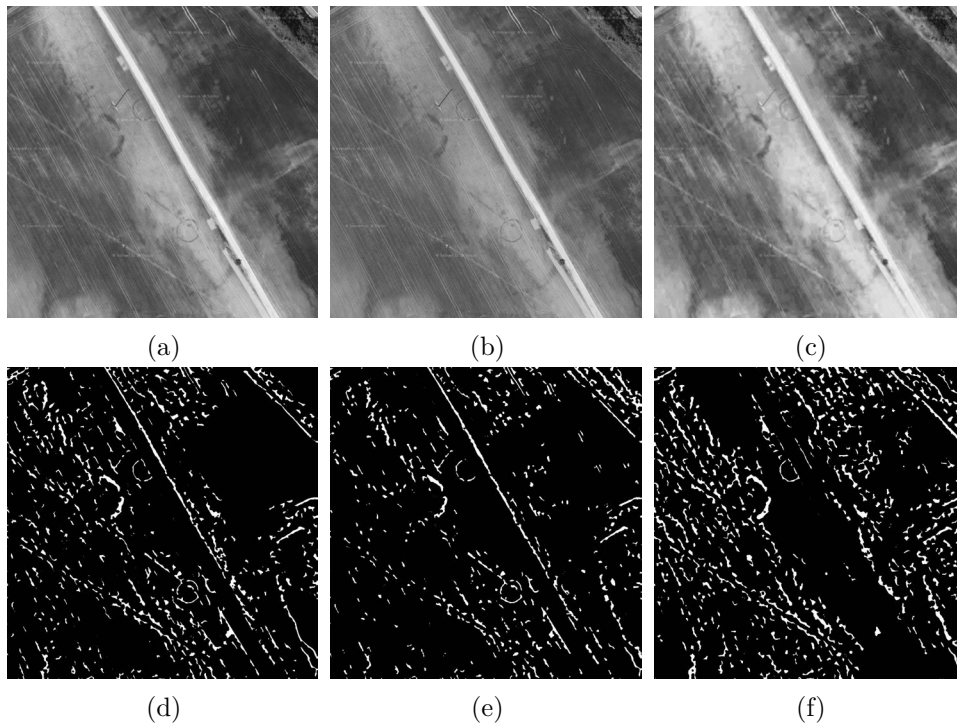


Figure 6.11: Red channel (a,d), PC1 (b,e) and VARI (c,f) before and after pre-processing.

6.3.2 Templates

At first, the algorithm was tested with a template of near-exact copy of each crop mark shape, to make sure the algorithm is correctly implemented. Even-

tually, the template sizes and shapes became more and more generalized. As one of the future goals of our research is to automatically detect yet undiscovered crop marks, the successful detection with a generalized templates in both shape and sizing is desirable. Therefore, we finally decided to use only templates of double-ridged rectangles and circles.

The wall ratios of the rectangles was initially very incremental, but eventually we realized that the high number of ratios resulted in many duplicates in regions already retrieved with a different template. For the template sizes we used, the ratios in the set $\{0.6, 0.7, 0.8, 0.9, 1.0\}$ were sufficient. Were the spatial resolution of the crop marks larger in relation to the image size, we would probably have to employ a higher number of sizes and shapes.

The size of the ridge, or the wall thickness, depends on the type of pre-processing. Eventually, we settled on the value of 6 pixels (in the original image size). As explained in chapter 4, this ridge changes only slightly with the changing template size, depending on the number of phases. In our experiments we used 4 phases for rectangular templates and 3 phases for circular templates. The total number of scales between the maximum and minimum template dimensions was set to 40 for both template types.

6.3.3 Singlepass vs. region of interest detection

The algorithm in its final form runs the generalized Hough transform on the entire input image. Because after filtration many parts of the image are effectively empty, we developed a method that would first extract regions containing compact structures. Those regions would then be used as the input to the generalized Hough transform.

The motivation behind this idea is two-fold. First, very large structures (such as the borders where two fields meet) would be filtered out during this process and would not be subject to the detection, possibly reducing the false alarm rate. Second, the resulting detection would be significantly faster, because for each detected ROI, much smaller than the entirety of the original image, we could only use templates of the corresponding sizes.

To extract the compact structures, dilation operator with round SE the size of 15 pixels was applied to the processed image P to connect fragmented structures. Dominant contours were detected in the image and enclosed in a bounding rectangle. Rectangles larger than 500 pixels, smaller than 50 pixels or with side ratio smaller than 0.4 were filtered out. The resulting rectangles were used to filter out ROIs from the original processed image before the dilation.

Since the process of ROI detection did not work well on heavily fragmented crop marks, as it would divide them into two or more separate ROIs, we decided not to pursue this approach further. Example is shown in Figure 6.12. The algorithm in its final form runs the generalized Hough transform on the entire input image.

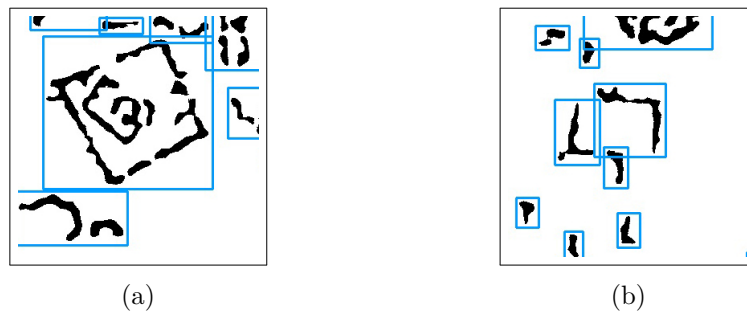


Figure 6.12: (a) correctly filtered crop mark region, Ctiněves^a (b) fragmented crop mark is divided into multiple regions, Březno^a

6.3.4 Hough accumulator threshold value

The threshold value was strongly dependent on the amount of filtration we conducted on the input image. For higher value of the threshold the Hough transform could miss some fragmented or misshapen crop marks. For too low value the algorithm would lose precision and retrieve too many false positives. This is known as the precision-recall trade-off. In our case, the goal is to retrieve as much crop marks as possible, while allowing for a higher rate of false positives to be retrieved and later filtered out in the classification stage. Eventually, we settled on the value $\tau = 1.1$ for rectangular template shapes and blob filtered images for both PC1 and RED input. For circular shapes $\tau = 1.0$ proved to be a good value for circular crop marks that were almost fully intact, like the two in site Černouček. The threshold needed to be lowered to 0.8 to retrieve fragmented shapes like the one in Vražkov^b.

6.3.5 Classification

The following section focuses on the discrimination of crop marks of archaeological significance from mistakenly retrieved locations. Employing the leave-one-out cross-validation process with the total of 14 sites, the Random Forest classifier was trained with the following experimentally found parameters:

- The number of decision trees in the ensemble: $n_estimators = 50$
- Maximum depth of each decision tree: $m_depth = 3$
- Positive class importance: $class_weight = 0.98$

Initially, the model was trained on data retrieved by both rectangular and circular templates. However, the model was better trained with each template type individually, so the template type feature became obsolete. Presented below are the results for each type of template.

Rectangular templates The Table 6.3 contains the numbers of matches retained by the generalized Hough transform followed by non-maximal suppression. The last column indicates how many sites that contain positive examples were retrieved, ideally we would want all sites with rectangular crop marks retrieved. While the processed red channel without the additional blob filtration retained the most positive examples, the quite random classification results showed, that the retained matches were not specific enough to successfully train the model.

Model	Retrieved	N_{FP}	N_{TP}	Sites retrieved
PC1 blob filtered $\tau = 1.1$	140	128	12	9
RED blob filtered $\tau = 1.1$	101	87	14	9
PC1 blob filtered $\tau = 1.2$	26	19	7	6
RED blob filtered $\tau = 1.2$	28	19	9	7
PC1 unfiltered $\tau = 1.1$	103	190	13	8
RED unfiltered $\tau = 1.1$	248	227	21	10

Table 6.3: Rectangular templates: Retrieval rates for the Hough transform.

Circular templates As for the previous case the best results overall were acquired with the PC1 blob filtered dataset.

Summary The best results in this work were reached with the pre-processing described in chapter 4 with the additional blob filtration. The PC1 proved to be a better resource than the red channel. The best case of crop mark detection before and after object recognition with Random Forest is shown

Model	Retrieved	N_{FP}	N_{TP}	Sites retrieved
PC1 blob filtered $\tau = 0.8$	77	59	18	9
RED blob filtered $\tau = 0.8$	65	42	23	10
PC1 blob filtered $\tau = 0.9$	41	29	12	8
RED blob filtered $\tau = 0.9$	36	18	18	10
PC1 blob filtered $\tau = 1.0$	17	9	8	7
RED blob filtered $\tau = 1.0$	16	6	10	7
PC1 unfiltered $\tau = 0.8$	84	61	23	10
RED unfiltered $\tau = 0.8$	54	33	21	10
PC1 unfiltered $\tau = 0.9$	39	25	14	8
RED unfiltered $\tau = 0.9$	27	13	14	10
PC1 unfiltered $\tau = 1.0$	19	9	10	7
RED unfiltered $\tau = 1.0$	9	1	8	5

Table 6.4: Circular templates: Retrieval rates for the Hough transform.

in its entirety in Figure F.8, Figure F.9, Figure F.10, Figure F.11 and Figure F.12 for rectangular templates and Figure F.13, Figure F.14, Figure F.15, Figure F.16 and Figure F.17 in Appendix F, for circular templates. We managed to successfully detect 7 rectangular crop marks with the GHT algorithm applied with rectangular templates. Using circular templates the algorithm successfully detected 6 circular shapes, 7 if crop mark in site Březno^b could be considered a circular shape, in addition to few rectangular crop marks.

6.4 Discussion

Throughout the process the thesis completion, the lack of available data was a known obstacle. For that reason, the commonly used methods of systematic measurement and statistical evaluation are not applicable. In the following text the advantages and limitations of the implemented methods are discussed.

6.4.1 Summary

The most obvious limitation of the corner based methods is the fact, that well defined corners are sparse in our examples of crop marks, but are otherwise well defined around irrelevant structures such as trees or plough rails. Only when much of the additional noise was filtered out, by the same pre-processing sequence used in the template based method, were we able to significantly lower the amount of retrieved false positives. Even then, the method did not return very good results, as only the most clearly defined crop marks were detected. Overall, the OHC algorithm performed slightly better than LHC, since it was easier to detect just 3 corners per crop mark, however improvements still need to be made in the way the corner orientation is determined. Overall, the corner based methods are evaluated as unsuitable for the task of crop mark detection from orthomaps based on our testings.

For the RE algorithm, the introduced compactness measure has shown good performance discriminating crop marks from other structures in addition to the original rectangularity and structure size measures. While there is surely still room for improvement in the classification stage of the method, the main limiting factor of the algorithm is its detection phase. Unlike the results reported from the author of the original method, we encountered significantly more noise in the processed data, than the original paper [6] suggested. The reason behind it might be that the spatial resolution of our data is much higher than that of the satellite imagery used in the original. Additionally, arable land already contains many linear structures by design, be it borders between two neighboring fields or rails caused by industrial farming methods.

The RE algorithm is, unlike the third proposed method, difficult to generalize for other shapes than those of imperfect rectangles. In the future, it may be co-opted for the detection of shapes illustrated in group A2.2. in Figure 4.10, however it's usability for the crop mark detection is otherwise limited.

The GHT algorithm is by a large margin the most successful method proposed in this work. The classifier shows good performance on data retrieved with a rectangular template and higher accumulator threshold value.

The Hough transform with a circular template, retrieved a significant amount of rectangular shaped crop marks, which can be mostly attributed to the lower threshold used on the accumulator value, and requires additional experiments to be conducted.

In the future, it could be worth to experiment with a greater variety of template shapes and higher threshold values. While this approach would be very computationally expensive, it would allow us to attempt the detection of more uniquely shaped marks, such as are the evenly distributed dots caused by stake pits visible in site Straškov^a or the bastion shaped remnants in the unused site Hořín, as well as significantly fragmented shapes as are the undetected circles in sites Straškov^b and Cítov.

Unlike the previous methods, the GHT algorithm is not limited to rectangular shapes but can be applied to any predefined shape. Our experiments suggest that the lack of prior knowledge of the exact shape and size of a detected crop mark can be to a certain extent mitigated with a large number of slightly varied templates, nonetheless, it comes at a price in terms of additional computation.

6.4.2 Future improvements

The work that was done within the boundaries of this thesis serves more as a stepping stone for future research than as a standalone and complete topic. From the proposed methods, only one yielded results promising enough, to be built and improved upon. However, the experiments were conducted on an extremely small dataset, with a very limited amount of negative examples. In future work, the author would plan to extend the dataset to the full area around Říp, without site cherry-picking, so that the algorithm can be properly tested and evaluated on a sufficient amount of data, which was something that could not be achieved in such a short amount of time. In the meantime, further development of the implementation process is desirable, as the current state was created for fast prototyping but not processing of large amounts of data. Improvements can be also made in further exploration of the effects of different ratios of spectral bands on crop mark detection, as this work offers only a very brief introduction into the problematic. Were we able to access CIR images from a more suitable vegetation season, it is possible the initial image pre-processing would yield better results.

The initial aspiration that inspired this thesis was the possibility of conducting multi-modal detection on multiple sources of data. As we could not access enough quality data for this task, the thesis' scope was significantly limited. However, because of the popularity of land surveillance and open data is on the rise, and the gradually decreasing price of high-resolution spec-

tral and radar sensors, it may be only a question of a couple of years until more suitable data becomes available. If it will come to that, we intend to utilize the research and experience gained through the work conducted on this project.

Until then, the problem with a lack of data may be possibly solved with data augmentation, which could lead to the possibility of deep learning. This option was not sufficiently explored in this work and will be the next step in the author's work.

6.4.3 Contribution of this work

In conclusion, this work may serve as a proof of concept for the fact that crop marks have the potential to be automatically detected from publicly available aerial images. Because we were very limited by the number of positive examples available to us, more systematic methods of data pre-processing, object recognition, and evaluation were traded for a very experiential approach to the task.

Conclusion

In this thesis, we provided a brief overview of the state-of-the-art methods of remote sensing and automatic detection in relation to aerial archaeology. We conducted research into the publicly available data sources and used according to our research the most suitable data to construct a dataset. Methods of image processing, feature extraction, and classification considered for our task were reviewed and three approaches were designed and implemented. We experimentally tested each algorithm on our dataset and comparatively evaluated their results. The limitations of methods were described and further discussed. The implemented GHT algorithm showed good performance in the detection of the crop marks in our limited dataset and as such is evaluated as the most promising of the presented methods for future improvement. While we can conclude that crop marks have the potential to be automatically detected from publicly available aerial images, further assessment with a larger dataset is necessary.

Bibliography

- [1] Royo, C.; Villegas, D. *Field Measurements of Canopy Spectra for Biomass Assessment of Small-Grain Cereals*. 09 2011, ISBN 978-953-307-492-4, doi:10.5772/17745.
- [2] Roman, A.; Ursu, T. *Multispectral satellite imagery and airborne laser scanning techniques for the detection of archaeological vegetation marks*. 12 2016, ISBN 978-606-543-787-6, pp. 141–152.
- [3] Bolstad, P. *GIS fundamentals : a first text on geographic information systems / Paul Bolstad, University of Minnesota - Twin Cities*. Ann Arbor, MI: XanEdu, 6th edition, 2019, ISBN 9781593995522.
- [4] Sýkora, D. Digital Image Processing - Lecture 5: Non-Linear Filtering [university lecture]. 2021. Available from: <https://dcgi.fel.cvut.cz/home/sykorad/dzo/slides/dzo-105.pdf>
- [5] Sonka, M.; Hlavac, V.; et al. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, fourth edition, 2007, ISBN 049508252X.
- [6] Zingman, I.; Saupe, D.; et al. Detection of Fragmented Rectangular Enclosures in Very High Resolution Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, volume 54, no. 8, 2016: pp. 4580–4593, doi:10.1109/TGRS.2016.2545919.
- [7] Gojda, M.; Hejcman, M. Cropmarks in main field crops enable the identification of a wide spectrum of buried features on archaeological sites in Central Europe. *Journal of Archaeological Science*, volume 39, no. 6, 2012: pp. 1655–1664, ISSN 0305-4403, doi:<https://doi.org/10.1016/j.jas.2012.01.023>. Available from: <https://www.sciencedirect.com/science/article/pii/S0305440312000350>
- [8] Agapiou, A.; Lysandrou, V.; et al. Study of the Variations of Archaeological Marks at Neolithic Site of Lucera, Italy Using High-Resolution Multi-

- spectral Datasets. *Remote Sensing*, volume 8, no. 9, 2016, ISSN 2072-4292, doi:10.3390/rs8090723. Available from: <https://www.mdpi.com/2072-4292/8/9/723>
- [9] Gojda, M. *Archeologie a dálkový průzkum*. Academia, 2017, ISBN 978-80-200-2644-6.
- [10] Starková, L. Toward a High-Definition Remote Sensing Approach to the Study of Deserted Medieval Cities in the Near East. *Geosciences*, volume 10, no. 9, 2020, ISSN 2076-3263, doi:10.3390/geosciences10090369. Available from: <https://www.mdpi.com/2076-3263/10/9/369>
- [11] Lasaponara, R.; Leucci, G.; et al. Towards an operative use of remote sensing for exploring the past using satellite data: The case study of Hierapolis (Turkey). *Remote Sensing of Environment*, volume 174, 2016: pp. 148 – 164, ISSN 0034-4257, doi:<https://doi.org/10.1016/j.rse.2015.12.016>. Available from: <http://www.sciencedirect.com/science/article/pii/S0034425715302376>
- [12] Calleja, J. F.; Requejo Pagés, O.; et al. Detection of buried archaeological remains with the combined use of satellite multispectral data and UAV data. *International Journal of Applied Earth Observation and Geoinformation*, volume 73, 2018: pp. 555–573, ISSN 0303-2434, doi:<https://doi.org/10.1016/j.jag.2018.07.023>. Available from: <https://www.sciencedirect.com/science/article/pii/S0303243418304410>
- [13] Faltynova, M.; Pavelka, K. Aerial Laser Scanning in Archeology. *Geoinformatics FCE CTU*, volume 6, 12 2011, doi:10.14311/gi.6.14.
- [14] Luo, L.; Wang, X.; et al. Automated Extraction of the Archaeological Tops of Qanat Shafts from VHR Imagery in Google Earth. *Remote Sensing*, volume 6, 12 2014: pp. 11956–11976, doi:10.3390/rs61211956.
- [15] Agapiou, A.; Lysandrou, V. Remote sensing archaeology: Tracking and mapping evolution in European scientific literature from 1999 to 2015. *Journal of Archaeological Science: Reports*, volume 4, 2015: pp. 192–200, ISSN 2352-409X, doi:<https://doi.org/10.1016/j.jasrep.2015.09.010>. Available from: <https://www.sciencedirect.com/science/article/pii/S2352409X15301139>
- [16] Lu, P.; Yang, R.; et al. On the use of historical archive of aerial photographs for the discovery and interpretation of ancient hidden linear cultural relics in the alluvial plain of eastern Henan, China. *Journal of Cultural Heritage*, volume 23, 2017: pp. 20 – 27, ISSN 1296-2074, doi: <https://doi.org/10.1016/j.culher.2015.09.010>, beyond the modern landscape: Earth Observation to see the unseen. Available from: <http://www.sciencedirect.com/science/article/pii/S1296207416300681>

-
- [17] Kazimi, B.; Thiemann, F.; et al. Deep Learning for Archaeological Object Detection in Airborne Laser Scanning Data. 09 2018.
- [18] Landauer, H. R., J. Machine learning for large area archaeological feature detection: Applying transfer learning to airborne lidar data. 2019. Available from: <https://www.chnt.at/wp-content/uploads/Machine-learning-for-large-area-archaeological-feature-detection.pdf>
- [19] Stott, D.; Kristiansen, S. M.; et al. Searching for Viking Age Fortresses with Automatic Landscape Classification and Feature Detection. *Remote Sensing*, volume 11, no. 16, 2019, ISSN 2072-4292, doi:10.3390/rs11161881. Available from: <https://www.mdpi.com/2072-4292/11/16/1881>
- [20] Liu, W.; Zhang, Z.; et al. Road Detection by Using a Generalized Hough Transform. *Remote Sensing*, volume 9, no. 6, 2017, ISSN 2072-4292, doi:10.3390/rs9060590. Available from: <https://www.mdpi.com/2072-4292/9/6/590>
- [21] San, D.; Seçkin, M. Building extraction from high resolution satellite images using Hough transform. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, volume 38, 01 2010.
- [22] Baumann, P. Aerial Photography: History and Georeferencing. *The Geographic Information Science Technology Body of Knowledge*, 2019, doi:10.22224/gistbok/2019.2.5. Available from: <https://gistbok.ucgis.org/bok-topics/aerial-photography-history-and-georeferencing>
- [23] Verhoeven, G. Near-Infrared Aerial Crop Mark Archaeology: From its Historical Use to Current Digital Implementations. *Journal of Archaeological Method and Theory*, volume 19, 04 2012: pp. 132–160, doi:10.1007/s10816-011-9104-5.
- [24] Chase, A.; Chase, D.; et al. Airborne LiDAR, archaeology, and the ancient Maya landscape at Caracol, Belize. *Journal of Archaeological Science*, volume 38, 02 2011: pp. 387–398, doi:10.1016/j.jas.2010.09.018.
- [25] Third edition, 1997. Available from: <http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>
- [26] cuzk@cuzk.cz, C. w. e.-m. Souřadnicové systémy. May 2018. Available from: [https://geoportal.cuzk.cz/\(S\(ti0untk5wnsl3rbxwvxmr1tc\)\)/Default.aspx?mode=TextMeta&side=sit.trans&text=souradsystem](https://geoportal.cuzk.cz/(S(ti0untk5wnsl3rbxwvxmr1tc))/Default.aspx?mode=TextMeta&side=sit.trans&text=souradsystem)

- [27] Deng, G.; Cahill, L. An adaptive Gaussian filter for noise reduction and edge detection. 01 1993, ISBN 0-7803-1487-5, pp. 1615 – 1619 vol.3, doi: 10.1109/NSSMIC.1993.373563.
- [28] Sharifi, M.; Fathy, M.; et al. A classified and comparative study of edge detection algorithms. In *Proceedings. International Conference on Information Technology: Coding and Computing*, 2002, pp. 117–120, doi: 10.1109/ITCC.2002.1000371.
- [29] Sýkora, D. Digital Image Processing - Lecture 3: Convolution [university lecture]. 2021. Available from: <https://dcgi.fel.cvut.cz/home/sykorad/dzo/slides/dzo-103.pdf>
- [30] Yu, T.-H.; Woodford, O.; et al. A performance evaluation of volumetric interest point detectors. *Int. J. Computer Vision*, volume 102, 2013: pp. 180–197.
- [31] Sirisha, B.; Sandhya, B. Evaluation of distinctive color features from harris corner key points. In *2013 3rd IEEE International Advance Computing Conference (IACC)*, 2013, pp. 1287–1292, doi:10.1109/IAdCC.2013.6514414.
- [32] Bala, K. Harris Corner Detection [university lecture]. 2015. Available from: http://www.cs.cornell.edu/courses/cs4670/2015sp/lectures/lec07_harris_web.pdf
- [33] Sebe, N.; Gevers, T.; et al. Evaluation of intensity and color corner detectors for affine invariant salient regions. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, IEEE, 2006, pp. 18–18.
- [34] Bouman, C. A. Opponent Color Spaces [university lecture]. 2021. Available from: <https://engineering.purdue.edu/~bouman/ece637/notes/pdf/Opponent.pdf>
- [35] Hough, P. V. Method and Means for Recognizing Complex Patterns. 12 1962.
- [36] Duda, R. O.; Hart, P. E. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM*, volume 15, no. 1, Jan. 1972: p. 11–15, ISSN 0001-0782, doi:10.1145/361237.361242. Available from: <https://doi.org/10.1145/361237.361242>
- [37] Ballard, D. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, volume 13, no. 2, 1981: pp. 111–122, ISSN 0031-3203, doi:[https://doi.org/10.1016/0031-3203\(81\)90009-1](https://doi.org/10.1016/0031-3203(81)90009-1). Available from: <https://www.sciencedirect.com/science/article/pii/0031320381900091>

- [38] Zingman, I.; Saupe, D.; et al. Automated search for livestock enclosures of rectangular shape in remotely sensed imagery. 10 2013, doi:10.1117/12.2027704.
- [39] Zingman, I.; Saupe, D.; et al. A morphological approach for distinguishing texture and individual features in images. *Pattern Recognition Letters*, volume 47, 04 2014: pp. 129–138, doi:10.1016/j.patrec.2014.03.019.
- [40] Vapnik, V.; Chervonenkis, A. *Theory of Pattern Recognition [in Russian]*. Moscow: Nauka, 1974, (German Translation: W. Wapnik & A. Tscherwonkenis, *Theorie der Zeichenerkennung*, Akademie-Verlag, Berlin, 1979).
- [41] Cutler, A.; Cutler, D.; et al. *Random Forests*, volume 45. 01 2011, ISBN 978-1-4419-9325-0, pp. 157–176, doi:10.1007/978-1-4419-9326-7_5.
- [42] Gojda, O.; Hanzlik, P.; et al. Study on processing and georeferencing of historical features. *International Journal of Mathematics and Computers in Simulation*, volume 9, 01 2015: pp. 35–43.

Basic Notation and Conventions

I grayscale image in function or matrix form

I_c multispectral image in function or matrix form

I_x, I_y derivatives of image I in x, y directions

x, y image coordinates

τ threshold

σ standard deviation

\mathbb{R} real numbers

Σ covariance matrix

Ψ Morphological transformation

X^C complement of a set X

$A \setminus B$ set of A elements which are not in B

$A \cup B$ union of sets A and B

$S(r, \theta, l)$ linear segment of length l , orientation θ and distance r from candidate point

G graph

G^c clique in graph G

$E(G)$ set of all edges in graph G

$V(G)$ set of all nodes in graph G

$e(u, v)$ edge between nodes $u, v \in V(G)$

$\| \cdot \|$ norm of a matrix

Acronyms

The following list contains explanation of used abbreviations.

ALS Airborne laser scanning

CIR Color infra-red image

ČÚZK The Czech Office for Surveying, Mapping and Cadastre

DTM Digital terrain model

LiDAR Light detection and ranging

GHT Generalized Hough transform, refers to one of the developed methods

HC Harris corner detector

MFC Morphological feature contrast

NIR Near-infra red

OHC 3-corner based detection algorithm

LHC 4-corner based detection algorithm

RE Rectangular enclosures detection algorithm

RGB Red, green and blue image channels

ROI Region of interest

SE Structuring element

TIN Triangulated irregular network

UAV Unmanned aerial vehicle

WMS Web map service

Contents of enclosed flash disk

	readme.txt.....	the file with disk contents description
	data.....	the directory with image data
	src.....	the directory of source codes
	code.....	implementation sources
	thesis.....	the directory of \LaTeX source codes of the thesis
	text.....	the thesis text directory
	thesis.pdf.....	the thesis text in PDF format
	thesis.ps.....	the thesis text in PS format

Algorithms

Algorithm 2: GHT: New template is generated for each phase.

Input:

- Processed image I ,
- number of phases ϕ ,
- max and min template dimensions $(a, b), (a_m, b_m)$,
- list of permissible template rotations \mathbf{r} .
- wall thickness t .

Output: Matrix X of feature vectors. $i \leftarrow 0$ $coef \leftarrow (\frac{a-a_m}{\phi}, \frac{b-b_m}{\phi})$ $X \leftarrow \emptyset$ **while** $i < \phi$ **do** $temp_dim \leftarrow (a, b) - i * coef$ $template \leftarrow createTemplate(temp_dim, t)$ $i \leftarrow i + 1$ $temp_dim_{min} \leftarrow (a, b) - i * coef$ $X.append(runGHT(I, template, \mathbf{r}, temp_dim_{min}))$ **return** X

Algorithm 3: Corner based detection: LHC algorithm

Input:

- Image I ,
- pre-processed image P contains edges between corners,
- *corners* - list of detected, each corner defined by coordinates and orientation as (x,y) ,
- *min_length*, *max_length* - interval of valid line lengths,
- δ_r - allowed deviation for line parallelism,
- δ_p - allowed deviation for line perpendicularity.

Output: Image I with rectangles superimposed in locations of detected crop marks.

$valid_lines \leftarrow \emptyset$, $parallel_lines \leftarrow \emptyset$

foreach $C_1, C_2 \in corners$ **do**

if $dist(C_1, C_2) \in [min_length, max_length]$

and $edge(C_1, C_2)$ exists in P **then**

$valid_lines \leftarrow (C_1, C_2)$

foreach A, B **in** $valid_lines$ **do**

if $|angle(A, B)| < \delta_r$ **or** $|angle(A, B)| - 180^\circ < \delta_r$ **then**

$parallel_lines \leftarrow (A, B)$

for (A, B) **in** $parallel_lines$ **do**

for (C, D) **in** $parallel_lines \setminus (A, B)$ **do**

if lines share all corners **then**

if $90^\circ - \delta_p < |angle(A', B')| < 90^\circ + \delta_p$ **foreach**

$A' \in (A, B), B' \in (C, D)$ **then**

$I \leftarrow rectangle(A, B, C, D)$

return I

Algorithm 4: Corner based detection: OHC algorithm

Input:

- Image I ,
- pre-processed image P contains edges between corners,
- *corners* - list of detected, each corner defined by coordinates and orientation as (x,y,o) ,
- *min_length*, *max_length* - interval of valid line lengths,
- δ_p - allowed deviation for line perpendicularity,
- δ_d - allowed deviation for corners on diagonal.

Output: Image I with triangles superimposed in locations of detected crop marks.

valid_lines $\leftarrow \emptyset$, *perpendicular_lines* $\leftarrow \emptyset$

foreach $C_1, C_2 \in \textit{corners}$ **do**

if $\textit{dist}(C_1, C_2) \in [\textit{min_length}, \textit{max_length}]$
and $90^\circ - \delta_p < |C_1.o - C_2.o| < 90^\circ + \delta_p$
and *edge exists in P* **then**
 \perp *valid_lines* $\leftarrow (C_1, C_2)$

foreach $(C_1, C_2), (C_2, C_3)$ **in** *valid_lines* **do**

if $180^\circ - \delta_d < |C_1.o - C_3.o| < 180^\circ + \delta_d$ **then**
 \perp $I \leftarrow ((C_1, C_2), (C_2, C_3))$

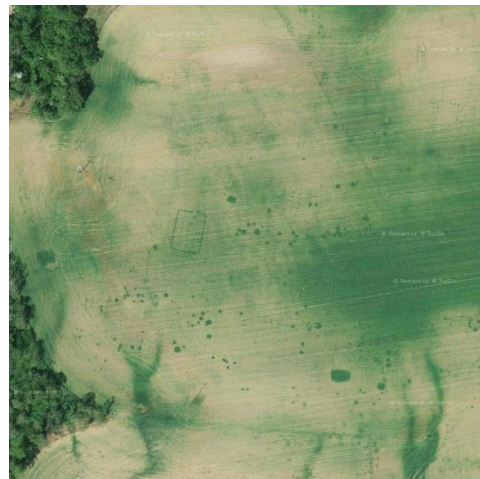
return I

APPENDIX **E**

Sites



(a) Černouček



(b) Straškov^a



(c) Ctiněves^a

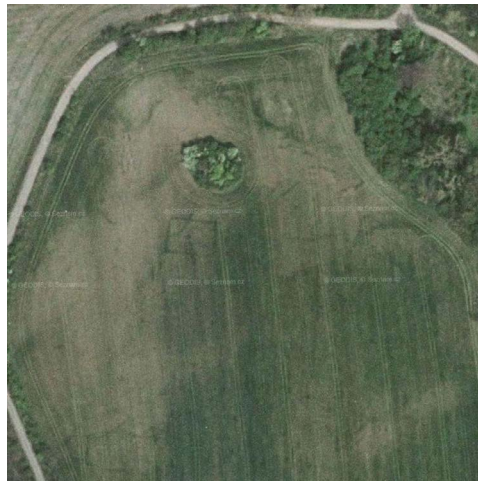


(d) Ctiněves^b

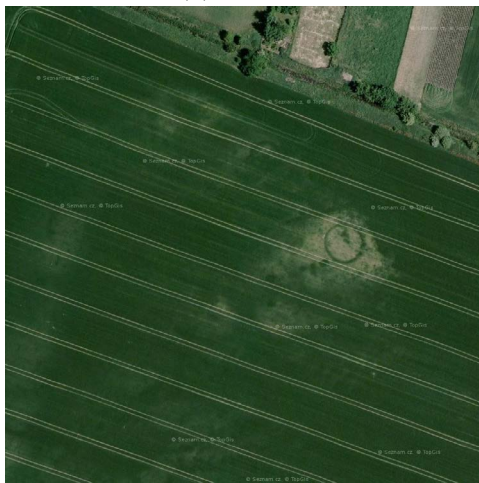
Figure E.1: Positive examples: part 1/3



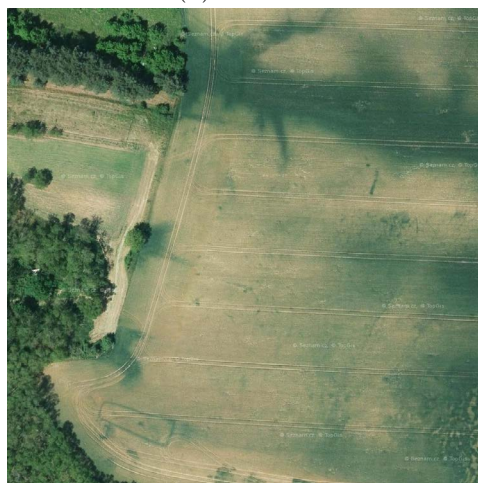
(a) Ledčice



(b) Březno^a

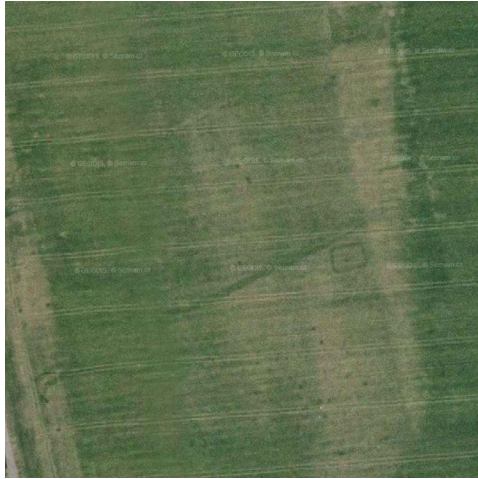


(c) Cítov

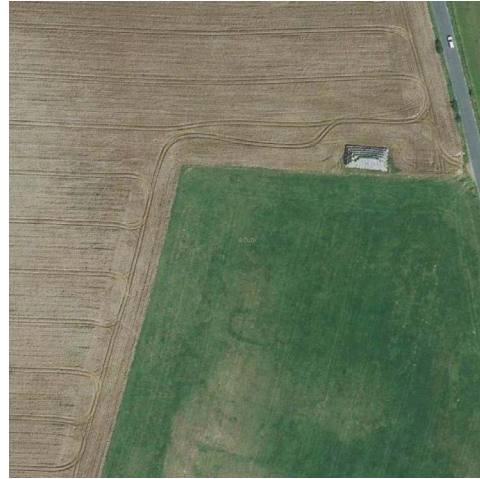


(d) Vražkov^a

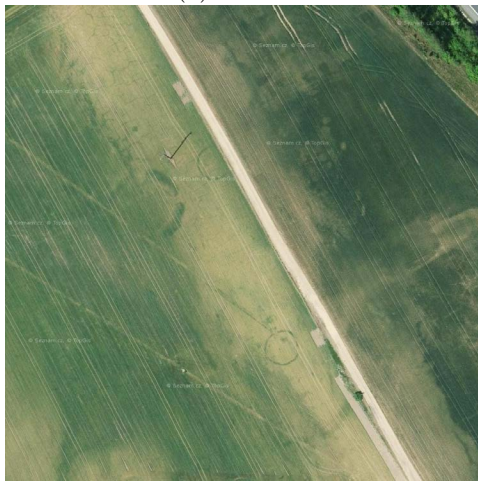
Figure E.2: Positive examples: part 2/3



(a) Březno^b



(b) Vražkov^b



(c) Straškov^b

Figure E.3: Positive examples: part 3/3

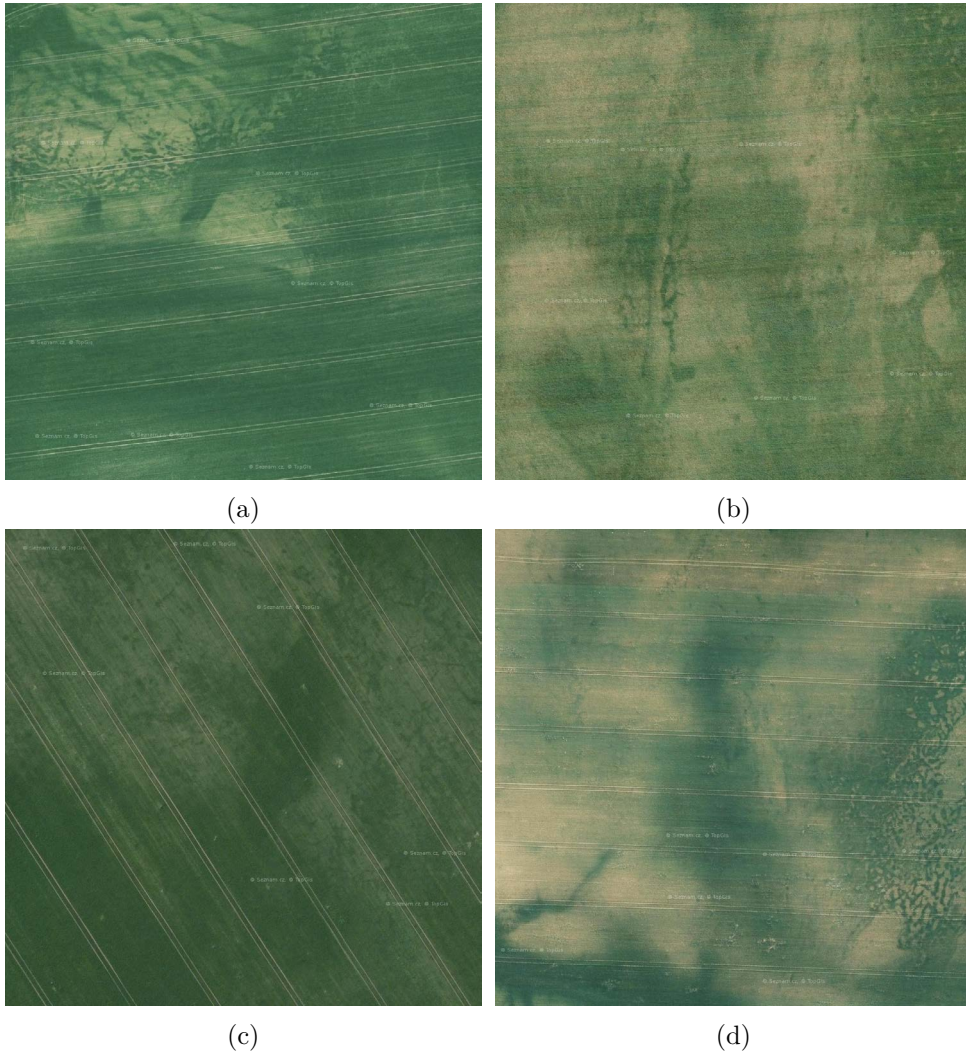
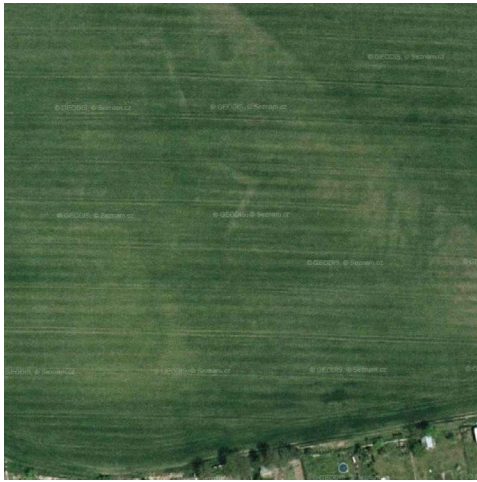


Figure E.4: Negative examples.

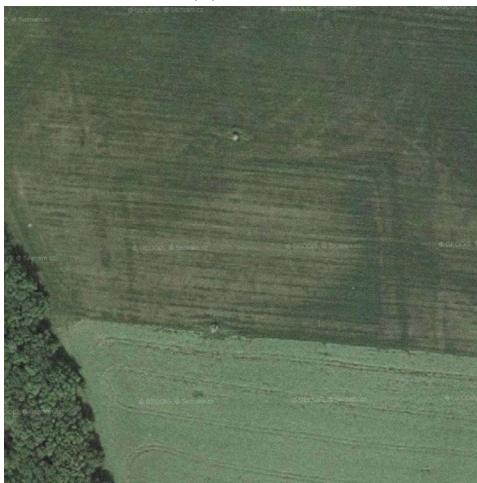
E. SITES



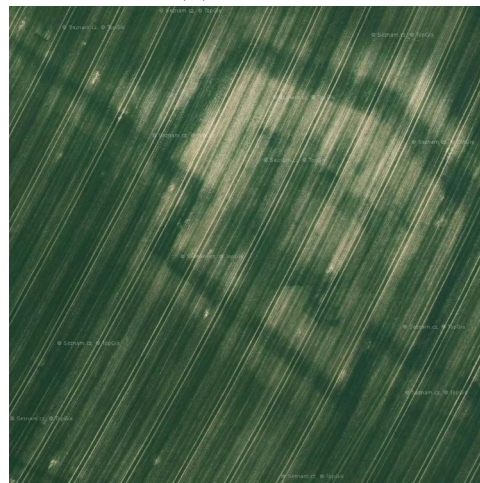
(a) Hořín



(b) Nechanice



(c) Rakovice



(d) Třeboutice

Figure E.5: Unused sites.

Experiments

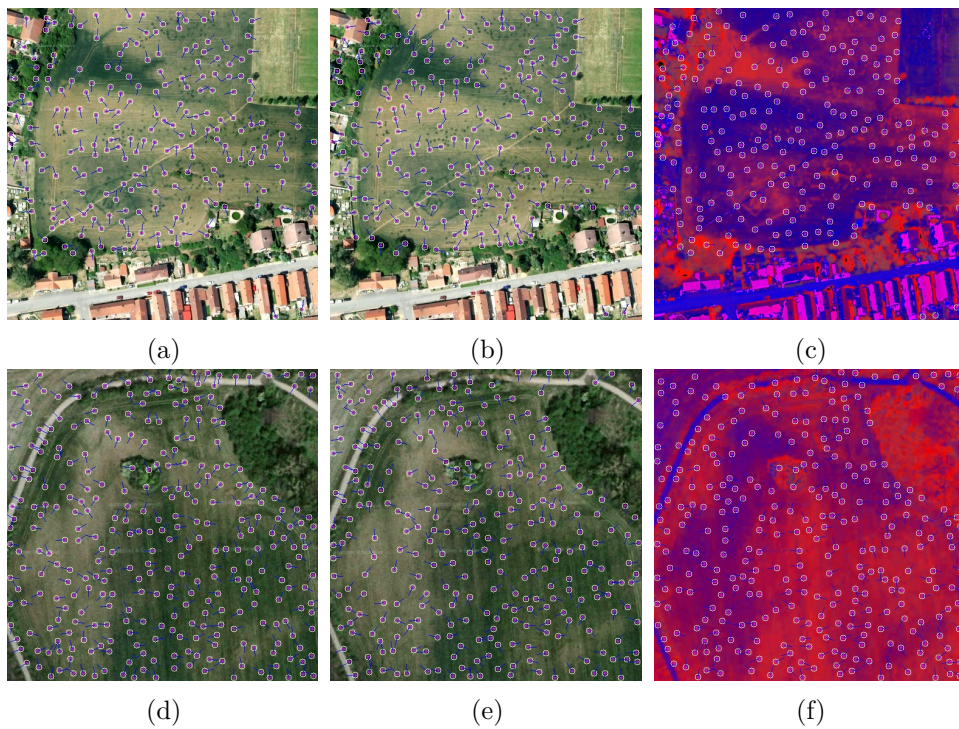


Figure F.1: Color extended Harris corner detector: (a,d) RGB, (b,e) ratio color space (corners projected onto RGB image), (c,f) opponent color space. Threshold for corner selection is set to 0.

F. EXPERIMENTS



Figure F.2: RE classification results: PC1 input with `class_weight` parameter set to (a) .99 (b) .98 (c) .95, Part 1/3

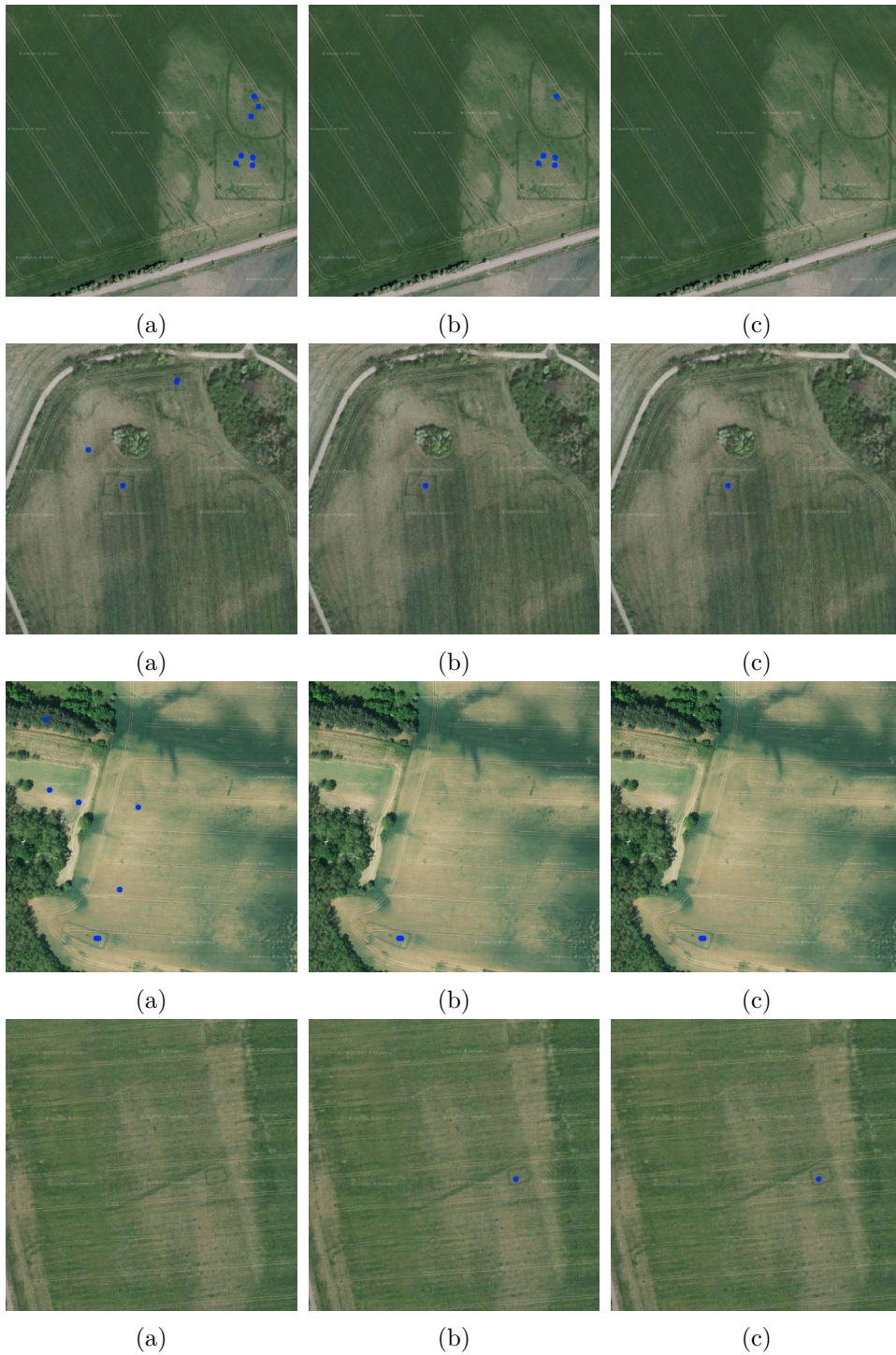


Figure F.3: RE classification results: PC1 input with class_weight parameter set to (a) .99 (b) .98 (c) .95, Part 2/3

F. EXPERIMENTS

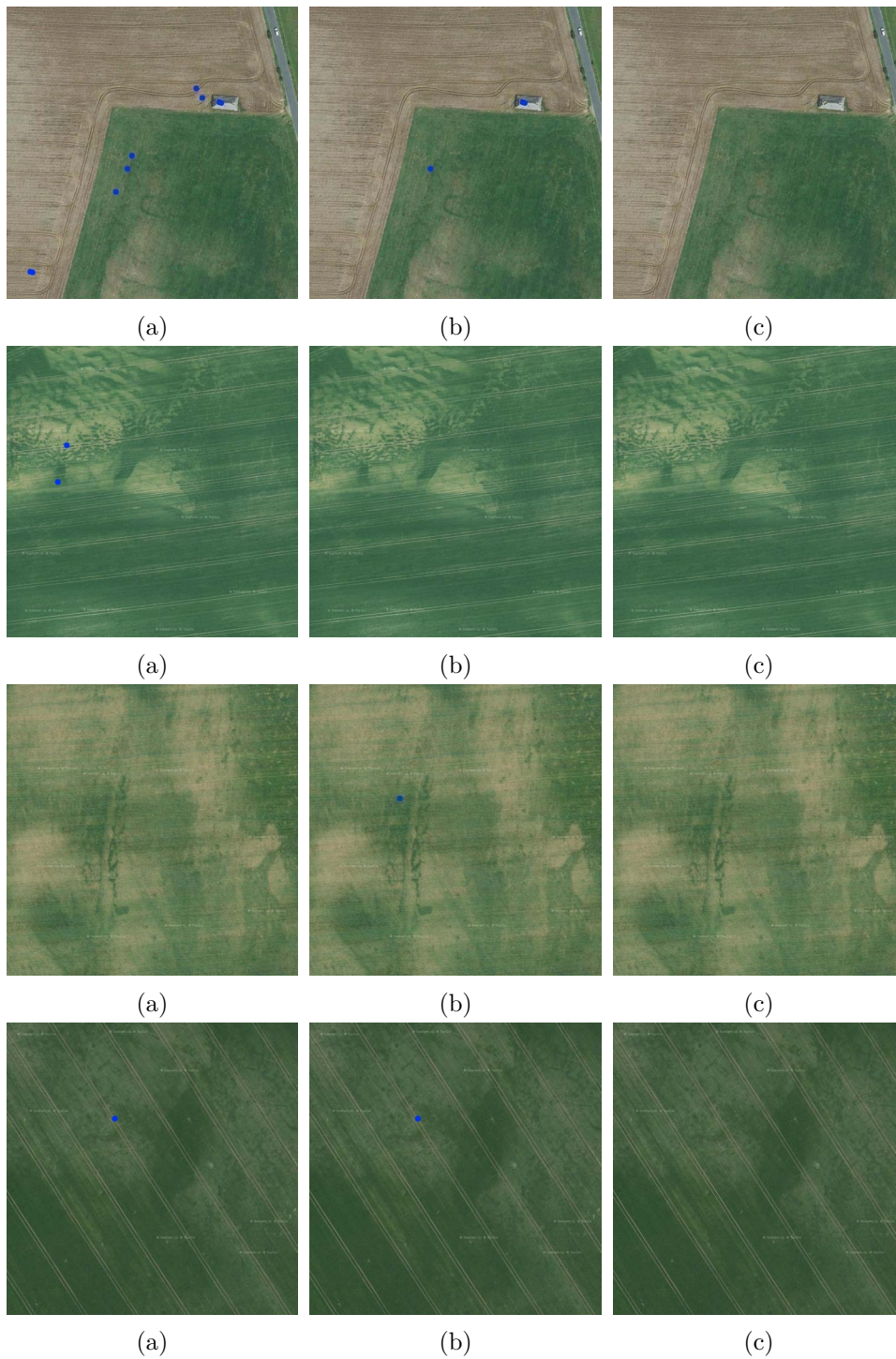


Figure F.4: RE classification results: PC1 input with class_weight parameter set to (a) .99 (b) .98 (c) .95, Part 3/3



Figure F.5: RE classification results: RED input with class_weight parameter set to (a) .99 (b) .98 (c) .90, Part 1/3

F. EXPERIMENTS

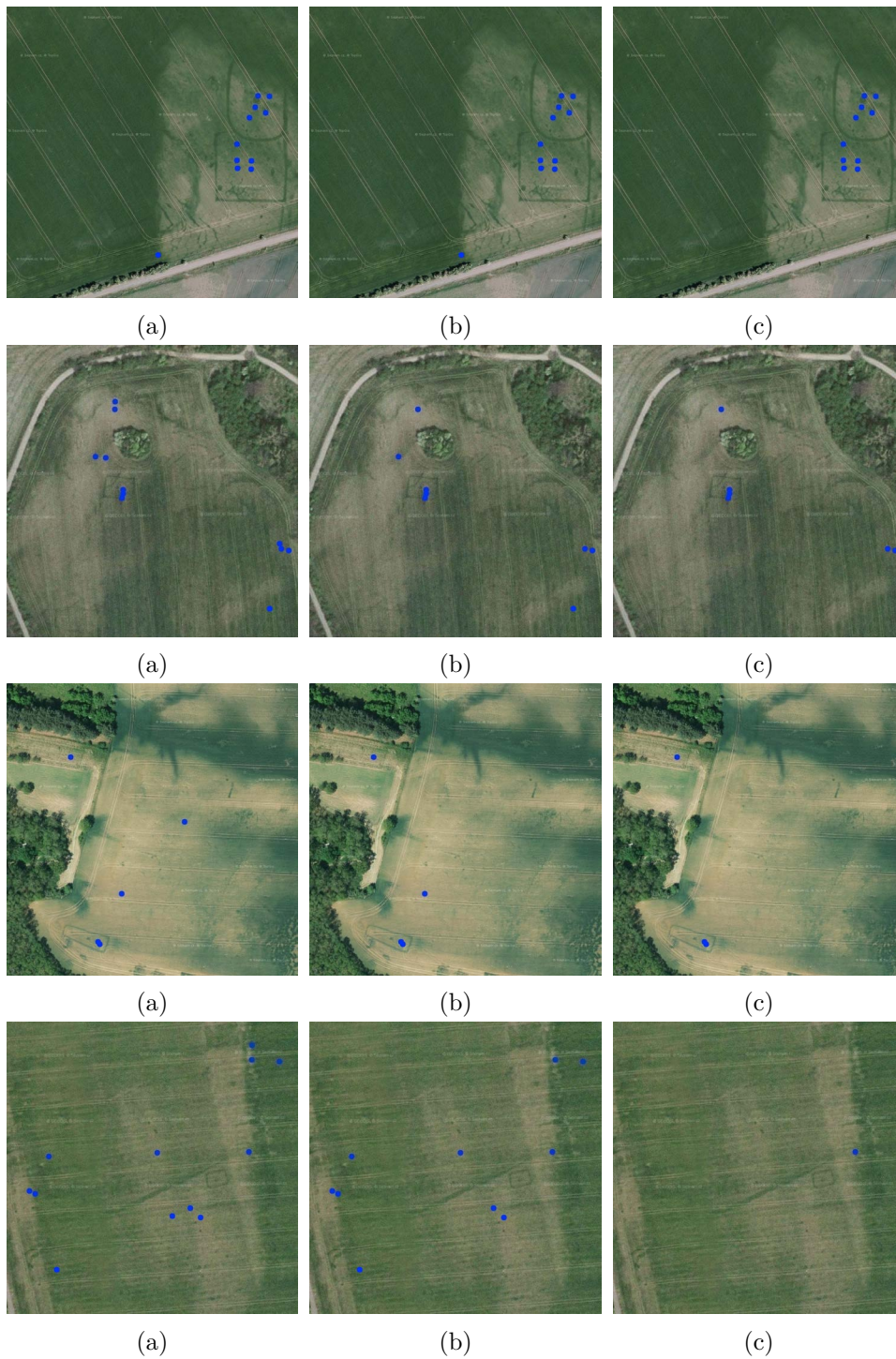


Figure F.6: RE classification results: RED input with `class_weight` parameter set to (a) .99 (b) .98 (c) .90, Part 2/3

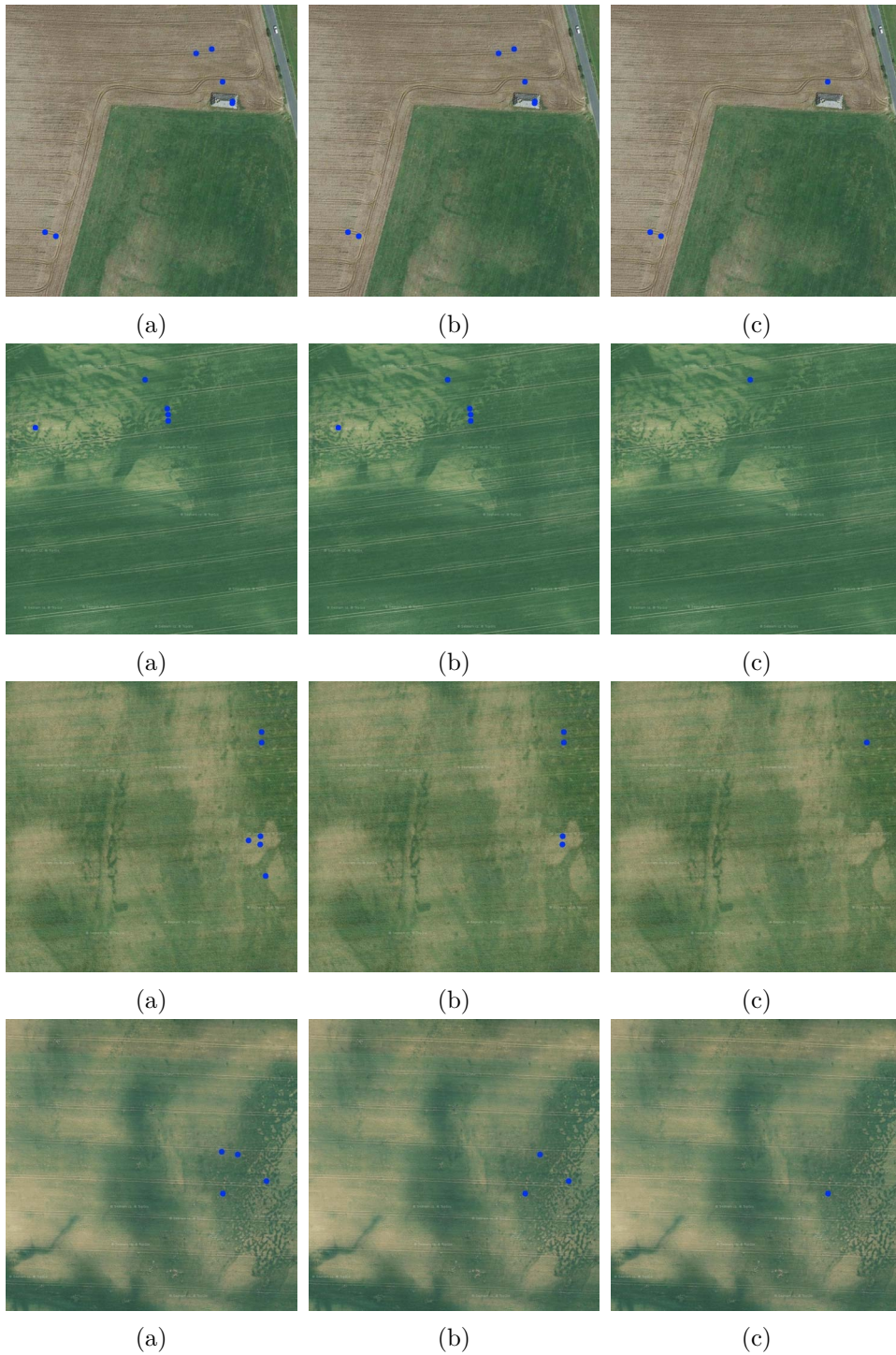


Figure F.7: RE classification results: RED input with class_weight parameter set to (a) .99 (b) .98 (c) .90, Part 3/3



Figure F.8: GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.

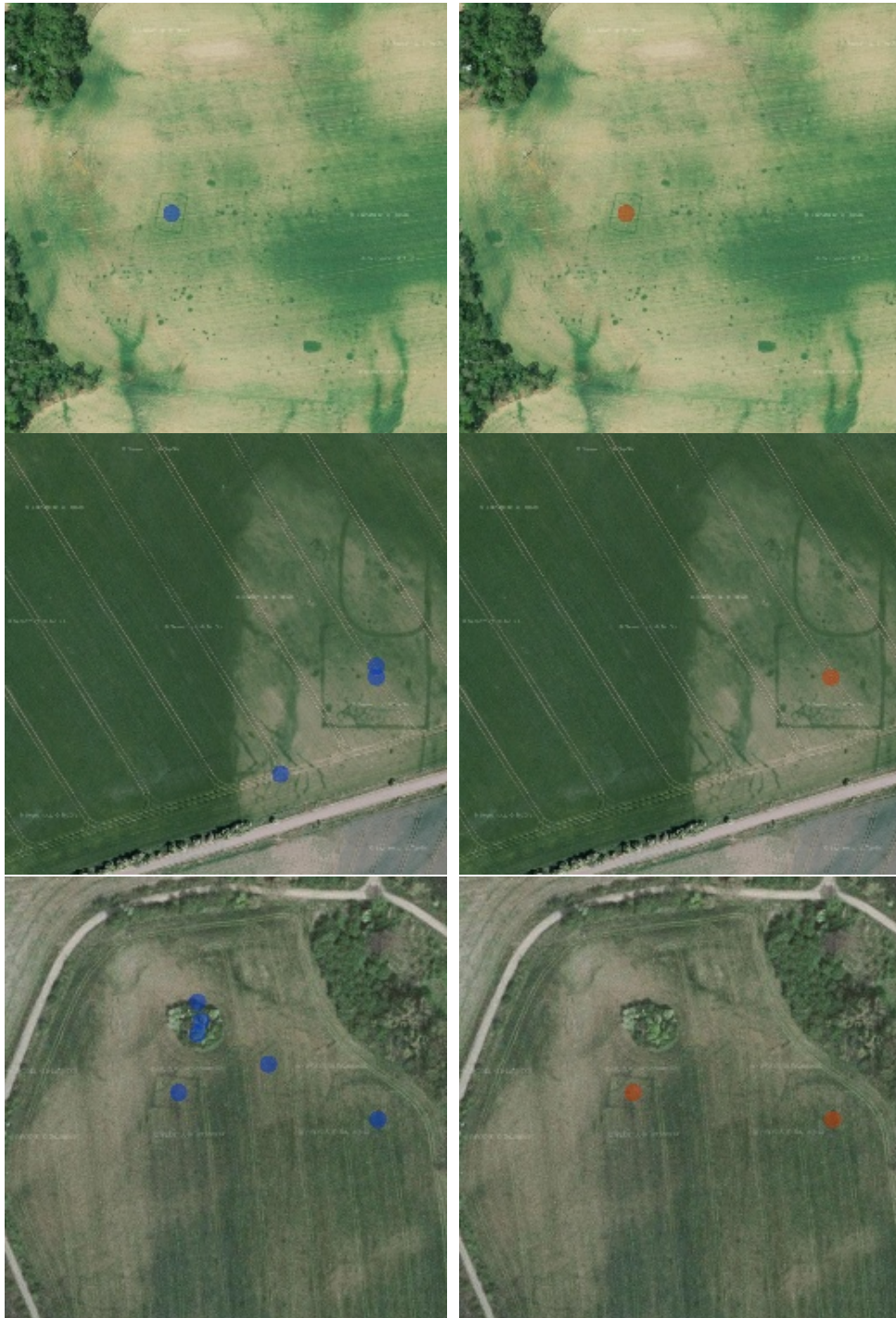


Figure F.9: GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.

F. EXPERIMENTS

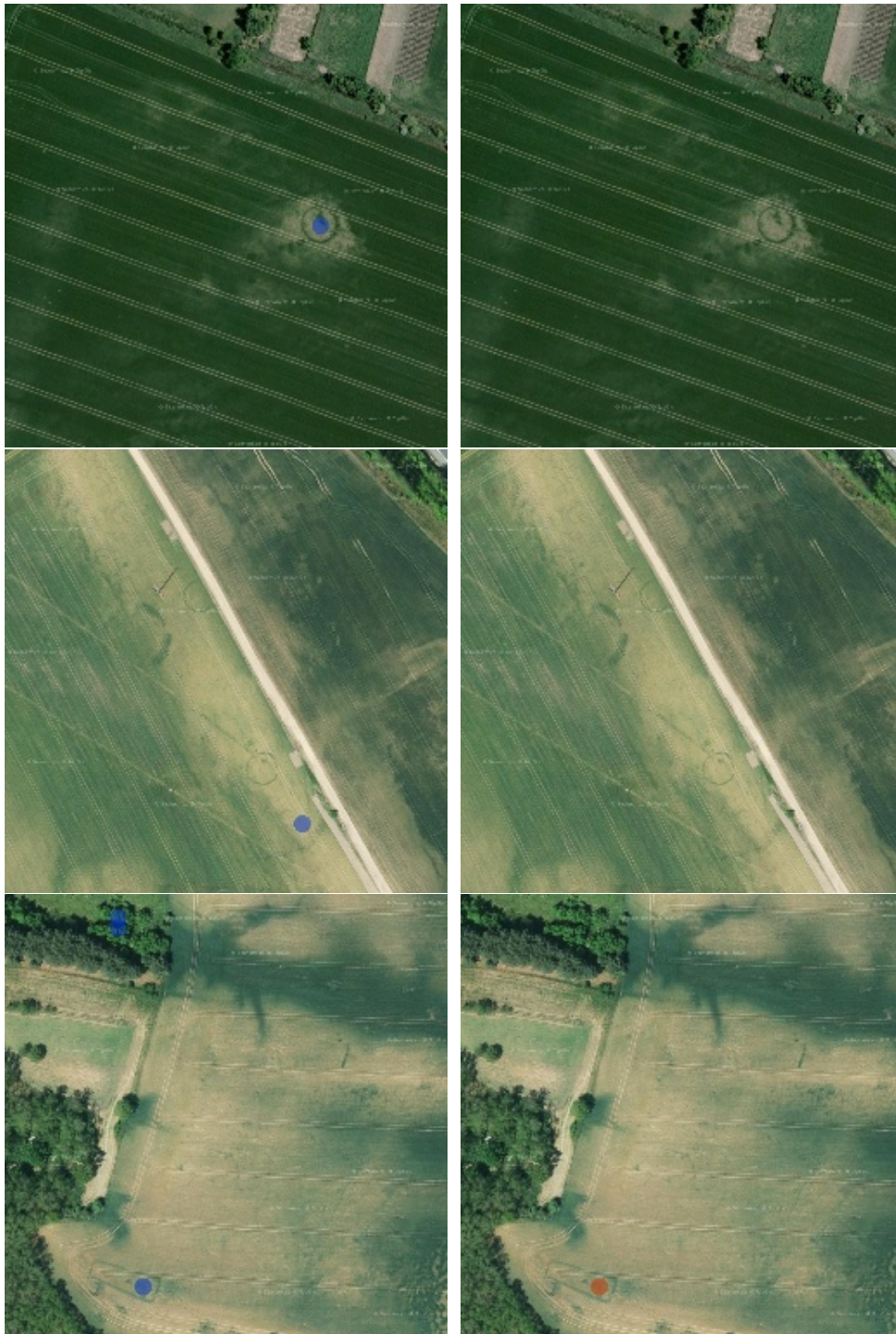


Figure F.10: GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.

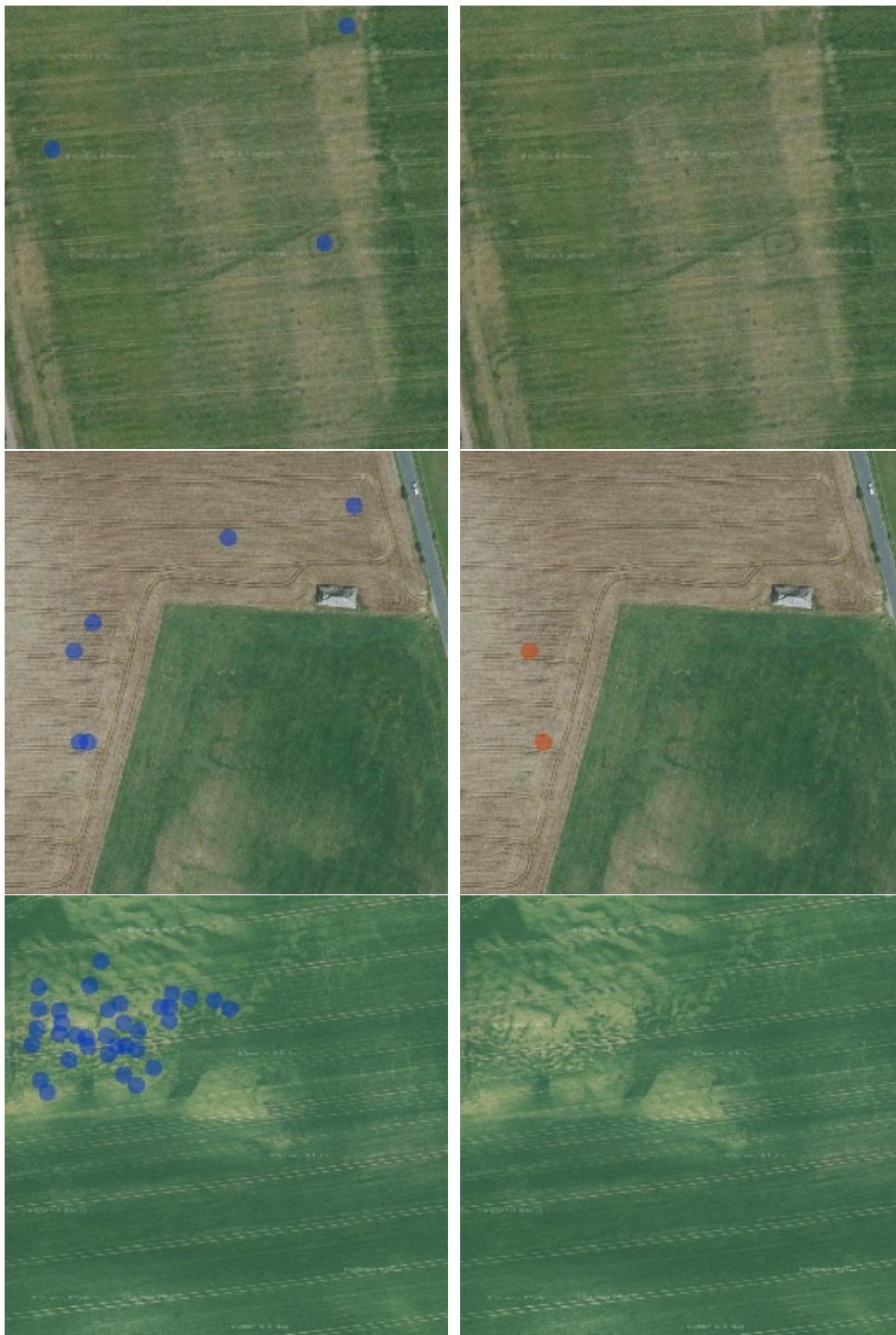


Figure F.11: GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.

F. EXPERIMENTS

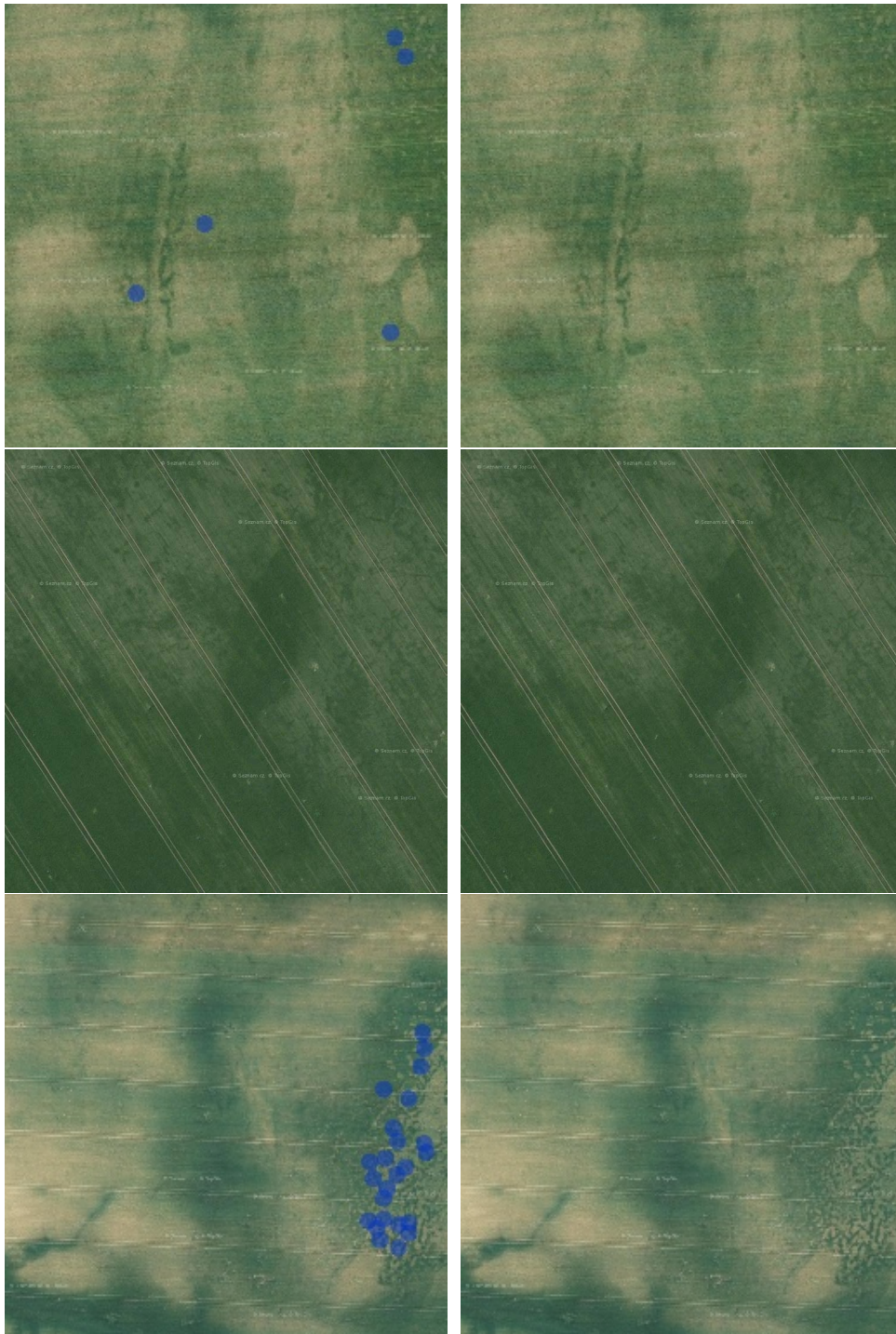


Figure F.12: GHT with rectangular templates: PC1 blob filtered dataset before (blue) and after (red) classification.



Figure F.13: GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.

F. EXPERIMENTS

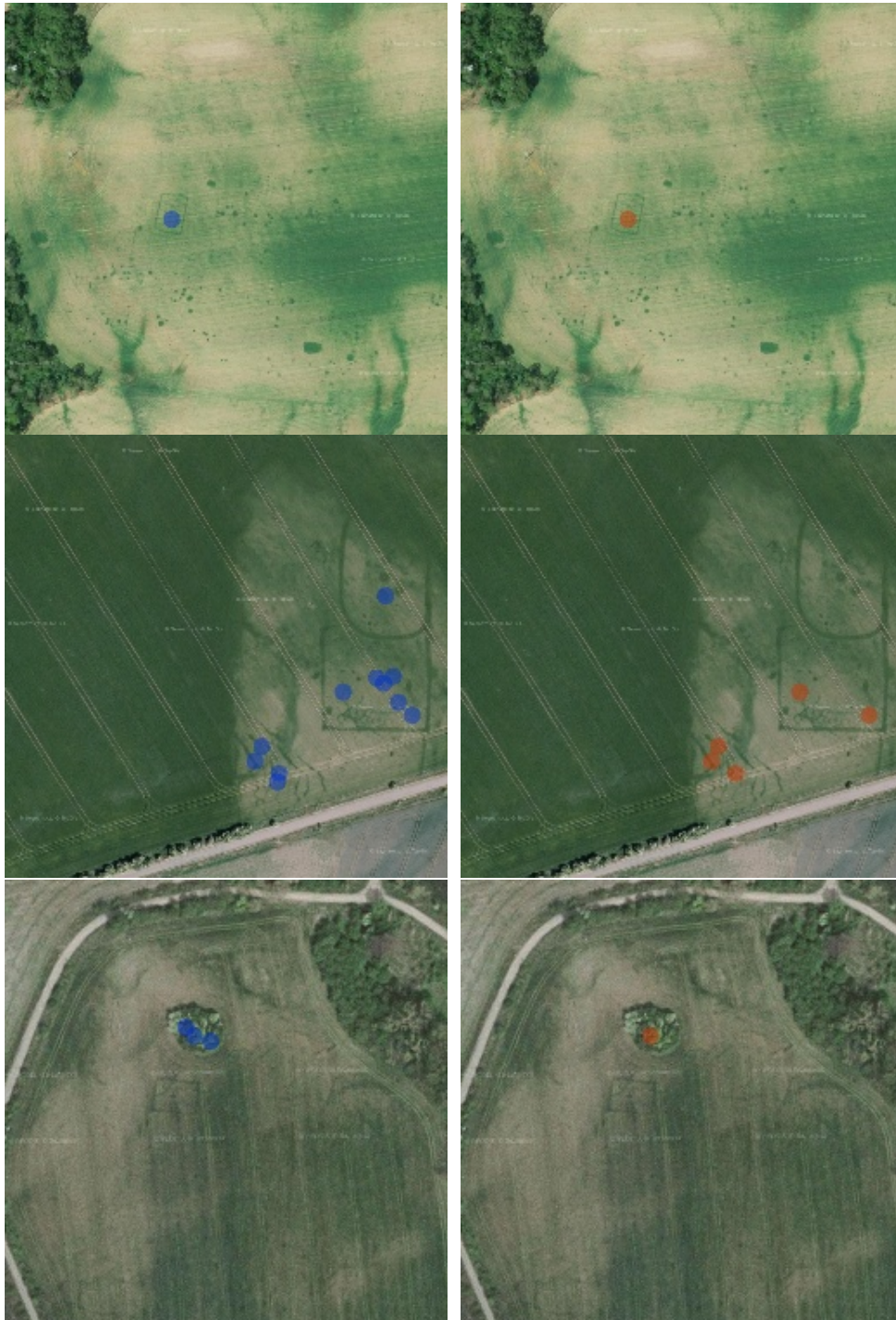


Figure F.14: GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.

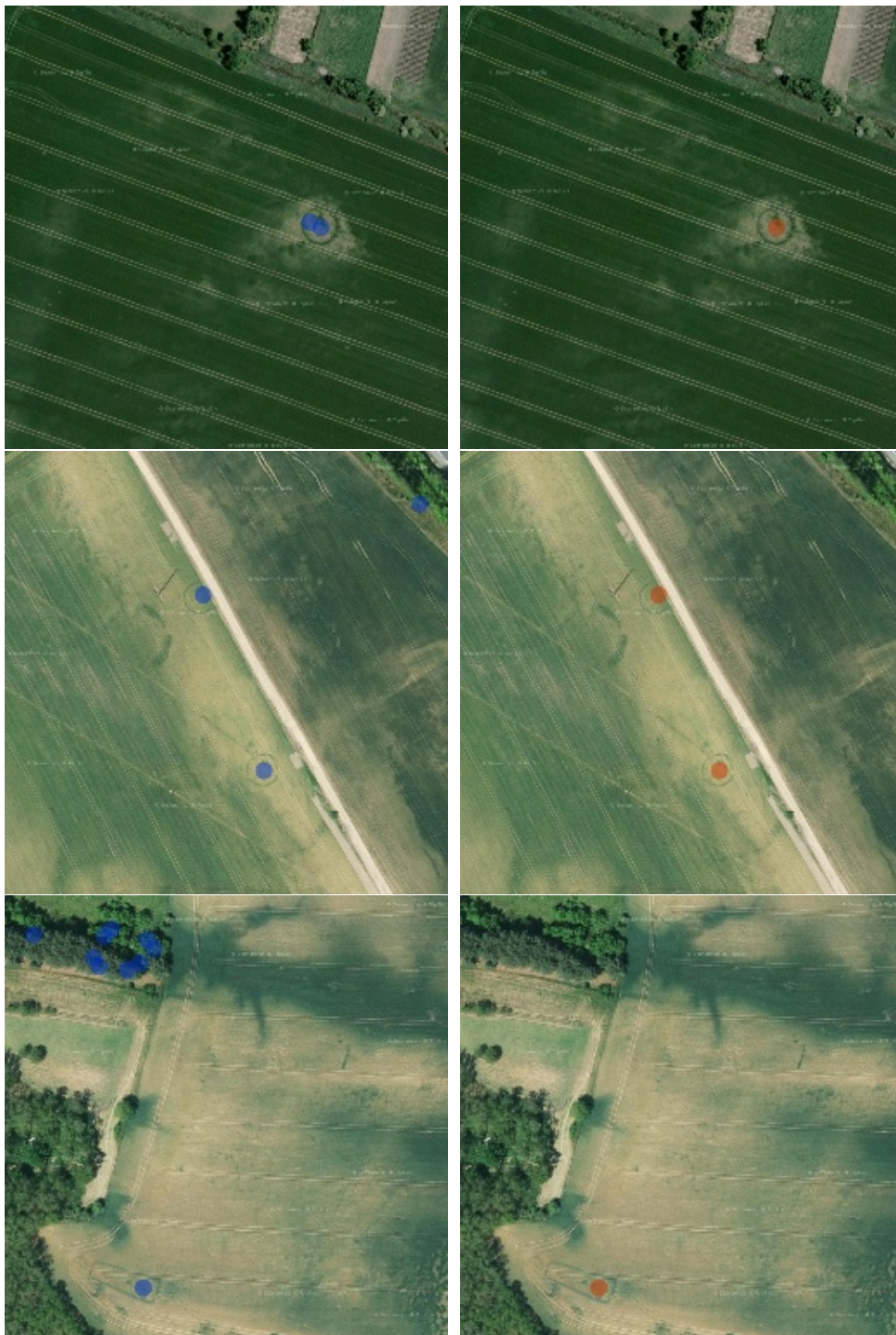


Figure F.15: GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.

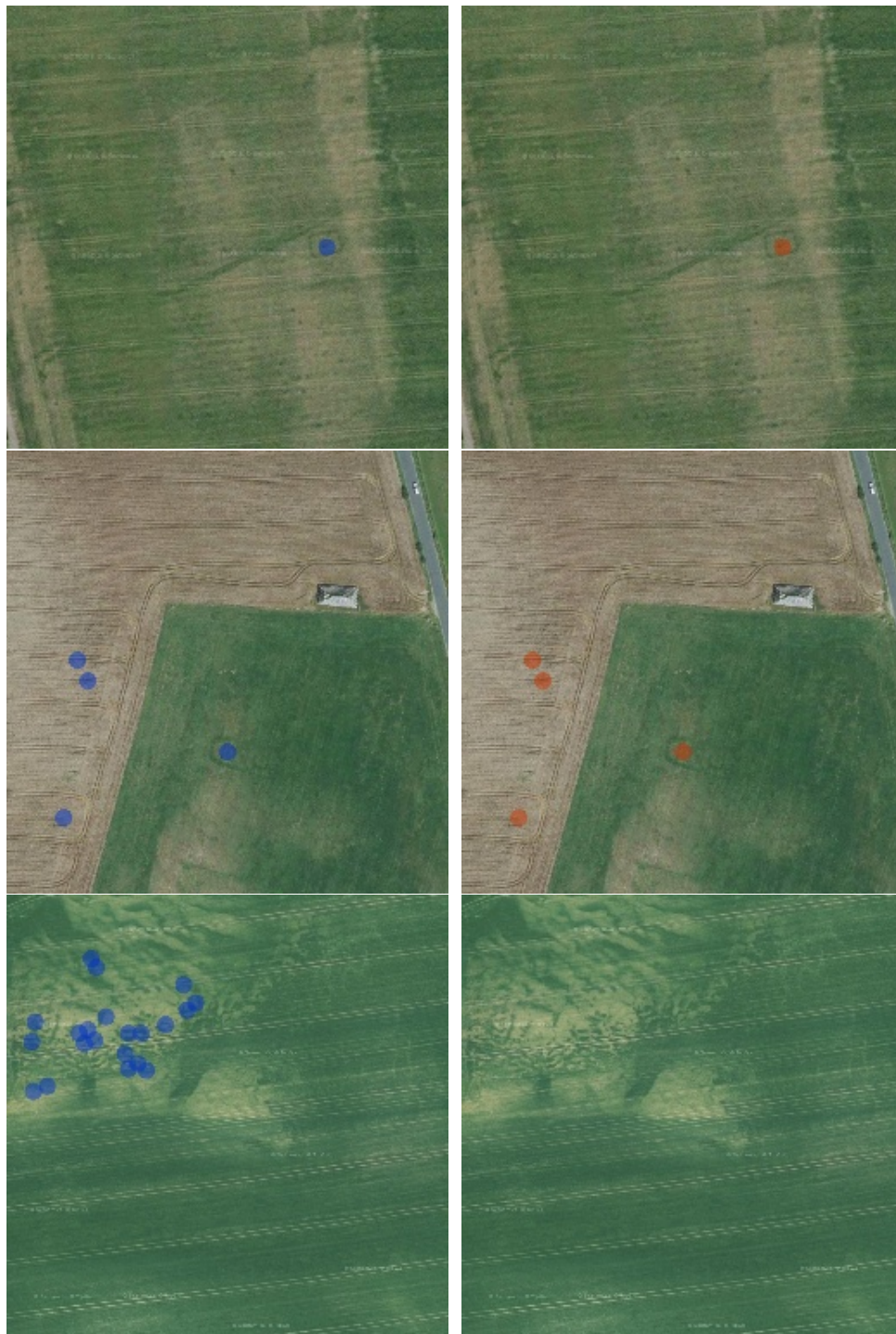


Figure F.16: GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.

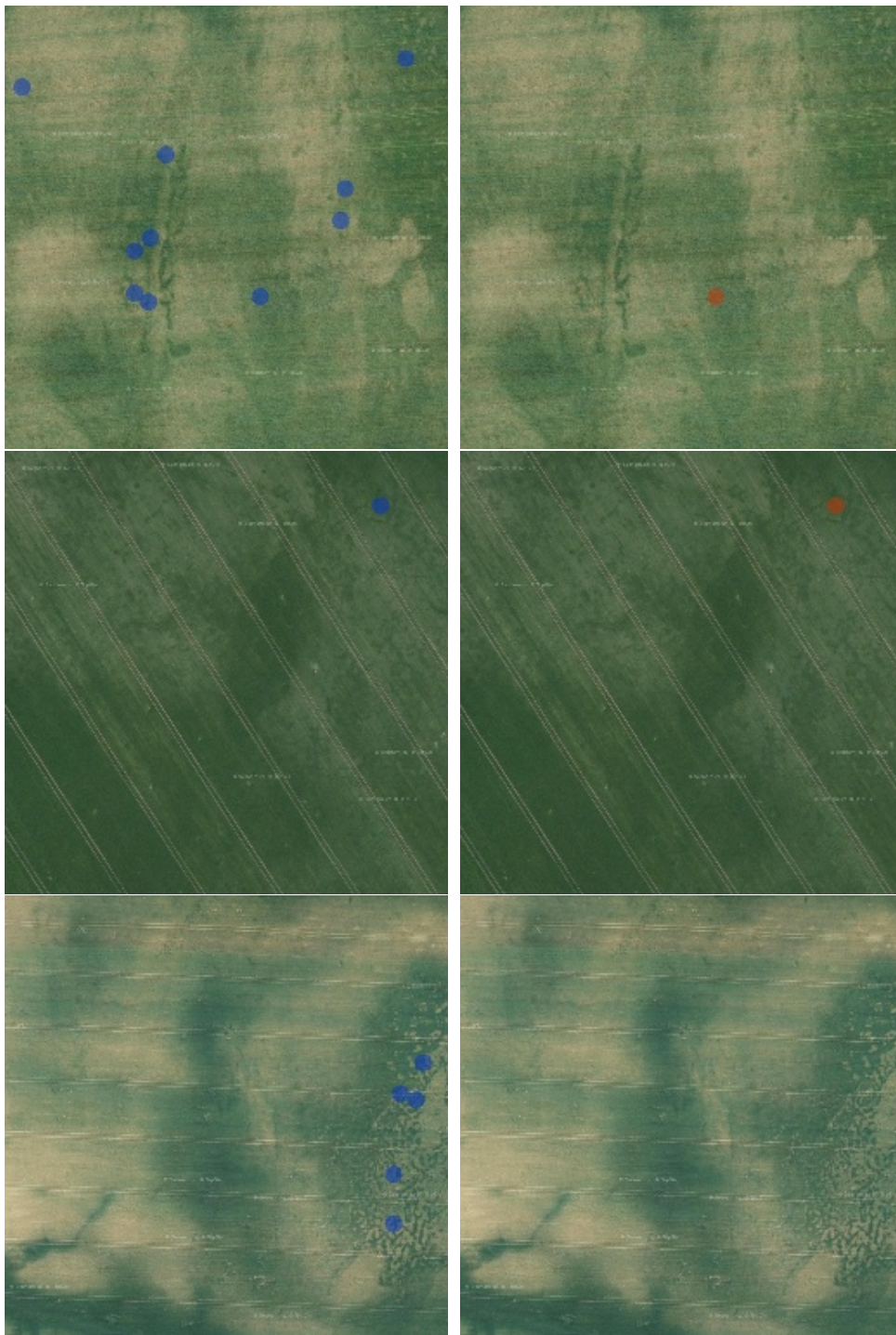


Figure F.17: GHT with circular templates: PC1 blob filtered dataset before (blue) and after (red) classification.