Statistical and image processing techniques for remote sensing in agricultural monitoring and mapping

Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Christian-Albrechts-Universität zu Kiel

vorgelegt von

Matthias Patrick Wagner

Kiel, 2020

Erste Gutachterin: Prof. Dr. Natascha Oppelt

Zweiter Gutachter: Prof. Dr. Thomas Slawig

Termin der mündlichen Prüfung: 10. September 2020

Zum Druck genehmigt: 10. September 2020

gez. Prof. Dr. Frank Kempken

Acknowledgements

Firstly, I would like to thank my supervisor Prof. Dr. Natascha Oppelt for her patience and support. I wish to also thank Prof. Dr. Thomas Slawig for the fruitful collaboration and insightful discussions.

I further thank the project partners and collaborators during the GeoCare project, namely the German Weather Service (DWD), GAF AG, the Bayerische Landesanstalt für Landwirtschaft (LfL), MunichRE, as well as the Bundesanstalt für Landwirtschaft und Ernährung (BLE) that funded the project. I acknowledge financial support by the Graduate Center of Kiel University and appreciate the valuable educational programmes they offered.

I also thank my previous and recent colleagues in the working group, namely Marcel König, Katja Kuhwald, Felix Linhardt, Alireza Taravat, Florian Uhl and Mingfeng Wang. I thank all other contributors to my papers and presentations over the years. In this context, I also thank the students and student assistants who helped in my research through field campaigns and supporting work.

I thank my family, especially my parents, grandparents, aunts and uncles for their love and support. Finally, I want to thank all my friends and everyone else who has contributed to this work in one way or another.

Abstract

Throughout most of history, increasing agricultural production has been largely driven by expanded land use, and – especially in the 19th and 20th century – by technological innovation in breeding, genetics and agrochemistry as well as intensification through mechanization and industrialization. More recently, information technology, digitalization and automation have started to play a more significant role in achieving higher productivity with lower environmental impact and reduced use of resources. This includes two trends on opposite scales: precision farming applying detailed observations on sub-field level to support local management, and large-scale agricultural monitoring observing regional patterns in plant health and crop productivity to help manage macroeconomic and environmental trends. In both contexts, remote sensing imagery plays a crucial role that is growing due to decreasing costs and increasing accessibility of both data and means of processing and analysis. The large archives of free imagery with global coverage, can be expected to further increase adoption of remote sensing techniques in coming years.

This thesis addresses multiple aspects of remote sensing in agriculture by presenting new techniques in three distinct research topics: (1) remote sensing data assimilation in dynamic crop models; (2) agricultural field boundary detection from remote sensing observations; and (3) contour extraction and field polygon creation from remote sensing imagery. These key objectives are achieved through combining methods of probability analysis, uncertainty quantification, evolutionary learning and swarm intelligence, graph theory, image processing, deep learning and feature extraction. Four new techniques have been developed.

Firstly, a new data assimilation technique based on statistical distance metrics and probability distribution analysis to achieve a flexible representation of model- and measurement-related uncertainties. Secondly, a method for detecting boundaries of agricultural fields based on remote sensing observations designed to only rely on image-based information in multi-temporal imagery. Thirdly, an improved boundary detection approach based on deep learning techniques and a variety of image features. Fourthly, a new active contours method called Graph-based Growing Contours (GGC) that allows automatized extraction of complex boundary networks from imagery. The new approaches are tested and evaluated on multiple study areas in the states of Schleswig-Holstein, Niedersachsen and Sachsen-Anhalt, Germany, based on combine harvester measurements, cadastral data and manual mappings.

All methods were designed with flexibility and applicability in mind. They proved to perform similarly or better than other existing methods and showed potential for large-scale application and their synergetic use. Thanks to low data requirements and flexible use of inputs, their application is neither constrained to the specific applications presented here nor the use of a specific type of sensor or imagery. This flexibility, in theory, enables their use even outside of the field of remote sensing.

Kurzfassung

Landwirtschaftliche Produktivitätssteigerung wurde historisch hauptsächlich durch Erschließung neuer Anbauflächen und später, insbesondere im 19. und 20. Jahrhundert, durch technologische Innovation in Züchtung, Genetik und Agrarchemie sowie Intensivierung in Form von Mechanisierung und Industrialisierung erreicht. In jüngerer Vergangenheit spielen jedoch Informationstechnologie, Digitalisierung und Automatisierung zunehmend eine größere Rolle, um die Produktivität bei reduziertem Umwelteinfluss und Ressourcennutzung weiter zu steigern. Daraus folgen zwei entgegengesetzte Trends: Zum einen Precision Farming, das mithilfe von Detailbeobachtungen die lokale Feldarbeit unterstützt, und zum anderen großskalige landwirtschaftliche Beobachtung von Bestands- und Ertragsmustern zur Analyse makroökonomischer und ökologischer Trends. In beiden Fällen spielen Fernerkundungsdaten eine entscheidende Rolle und gewinnen dank sinkender Kosten und zunehmender Verfügbarkeit, sowohl der Daten als auch der Möglichkeiten zu ihrer Verarbeitung und Analyse, weiter an Bedeutung. Die Verfügbarkeit großer, freier Archive von globaler Abdeckung werden in den kommenden Jahren voraussichtlich zu einer zunehmenden Verwendung führen.

Diese Dissertation behandelt mehrere Aspekte der Fernerkundungsanwendung in der Landwirtschaft und präsentiert neue Methoden zu drei Themenbereichen: (1) Assimilation von Fernerkundungsdaten in dynamischen Agrarmodellen; (2) Erkennung von landwirtschaftlichen Feldgrenzen auf Basis von Fernerkundungsbeobachtungen; und (3) Konturextraktion und Erstellung von Polygonen aus Fernerkundungsaufnahmen. Zur Bearbeitung dieser Zielsetzungen werden verschiedene Techniken aus der Wahrscheinlichkeitsanalyse, Unsicherheitsquantifizierung, dem evolutionären Lernen und der Schwarmintelligenz, der Graphentheorie, dem Bereich der Bildverarbeitung, Deep Learning und Feature-Extraktion kombiniert. Es werden vier neue Methoden vorgestellt.

Erstens, eine neue Methode zur Datenassimilation basierend auf statistischen Distanzmaßen und Wahrscheinlichkeitsverteilungen zur flexiblen Abbildung von Modell- und Messungenauigkeiten. Zweitens, eine neue Technik zur Erkennung von Feldgrenzen, ausschließlich auf Basis von Bildinformationen aus multi-temporalen Fernerkundungsdaten. Drittens, eine verbesserte Feldgrenzenerkennung basierend auf Deep Learning Methoden und verschiedener Bildmerkmale. Viertens, eine neue Aktive Kontur Methode namens Graph-based Growing Contours (GGC), die es erlaubt, komplexe Netzwerke von Konturen aus Bildern zu extrahieren. Alle neuen Ansätze werden getestet und evaluiert anhand von Mähdreschermessungen, Katasterdaten und manuellen Kartierungen in verschiedenen Testregionen in den Bundesländern Schleswig-Holstein, Niedersachsen und Sachsen-Anhalt.

Alle vorgestellten Methoden sind auf Flexibilität und Anwendbarkeit ausgelegt. Im Vergleich zu anderen Methoden zeigten sie vergleichbare oder bessere Ergebnisse und verdeutlichen das Potenzial zur großskaligen Anwendung sowie kombinierter Verwendung. Dank der geringen Anforderungen und der flexiblen Verwendung verschiedener Eingangsdaten ist die Nutzung nicht nur auf die hier beschriebenen Anwendungen oder bestimmte Sensoren und Bilddaten beschränkt. Diese Flexibilität erlaubt theoretisch eine breite Anwendung, auch außerhalb der Fernerkundung.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Agricultural production and food security are crucial topics facing civilization today. Recent decades saw rising global population and unprecedented growth in wealth and living standards, raising millions of people out of poverty (Roser and Ortiz-Ospina, 2013). Simultaneously, agricultural yields in Europe increased continuously in the second half of the 20th century but global productivity is expected to slow down or even decrease under different future scenarios (Asseng et al., 2015; Food and Agriculture Organization of the United Nations, 2015). A projected global population of 9.7 billion in 2050 and a growing demand for organically (supplemented) fuels and renewable energy are expected to lead to accelerating demand growth in coming decades, driving the need for higher agricultural production (United Nations et al., 2015; Alexandratos and Bruinsma, 2012). This raises the question of how best to meet global demand.

For most of history, increase in agricultural production was primarily achieved by expanded land use. The 19th and especially 20th centuries, however, saw significant technological development through industrialization and mechanization of agriculture as well as innovation in genetics and agrochemistry, most notably modern fertilizers, pesticides and higher-yielding varieties (Sands et al., 2014). As further expansion of agricultural land is impractical or undesired in many regions, intensification and sustainable use of existing farmland is key to meeting growing demand for agricultural products (Food and Agriculture Organization of the United Nations, 2018). Nevertheless, economic limitations and increasing costs often limit further intensification, while environmental concerns about effects on soil quality, ground water pollution and sustainability of agricultural production, raise opposition against existing practices (Food and Agriculture Organization of the United Nations, 2018; Hart et al., 2013).

Especially in recent years, agriculture has therefore seen a shift towards more sustainable and more efficient use of resources. Trends of automation and digitalization prove crucial in this context and offer great potential to achieve efficiency and sustainability

goals, resulting in a growing interest in techniques of precision farming and large-scale agricultural monitoring (Finger et al., 2019; Ting et al., 2011; Food and Agriculture Organization of the United Nations, 2018).

Modern imaging technology plays a key role in this technological transformation. In precision farming, land-based sensors as well as Unmanned Aerial Vehicles (UAVs) and airborne imaging are prevalent to aid in more effective use of fertilizers and herbicides, plant health monitoring, water and irrigation management as well as early detection of pests and diseases (Finger et al., 2019).

In the context of large-scale monitoring of croplands, production, management practices and resource use, remote sensing supports informed agricultural policies as well as regional planning and management of subsidies. Information on structure and distribution of agricultural activity is needed for a variety of topics such as environmental monitoring, insurance and biodiversity management (Rembold et al., 2013; García-Pedrero et al., 2017; Atzberger, 2013). Traditionally, such data is obtained through costly and time-consuming field campaigns, empirical surveys, estimations or manual interpretation (Tiwari et al., 2009; Rahman et al., 2019; Rembold et al., 2013). Automatized methods based on earth observation data, however, allow for a much cheaper and faster way to gather regular, timely information. The potential of earth observation data is particularly evident in remote places, especially in the developing world. Agricultural statistics and surveys in these regions often suffer from a lack of funding or manpower (Grassini et al., 2015).

While very high-resolution imagery from airborne and, more recently, satellite sensors is commonly used for precision farming applications, larger-scale monitoring has often been achieved using low- to medium-resolution imagery of sensors such as the Advanced Very High Resolution Radiometer (AVHRR) or the MODerate resolution Imaging Spectroradiometer (MODIS) (Rembold et al., 2013; Chaubey et al., 2011). For the most part, this is due to an inherent trade-off between spatial and temporal resolution. While lower spatial resolution platforms often provide high temporal resolution due to their wide swath, high-resolution data from platforms such as the Landsat satellites has been constrained by long revisit times (Rembold et al., 2013; Chaubey et al., 2011). Combined with obscuring by cloud cover, this inhibits sufficient availability of high-resolution imagery and complicates many possible applications, especially in the time-sensitive field of agricultural monitoring.

With trends towards open access and free-of-charge data distribution as well as government agencies embracing open data policies, however, today remote sensing data is more abundant than ever before. The opening of the Landsat archive in 2008 and the initiation of the Copernicus programme by the European Commission in partnership with the European Space Agency (ESA) in 2014, led the way for an unprecedented abundance

of free-of-charge high-resolution satellite data and the transition into a Big Data era for remote sensing (Wulder et al., 2012; Koubarakis et al., 2019). The availability of low-cost data has also enabled new applications and use cases. Reduced initial investment of imagery facilitates and encourages use by small businesses and even interested individuals. It also opened the path towards applications of both large spatial scale allowing for high-resolution analyses on national and even continental or global level as well as large temporal scale utilizing long time series spanning decades.

Especially in the context of large-scale agricultural monitoring, the Sentinel-2 satellite constellation of the Copernicus programme offers a significant improvement in several respects. It provides higher spatial resolution than current Landsat missions, more spectral information and increased coverage through the use of two identical satellites Sentinel-2A and -2B. Better exploiting these advantages is an important task in agricultural remote sensing applications today and stands in the center of this thesis.

The amount of data, however, comes at a cost in terms of processing requirements. Many traditional approaches are unsuitable or simply too inefficient for the application on such large datasets. The rapid adaptation of machine learning and artificial intelligence, primarily in the form of Artificial Neural Networks of different types demonstrate the efforts to make effective and efficient use of the newly abundant data (Tsagkatakis et al., 2019; Zhu et al., 2017; Ma et al., 2019). Advances in processing power and analytical techniques further aid in broadening the use of remote sensing data and opening up the market to new ideas, applications and technologies.

The focus of this thesis is the use of remote sensing data, particularly multispectral satellite imagery, for larger-scale monitoring applications of whole farms, large numbers of fields or on regional scales. This encompasses, in particular, exploiting the capacities of the Sentinel-2 constellation.

1.1 Background and Context

This section provides some background information for the following chapters. It gives more specific context for topics and concepts that are mentioned but not discussed in detail in the following chapters.

1.1.1 Dynamic Crop Models

In contrast to statistical models, empirical extrapolations or semi-empirical techniques based on established relationships between certain plant characteristics (e.g. photosynthetic activity and biomass accumulation), dynamic crop models attempt to simulate

actual processes and interactions between soil, plant and atmosphere (Basso et al., 2013; Atzberger, 2013). These simulations can go as far as simulating individual leaves, cells and organelles within the plant in time steps of minutes or seconds.

The most common dynamic crop model variants intended for general applications, however, are simplified to a certain degree and may include functional approximations that replace complex process simulations (Penning de Vries and Rabbinge, 1995; Van Ittersum et al., 2003b). These are usually point-based models simulating growth and development of a single plant or an assumed homogeneous field of plants in daily time steps. Famous models and model suites of this type are the Agricultural Production Systems sIMulator (APSIM), the Decision Support System for Agrotechnology Transfer (DSSAT), the WOrld FOod STudies model (WOFOST) and the AquaCrop model by the Food and Agriculture Organization of the United Nations (FAO) (Keating et al., 2003; Jones et al., 2003; Hoogenboom et al., 2019; van Diepen et al., 1989, 1994; Steduto et al., 2009; Raes et al., 2009; Hsiao et al., 2009).

In crop models, three types of information are distinguished: (a) driving variables describing external influences that are not altered by the model itself (e.g. meteorological inputs); (b) state variables representing the current state of the simulation (e.g. biomass, leaf area); and (c) parameters as fixed predefined settings (e.g. light extinction coefficients or biomass accumulation rates) (Boogaard et al., 2014). The simulation may represent these in different processes and different levels of detail. The phenological development of the plant stem, roots and/or canopy is often the core process that determines changes in growth stages and crop behavior influencing many parts of the simulation (Boote et al., 2013). Some models simulate development of individual leaves and roots (Birch et al., 1998; Fournier and Andrieu, 1998).

Closely linked to physiological development is the simulation of biomass accumulation and yield formation. This is most commonly represented by a functional relationship calculating change in biomass according to photosynthetic activity, canopy development and growth stage and obtaining final yield via a simple harvest index (percentage of total above-ground biomass) (Boote et al., 2013; Bouman et al., 1996; Van Ittersum et al., 2003a). Some models, however, attempt higher detail through simulation of individual seeds or detailed partitioning of biomass to different parts of the plant (Boote et al., 2013).

Actual photosynthetic activity is often reduced to a relationship based on light use efficiency (Monteith, 1977). The relationship between physiology and biomass accumulation is therefore reduced to empirical or functional relationships influenced by other aspects such as current growth stage or environmental factors.

Lastly, soil water balance is an important part of most dynamic crop models, often including a detailed simulation distinguishing multiple soil depths, surface runoff, percolation and drainage, water-logging and occasionally even capillary rise and salt accumulation which are especially relevant in arid regions (Van Ittersum et al., 2003b; van Keulen, 1975).

1.1.2 Data Assimilation

Data assimilation describes the process of combining observations (state estimates) from multiple sources to achieve a better estimate of a variable of interest. This may be, for example, the merging of multiple measurements or a combination of measurements with predicted and simulated values (Reichle, 2008). Most techniques also take into account uncertainties (errors) in the state estimates to gauge the new estimate towards more reliable observations.

The type of data assimilation primarily considered here is the use of external data (e.g. measurements) to improve the simulated (predicted) state of a model. Two general types of assimilation algorithms may be distinquished: filtering (sequential) and smoothing (batch) (Reichle, 2008). A filtering approach, such as the Kalman filter, performs the assimilation sequentially, i.e. progresses step by step propagating information about the state estimate through time (Kalman, 1960; Kalman and Bucy, 1961). Contrarily, a smoothing approach, such as 3DVAR or 4DVAR, considers multiple observations at different time steps within an interval simultaneously, including "future" observations (Reichle, 2008; Barker et al., 2004; Sasaki, 1970; Trémolet, 2006).

Both approaches are employed in the context of remote sensing and crop modeling. Nevertheless, a specific naming convention is used. In crop modeling, three major types of data assimilation techniques may be distinguished (Delécolle et al., 1992):

Firstly, forcing refers to the direct continuous insertion of measured data into the model, for example daily measurements of a state variable. It is debatable if this can be regarded as an assimilation process by the definition given above as it completely neglects any prediction from the model itself and merely replaces a predicted with a measured value (Delécolle et al., 1992).

Secondly, re-calibration (sometimes further separated into re-initialization and re-parametrization) refers to smoothing or batch techniques as described above and represents the process of iteratively adjusting parameters and initial state variables of a model to minimize deviations between predicted and measured variables throughout the simulation. This usually requires multiple model runs and is often performed on the predicted time series of a state variable through the entire simulation run (Dorigo et al., 2007; Verrelst et al., 2015; Rodriguez et al., 2004).

Thirdly, updating is a technique that, in contrast to re-calibration, performs assimilation during model execution and only at time steps where external observations are available. Taking both the predicted value of the model and the external observation into account, a new model estimate is obtained. This may be achieved through filtering techniques such as the Kalman filter and its variants or particle filters (Kalman, 1960; Evensen, 2003; Xie et al., 2017; Li et al., 2014).

Most commonly, data assimilation in crop modeling refers to assimilation of one or more state variables assuming that their key role in the model structure will affect the quality of the whole simulation, including other variables in subsequent time steps (Jin et al., 2018).

1.1.3 Graph Theory

A graph $G = (V, E)$ consists of a set of vertices (or nodes) $V(G)$ connected via a set of edges (or links) $E(G)$ defined by the incidence function ψ_G associating each edge with a pair of vertices. An edge $\{u, v\}$ between two vertices u and v can be directed (connecting them only in one direction) or undirected (connecting in both directions), weighted or unweighted (Bondy and Murty, 2008; Diestel, 2017). The number of vertices in a graph is given as its order $|G|$, the number of edges denoted by $||G||$. A graph may be separated into multiple subgraphs, with a graph F being a subgraph of G if $V(F) \subseteq V(G)$, $E(F) \subseteq E(G)$, and ψ_F being the restriction of ψ_G to $E(F)$ (Bondy and Murty, 2008). In this case, G contains F, $G \supseteq F$.

Two of the most important characteristics that are analyzed in graph theory are paths and cycles in graphs. A path is a graph $P = (V, E)$ with $V = \{x_0, x_i, ..., x_k\}$ and $E = \{x_0x_1, x_1x_2, ..., x_{k-1}x_k\}$ with all x_i being distinct (Diestel, 2017). The vertices x_0 and x_k are the ends linked by path P. If length $k \geq 3$ and start point x_0 and end point x_{k-1} are linked, P is called a cycle (Diestel, 2017). Relationships in a graph may be extracted, for example, by searching for the shortest paths linking vertices in a graph, the number of paths connecting two vertices, finding cycles in the graph, detecting subsets or finding cliques (parts of a graph in which all vertices are connected).

There are many different types of graphs that can be distinguished based on their properties. Some of the most important types are shown in Figure 1.1: complete (any two vertices are adjacent, i.e. connected), bipartite (vertex set can be separated into two subsets with all edges having one end in each subset) and trees (containing no cycles) (Bondy and Murty, 2008; Diestel, 2017).

Graphs can be used to represent very complex relationships and are adopted in a large variety of applications, ranging from engineering (e.g. pipeline and network struc-

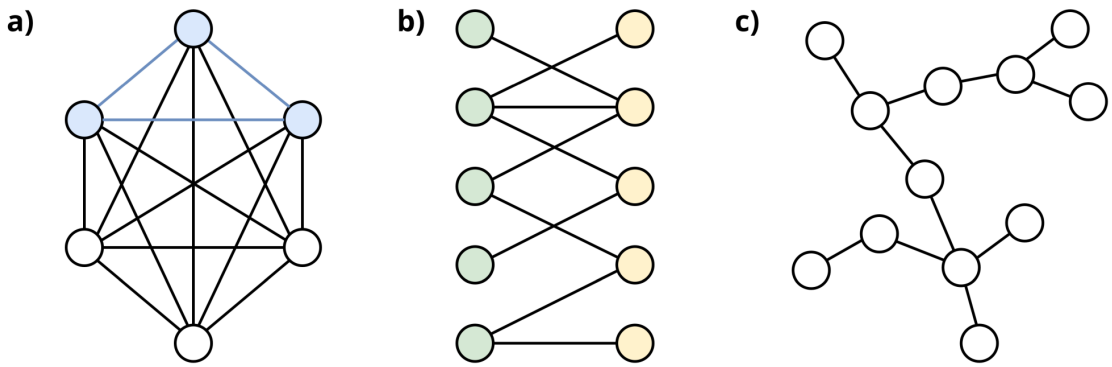

Figure 1.1. Examples of undirected, unweighted graphs: (a) complete graph (example for a cycle highlighted in blue), (b) bipartite graph (the two subsets highlighted in green and yellow), (c) tree.

tures) to social sciences (e.g. relationships within and across social groups) and image processing (e.g. graph-cut segmentation).

1.2 Data and Materials

The remote sensing data that was primarily used in all following chapters is multispectral imagery of Sentinel-2A and -2B, provided within the Copernicus programme by the European Commission in collaboration with the European Space Agency and multiple partner organizations. The data mostly comprised of Level-2A atmospherically corrected imagery. It was partly provided by GAF AG in the context of the joint research project GeoCare, partly downloaded directly from the Copernicus Open Access Hub. All subsequent processing was performed locally. Land cover datasets were obtained in the form of the CORINE Land Cover inventory and the Land Cover DE product of the German Aerospace Center (DLR) (Büttner et al., 2017; Weigand et al., 2020). Weather data was provided by the German Weather Service (DWD) as both weather station data and rasterized maps in the context of the GeoCare project.

Agricultural yield data (esp. used as reference in chapter 2) was provided by collaborating farmers as per-field mean values or geolocated point measurements from combine harvesters in the context of the GeoCare project. Reference field datasets for boundary detection and field extraction were based on ALKIS state cadastral data that was manually updated based on recent imagery with the help of student assistants (Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland, 2016).

Method development, data processing, analysis and visualization were almost exclusively performed in a Python 3 environment. Software packages such as ArcGIS Pro 2.0+, ENVI 5.3+ as well as the free software SNAP 6.0+ (incl. Sentinel toolboxes) by ESA were

occasionally used for individual tasks. More detail on datasets and the software used for their preparation is provided in subsequent chapters.

1.3 Research Objectives and Outline

The aim of this thesis is to address three important topics in the application of remote sensing in agriculture: (1) remote sensing data assimilation for improved yield prediction in dynamic crop models, (2) agricultural field boundary detection, and (3) field contour and polygon extraction to obtain information about structure and distribution of agricultural landscapes. Subsequent chapters address the following specific research objectives:

Research objective 1: Enabling flexible representation of different measurement- and model-related uncertainties in an assimilation technique for remote sensing data in dynamic crop models.

Research objective 2: Developing field boundary detection methods based on limited image information (RGB imagery) and exploring the usefulness of image features for improved detection performance using deep learning models.

Research objective 3: Developing a method for effective automatized extraction of complex, heterogeneous boundary networks and subsequent creation of field polygons at sub-pixel level with minimal supervision.

To achieve these objectives, the following chapters explore and merge aspects and techniques of multiple different fields of research, including:

- **Probability analysis:** Interpretation of probability density functions, their representation, comparison and statistical similarity to enable flexible data assimilation allowing for representation of individual uncertainties;
- **Uncertainty quantification:** Representation of different sources of uncertainties as probability density functions using Monte Carlo simulations and kernel density estimation;
- **Evolutionary learning and swarm intelligence:** Approaches of evolutionary optimization, particularly Particle Swarm Optimization (PSO), for complex optimization problems without prior knowledge of the solution space;
- **Graph theory:** Using representations of local image environments based on graph theory concepts in the form of a new contour extraction method;

- **Image processing:** Exploration and analysis of a variety of image features, filtering and image enhancement techniques for field boundary detection;
- **Deep learning:** Development and application of deep learning models for agricultural field boundary detection;
- **Feature extraction**: Use of image segmentation techniques (active contours) in combination with graph theory concepts to extract agricultural field polygons.

Chapter 2 presents a new data assimilation technique based on statistical distance between probability density functions. The method uses Monte Carlo simulations and kernel density estimation to represent different uncertainties in the model and the observations assimilated. The minimization process is achieved via minimizing the statistical distance of an optimal Gaussian distribution to the uncertainty probability density functions using PSO. The use of evolutionary learning adds flexibility in terms of the representation of uncertainties and enables an optimization with no prior knowledge of the solution space. The method is tested in the context of yield prediction using the AquaCrop-OS model, an open source implementation of the AquaCrop model by FAO, and evaluated on pixel-, field- and pixel-to-field aggregated scale in a study area near the border of the states of Niedersachsen and Sachsen-Anhalt.

Chapter 3 demonstrates a new workflow for field boundary detection based on multi-temporal RGB satellite data, (adaptive) image enhancement, filtering and edge detection techniques. This is combined with a newly developed automatic contour extraction technique called “Graph-based Growing Contours” that is capable of handling complex, heterogeneous networks of boundaries with sub-pixel precision and the capacity for adaptive branching. It requires minimal supervision and serves as a pre-step for subsequent polygon creation. The use of only RGB imagery and universally applicable image processing techniques reduces the input requirements for the presented approach and makes it largely independent of the sensor type. The method is tested on two regions of interest in a study area north of Kiel, Schleswig-Holstein.

Chapter 4 expands on the concepts of the previous chapter by replacing the boundary detection workflow with an improved detection method based on deep learning for larger-scale applications. A variety of input features is generated and distinguished by concept as well as image information used (luminosity vs color). Results are used as input to an improved version of the contour and polygon extraction methodology presented in the previous chapter. Again, only RGB data is required allowing for more general insights into the usefulness of individual features and a broader applicability of the method to different types of remote sensing imagery. Performance is tested in two study areas north of Kiel, Schleswig-Holstein, and east of Hamburg.

Finally, chapter 5 summarizes the most important results and draws conclusions on the research objectives presented above. It further expands on challenges and limitations encountered and discusses possible improvements to the methodologies. It considers implications of the research results and gives an outlook on future research opportunities.

Chapter 2

Remote Sensing Data Assimilation in Dynamic Crop Models Using Particle Swarm Optimization

M. P. Wagner, T. Slawig, A. Taravat, N. Oppelt
ISPRS International Journal of Geo-Information, 2020, 9(2), 105
Special Issue "Uncertainty Modeling in Spatial Data Analysis"
DOI: 10.3390/ijgi9020105

Received: 14 December 2019; Accepted: 7 February 2020; Published: 10 February 2020

Abstract

A growing world population, increasing prosperity in emerging countries, and shifts in energy and food demands necessitate a continuous increase in global agricultural production. Simultaneously, risks of extreme weather events and a slowing productivity growth in recent years has caused concerns about meeting the demands in the future. Crop monitoring and timely yield predictions are an important tool to mitigate risk and ensure food security. A common approach is to combine the temporal simulation of dynamic crop models with a geospatial component by assimilating remote sensing data. To ensure reliable assimilation, handling of uncertainties in both models and the assimilated input data is crucial. Here, we present a new approach for data assimilation using particle swarm optimization (PSO) in combination with statistical distance metrics that allow for flexible handling of model and input uncertainties. We explored the potential of the newly proposed method in a case study by assimilating canopy cover (CC) information, obtained

from Sentinel-2 data, into the AquaCrop-OS model to improve winter wheat yield estimation on the pixel- and field-level and compared the performance with two other methods (simple updating and extended Kalman filter). Our results indicate that the performance of the new method is superior to simple updating and similar or better than the extended Kalman filter updating. Furthermore, it was particularly successful in reducing bias in yield estimation.

2.1 Introduction

After decades of continuously rising yields, recent years have seen a slowing down in agricultural productivity growth in Europe. Furthermore, decreasing global production may be expected under certain climate scenarios (Asseng et al., 2015; Food and Agriculture Organization of the United Nations, 2015). Simultaneously, a growing world population, rising income per capita, and increasing demand for energy are expected to drive demand for agricultural products (Alexandratos and Bruinsma, 2012; United Nations et al., 2015). Combined with increasing risks of extreme weather events, these factors emphasize the need for timely and accurate crop production monitoring. A common approach is the use of dynamic biophysical crop models that simulate the soil–plant–atmosphere interface (Basso et al., 2013). These models can simulate environmental interactions and field management, but have a limited capacity to represent geospatial information on larger scales.

To address this drawback, remote sensing imagery and crop models can be merged. Remote sensing can introduce high-resolution spatial information about plant development and health into the modeling process. The increasing availability of free satellite data helps to reduce costs, especially when replacing traditional field measurements or airborne campaigns. The abundance of data from the Landsat archive and the Copernicus program by the European Space Agency (ESA) further fosters the integration of satellite data into crop models (Jones et al., 2017).

Following the early work by Delécolle et al., crop model data assimilation techniques may be categorized into three broad groups: forcing, re-calibration, and updating (Delécolle et al., 1992). Forcing refers to the replacement of simulated values with measured data. This method is very efficient and easy to implement, but has several drawbacks. First, it requires measurements for each simulation step (e.g., daily observations), which are often unavailable or need to be interpolated. When integrating optical remote sensing data, in particular, frequent cloud cover can drastically reduce the number of available observations, even with shorter revisit times in constellations such as Sentinel-2. Second, forcing effectively breaks up the simulation loop because it replaces intermediate results with external inputs (Rembold et al., 2013). Third, it does not consider measurement un-

certainties and therefore directly transfers errors to the model. Due to these drawbacks, a few recent studies have considered forcing.

A frequently applied technique is re-calibration, sometimes separated into re-initialization and re-parametrization. Here, the initial values and parameters of the crop model are iteratively changed by minimizing a cost function measuring the distance between the simulated state variables and observed ones (Delécolle et al., 1992; Maas, 1988). Re-calibration therefore obtains a new set of parameters or initial values, thus allowing a simulation that resembles better observations. Although this method often improves model-based yield predictions, it has two flaws. First, re-calibration settings may be unrealistic or may represent an unreliable parameter setup (Maas, 1988). Second, re-calibration can be computationally demanding because it requires multiple re-runs of the model, hampering larger scale applications.

Updating performs the assimilation during the simulation, only interfering when an observation is available. It therefore performs well even with few and infrequent observations and reduces processing time when compared to re-calibration. Furthermore, updating allows uncertainties in both the simulation and the data assimilated to be addressed (Dorigo et al., 2007). However, it requires modifications in the model itself (i.e., the source code) and not all models allow such interference. The most commonly used updating techniques are the (extended) Kalman filter, particle filter, and the ensemble Kalman filter (Kalman, 1960; Evensen, 2003; Del Moral, 1997; Kalman and Bucy, 1961).

Following the definition by Kennedy and O'Hagan (2001), model uncertainties may be classified into parameter, parametric, model inadequacy, residual variability, observation, and code uncertainties. In the context of biophysical modeling, the most relevant sources of uncertainties are parameter uncertainty (errors related to suboptimal parameter settings), parametric or input uncertainty (errors in the input data driving the simulation, e.g., daily weather measurements), code uncertainty (approximations and inaccuracies in model implementation), and model inadequacy (e.g., model bias). Uncertainties related to implementations and inadequacies are usually addressed during model development and subsequent calibration and sensitivity studies (Asseng et al., 2013; Warszawski et al., 2014; Hoffmann et al., 2015; van Bussel et al., 2016). Parameter and input uncertainties, however, are highly application- and context-dependent and need to be assessed individually.

Most updating approaches are robust and fast, but often lack a detailed representation of such uncertainties. The Kalman filter, for example, approximates uncertainties in the model and the measurement by a simple scalar (e.g., the standard deviation in repeated measurements) or a covariance matrix in the case that multiple variables are updated (Reichle, 2008). This approach does not allow for a detailed handling of different

uncertainty sources. Techniques such as the above-mentioned ensemble Kalman or particle filter, may account for uncertainty in parameters and model states stochastically.

Both re-calibration and updating require the solution of an optimization problem, which is usually non-linear. For such kinds of problems, several numerical algorithms can be applied. In our updating technique, we employed particle swarm optimization (PSO) due to its reliable global optimization capacities and flexibility in inputs and objective functions (see Section 2.2.2.3). PSO has seen various applications in remote sensing, frequently in image segmentation and classification (Liu et al., 2008; Shen et al., 2018; Bansal et al., 2009), but also in agricultural applications. Guo et al., for example, used the algorithm to couple the PROSAIL canopy reflectance model with the WheatGrow crop model based on vegetation indices (Guo et al., 2017). Others have used it in combination with multiple classifiers and algorithms for crop classification (Omkar et al., 2008). The most frequent application, however, is the (re-)calibration of crop models such as the WOrld FOod STudies model (WOFOST) (Jin et al., 2015), the Simple Algorithm for Yield Estimate (SAFY) (Silvestro et al., 2017), the Decision Support System for Agrotechnology Transfer (DSSAT) (Li et al., 2015; Son et al., 2016), or AquaCrop (Jin et al., 2016; Jibo et al., 2016).

The main objective of this study was therefore to ensure increased flexibility of uncertainty handling. The new technique proposed allows the user to include different uncertainties in the process with minimal limitations on their type and definition. The technique should also be largely independent and self-calibrating to enable direct application with minimal prior adjustments, thus allowing fast assimilation of remote sensing observations.

Although many studies exist that combine remote sensing inputs with dynamic crop models, a direct comparison is difficult to draw. The diverse nature of approaches involving different sensors, input variables, crop types, crop models, calibration settings, application scales (field to national or even continental and global) and varying amounts of prior knowledge (e.g., detailed study plots with regular measurements), aggravate a direct comparison. To demonstrate the potential of the new method, we therefore decided to apply multiple updating schemes to the same datasets with the same model and calibration settings. We compared the results of the new approach to a simple updating scheme (replacing values in the model simulation directly) as well as an extended Kalman filter (EKF). As a case study, we assimilated Sentinel-2 canopy cover (CC) data into the AquaCrop-OS model v5.0a to improve the winter wheat yield estimation.

The rest of the paper consists of five parts. Section 2.2 describes the study area and data used and introduces the methodology. We provide some methodological background first, followed by a description of the updating technique. In Section 2.3, we describe

results and discuss them in Section 2.4. Finally, Section 2.5 will give a short conclusion and outlook.

2.2 Materials and Methods

2.2.1 Datasets

2.2.1.1 Study Area

Our study area is located in central Germany near the border of the states of Niedersachsen and Sachsen-Anhalt (see Figure 2.1). The climate is temperate/oceanic with warm summers and wet winters (Cfb in Koeppen-Geiger climate classification) (Kottek et al., 2006). The region is relatively warm and dry with an average temperature of 8.2 °C and an annual precipitation of 538 mm in the climate reference period 1960–1990 (German Weather Service (DWD), 2019). Our weather data of the years 2016 and 2017 revealed both years to be rather warm (9.8 and 10.8 °C), while precipitation was low in 2016 (436 mm) and high in 2017 (679 mm) compared to the long-term average. Soils in the region are typically stagnosols and brown earths originating from sandy and loamy glacial debris. Further, clayey soils from skeletal loam, sandy Loess over limestone, rendzinas, and some podzols also occur (Bundesanstalt für Geowissenschaften und Rohstoffe, 2018).

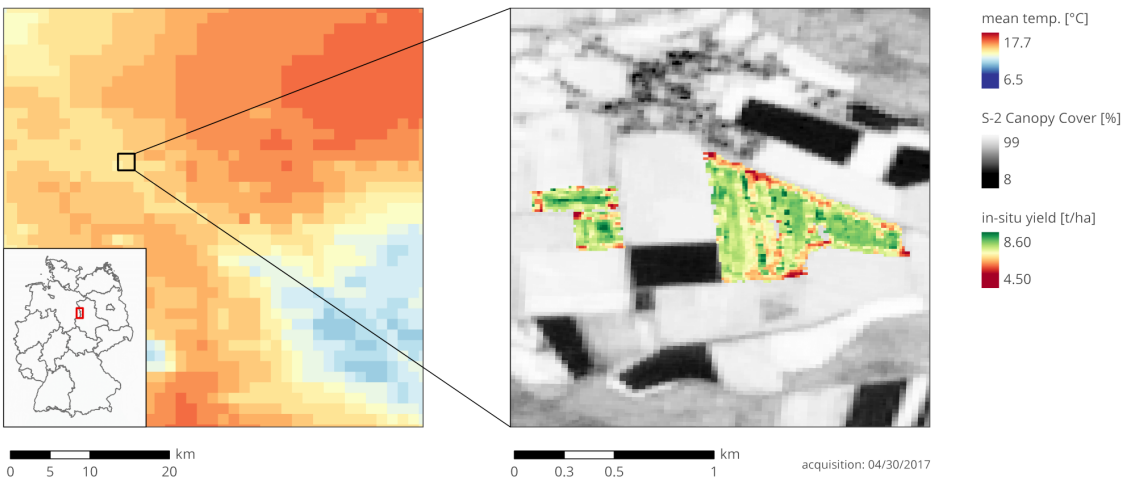

Figure 2.1. Map of study area and example of rasterized weather datasets in the form of mean air temperature (left). Example of yield data and canopy cover maps (right).

2.2.1.2 Weather Data

The German Weather Service (DWD) delivered daily weather data for the nearby weather station "Ummendorf" (11.18° E, 52.16° N) as well as 1 × 1 km² rasterized weather datasets for the whole of Germany (see Figure 2.1). Weather data include daily minimum and maximum temperatures, precipitation sums, and reference evapotranspiration based on the Penman–Monteith equation (Doorenbos and Pruitt, 1977). The raster datasets were used as input to the model, introducing a limited amount of spatial dynamics.

2.2.1.3 Canopy Cover Data

Our database consisted of atmospherically corrected Sentinel-2 Level-2A scenes between August 2015 and November 2017. We only considered scenes with generally low to moderate cloud cover (up to 50%). The dataset comprised of 116 scenes, covering the full winter wheat growing seasons for both harvest periods of 2016 and 2017. We used the biophysical processor implemented in the ESA Sentinel-2 Toolbox (S2TBX, version 6.0.4) to generate canopy cover (CC) maps from all scenes (see Figure 2.1). The processor employs artificial neural networks trained on a large dataset of radiative transfer simulations of canopy and leaf properties (Weiss and Baret, 2016). The documentation of the SNAP Biophysical Processor provides some theoretical performance indicators. The authors claim a low root mean square error (RMSE) of 0.04 for CC predictions on their validation dataset (Weiss and Baret, 2016). During pre-processing, we further performed a multi-threshold cloud and cloud shadow detection for each of our test fields to discard any potentially contaminated observations. The resulting number of observations ranged between three and 12 per growing season, depending on location.

2.2.1.4 Yield Data

Field data were obtained via GPS-based yield measurements on combine harvesters during harvest of 30 fields in 2016 and 32 fields in 2017. We removed outliers outside +/− 2.58 standard deviations (99% threshold in a standard normal distribution), particularly false zero measurements that frequently occurred at the start and end of the harvest procedure. We then aggregated the remaining points to 10 × 10 m² yield maps matching Sentinel-2 observations (see Figure 2.1). The resulting mean yields of all fields were in good agreement with the reported department-level yield statistics (Statistisches Landesamt Sachsen-Anhalt, 2019; Niedersachsen, 2018a,b). The observed yield on the pixel-level ranged from 2.38 to 9.60 t/ha, and the mean field yields ranged from 3.90 to 7.63 t/ha. No information on measurement accuracy was provided. For further analysis, we generated a pixel- and a field-level dataset. We split both randomly into 60% calibration (32 field

observations, 23,375 pixel observations) and 40% validation (20 field observations, 15,584 pixel observations) data.

2.2.2 Methodological Background

2.2.2.1 AquaCrop-OS Description

AquaCrop is a dynamic crop model developed by the Food and Agriculture Organization of the United Nations (FAO). It simulates the yield response of herbaceous crops on a homogeneous field, considering water response and various stress effects (Steduto et al., 2009; Raes et al., 2009; Hsiao et al., 2009). Inputs for daily simulation are the maximum and minimum temperature data, precipitation sum, and potential evapotranspiration (Steduto et al., 2012). The simulation is considerably simplified compared to complex model suites such as the Decision Support System for Agrotechnology Transfer (DSSAT) (Ines et al., 2013; Jones et al., 2003), focusing on a global model applicability with a potentially limited range of available data. The central part of the model is a crop productivity function that relates biomass accumulation to water productivity and evapotranspiration to obtain the total cumulative biomass:

$$B_T = Ks_b \cdot WP^* \cdot \sum t = 0^T \left(\frac{Tr_t}{ET_{ot}} \right) \tag{2.1}$$

where B_T is the total accumulated biomass from $t = 0$ days to $t = T$; K_{s_b} is an air temperature stress coefficient; WP^* is the water productivity normalized to annual mean CO_2 concentration; Tr_t is daily crop transpiration; and ET_{ot} is daily potential evapotranspiration (both in mm).

AquaCrop represents the heat, drought, and cold stress effects via stress coefficients that can influence canopy development, stomatal conductance, canopy senescence, or harvest index development. The stress coefficients change with the level of stress following a convex to concave response curve (Raes et al., 2009, 2018):

$$Ks = 1 - \frac{e^{S_{rel} f_{shape}} - 1}{e^{S_{rel}} - 1} \tag{2.2}$$

where Ks describes the stress response function; S_{rel} is the relative stress level (≤ 1); and f_{shape} is a shape factor defining the curvature of the function. The main state variable in the model is canopy cover (CC; sometimes referred to as Fraction of Vegetation Cover,

FVC or FCOVER) that directly influences Tr_t in Equation 2.1 via a crop transpiration coefficient:

$$K_{C_{Tr}} = CC^* K_{C_{Tr,x}} \tag{2.3}$$

$$Tr = K_{sw} K_{C_{Tr}} ET_o \tag{2.4}$$

where CC^* is the current canopy cover (adjusted for micro-adjective effects); $K_{C_{Tr,x}}$ is the maximum crop transpiration coefficient for well-watered soil and a complete canopy; K_{sw} represents a soil water stress coefficient; and $K_{C_{Tr}}$ is the current crop transpiration coefficient obtained. Therefore, CC is an important variable in biomass accumulation in Equation 2.1, and consequentially, yield as determined via a harvest index (i.e., percentage of biomass at crop maturity).

CC development over the growing season is determined mostly empirically. After crop emergence, CC first increases exponentially up to 50% of the maximum. A slowing growth follows until the maximum is reached. The value of CC stays constant until an exponential decay sets in at the beginning of senescence (Raes et al., 2018). This process is summarized in the following equations:

Canopy expansion:

$$CC = CC_o e^{tCGC} \quad for\ CC \leq \frac{CC_x}{2} \tag{2.5}$$

$$CC = CC_x - 0.25 \frac{CC_x^2}{CC_0} e^{-tCGC} \quad for\ CC \geq \frac{CC_x}{2} \tag{2.6}$$

Canopy senescence:

$$CC = CC_x[1 - 0.05(e^{\frac{CDC}{CC_x}} - 1)] \quad for\ CC \leq \frac{CC_x}{2} \tag{2.7}$$

where CC is the new canopy cover; CC_x is the maximum possible canopy cover; CC_0 is the initial canopy cover at the start of growth; and CGC and CDC are canopy growth and decline coefficients, respectively. Dry yield is obtained by applying a harvest index (percentage) to the biomass value at maturity.

We used the open source version of the model called AquaCrop-OS, allowing us to make the necessary source code changes for the updating procedures described in Section 2.2.3 (Foster et al., 2017).

2.2.2.2 AquaCrop-OS Calibration

Due to the lack of information on winter wheat varieties on our test fields or regular in situ sampling, our prior knowledge for calibration was limited. We therefore relied on an empirical calibration of model parameters. This also made the simulation more general and independent of specific field conditions.

AquaCrop-OS offers a large number of crop parameters, separated into conservative ones that have previously been proven accurate in many different environments and user-specific ones (Steduto et al., 2012). The former were ignored for the most part in our calibration, except for the Canopy Growth and Decline Coefficients (CGC and CDC, see Table 2.1) due to their particular relevance in this context. We did not consider irrigation management because agriculture in our study area is exclusively rain-fed. Similarly, we assumed that field management follows a "best practice" due to high technological standards and a long tradition of industrialized agriculture in our study area.

Table 2.1. Calibration ranges and obtained calibrated crop parameters in AquaCrop-OS.

Parameter	**Description**	**Calibration Range**	**Calibrated Value**
CDC	Canopy Decline Coefficient	$0.0015 - 0.0065$	0.0038
CGC	Canopy Growth Coefficient	$0.0025 - 0.0075$	0.006
Emergence	Time from sowing to emergence in Growing Degree Days (GDD)	$112 - 225$	188
fshape_b	Shape factor of biomass productivity reduction due to insufficient GDDs	$10.36 - 17.27$	14.504
HIstart	Time from sowing to start of yield formation in GDDs	$660 - 1.100$	748
Kcb	Maximum crop coefficient at full canopy development	$0.825 - 1.375$	1.045
Maturity	Time from sowing to maturity in GDDs	$1.800 - 3.000$	2.16
Senescence	Time from sowing to senescence in GDDs	$1.275 - 2.125$	1.615

We performed a sensitivity analysis based on iterative changes to individual parameters and observed the influence on predicted yield. It revealed the parameters listed in Table 2.1 to be those most relevant for winter wheat yield prediction in our study area. We calibrated the model by altering parameters iteratively and minimizing the RMSE

of yield. The calibration process improved the RMSE from 2.305 t/ha to 1.324 t/ha on field-level and from 2.264 t/ha to 1.521 t/ha on the pixel-level validation datasets. The optimal parameter settings for the pixel- and field-level were quite similar, so we decided to use the same calibration set on both scales. Table 2.1 provides a list of calibration settings.

2.2.2.3 Particle Swarm Optimization

Particle swarm optimization is a metaheuristic global optimization algorithm based on swarm intelligence principles of complex intelligent behavior emerging from primitive individual agents. As such, it is part of the larger family of evolutionary computing (Kennedy et al., 2001; Yang, 2014). Kennedy and Eberhart originally designed the algorithm following previous efforts by Reynolds in simulating realistic movements of bird flocks (Kennedy and Eberhart, 1995; Reynolds, 1987).

The particle swarm is a group ("swarm") of candidate solutions ("particles") moving in the multidimensional search space over time (i.e., iteration steps). Each particle is initiated as a random vector with a random initial velocity vector representing its movement in the search space. This velocity is updated at each iteration based on certain rules and the new particle fitness is obtained. In the original version, the process is only influenced by the best previous solution of the particle (previous best, pbest) and the best solution obtained in its neighborhood (neighborhood best, nbest) (Kennedy et al., 2001; Yang, 2014). This neighborhood is described by the topology representing connections between the particles in the swarm. There are many different topologies used in the literature including local best, global best, and von Neumann topologies, but also dynamic topologies changing throughout the process based on time, Euclidian distance, and fitness–distance ratios, among others (Peram et al., 2003; Mendes et al., 2003). For a more detailed discussion, readers may refer to the paper by Poli et al. (2007). The following equations describe the central velocity and position update (all multiplications are element-wise):

$$\vec{v_i}(t) = \vec{v_i}(t-1)\varphi_1\epsilon_1(\vec{p_i} - \vec{x_i}(t-1)) + \varphi_2\epsilon_2(\vec{p_n} - \vec{x_i}(t-1)) \tag{2.8}$$

$$\vec{x_i}(t) = \vec{x_i}(t-1) + \vec{v_i}(t) \tag{2.9}$$

where $\vec{v_i}(t)$ is the new (updated) velocity vector of particle i at time step t and $\vec{v_i}(t-1)$ is its previous velocity vector. The previous and new positions are given by $\vec{x_i}(t-1)$ and $\vec{x_i}(t)$, respectively. The previous best solution is represented by $\vec{p_i}$ and the neighborhood best solution by $\vec{p_n}$. The terms φ_1 and φ_2 refer to pbest and nbest coefficients, respectively, and ϵ_1, ϵ_2 are random vectors (Kennedy et al., 2001). One can

interpret pbest and nbest coefficients as the tendency of particles to move independently or "toward the swarm". The two elements are therefore closely related to exploration and exploitation.

Its metaheuristic approach distinguishes PSO from gradient-based optimization techniques. PSO does not use exact or approximated derivative information. It therefore does not need continuous or differentiable objective functions or any prior knowledge about the cost function (del Valle et al., 2008). This makes it very flexible in handling different types of inputs and even combinations of continuous and discrete functions. PSO is also considered to be reliable in finding global optima, even in highly heterogeneous, complex solution spaces as simulated by test functions like the Ackley or Hölder table functions (Kennedy et al., 2001; Liu et al., 2011). Moreover, PSO scales very well with high-dimensional inputs as the number of function evaluations is determined by the swarm size, not the number of input variables.

However, PSO is not a deterministic algorithm, but includes stochastic elements. The process is therefore not entirely predictable, even identical starting conditions may lead to different iteration steps and even to different solutions due to the random component of the process (Poli et al., 2007). As a result, it is up to the user to determine application-specific parameters (such as swarm size, coefficients, topology, etc.) that ensure a reliable and fast convergence. Furthermore, unlike gradient descent-related algorithms that reach a local minimum under certain assumptions, the convergence in PSO methods is only valid in a stochastic setting.

To ensure fast and reliable optimization results, we compared a number of different PSO variants and settings. This included different swarm sizes, inertia weights, cognitive and social coefficients, static and dynamic topologies (local best, global best, dynamic nearest neighbor, dynamic fitness–distance ratio, among others) as well as different distributions for random vector sampling (uniform, normal, Lévy).

Table 2.2. Settings used for particle swarm optimization algorithm.

Parameter	**Value**
swarm size (n)	20
pbest coefficient (φ_1)	2.05
nbest coefficient (φ_2)	2.05
maximum velocity (v_{max})	dynamic range
constriction coefficient (χ)	0.72984
topology	von Neumann (4 neighbors)
random distribution	uniform

We observed that Clerc's constriction coefficients (Clerc and Kennedy, 2002) were superior to inertia weights or velocity bounds alone. We therefore used $\varphi_1 = \varphi_2 = 2.05$, and the constriction coefficient χ calculated according to Clerc and Kennedy (2002). We further observed that, although not necessarily required when using inertia weights or constriction coefficients, limiting v_{max} to the dynamic range of the input was advantageous in some cases, as suggested in Eberhart and Shi (2000). The von Neumann and dynamic nearest neighbor topologies showed quite similar performances with the former being chosen due to higher computational efficiency. Different random distributions showed no significant impact in this context. The swarm size was set to 20, a common value in the literature. Larger numbers of particles were unable to improve convergence significantly, but logically increased the number of function evaluations, slowing down the process. Table 2.2 presents the fastest and most reliable setup we obtained. In our applications, this implementation usually converged very quickly to the optimum within 10–15 iterations.

2.2.2.4 Uncertainty Quantification

We considered multiple sources of uncertainties, both in the model and the remote sensing data. These need to be quantified before they can be included in the updating procedure. We are unable to account for potential weather measurement errors or instrument-related issues. Still, we are able to quantify the reaction of AquaCrop-OS canopy cover simulations to perturbations in weather inputs and parameter settings.

We achieved this via Monte Carlo simulations. First, we estimated input-related uncertainty by randomly perturbing a 10-year mean weather time series with Gaussian random noise. The magnitude of the noise was determined by the daily variance observed in the same 10-year period. We obtained 10,000 CC time series from AquaCrop-OS simulations on these randomized weather datasets. Second, we assessed parameter uncertainties accordingly by randomly sampling parameter settings from a normal distribution around the calibrated values in Table 2.1 with a standard deviation of 1/10 of the calibration range. This ensured a sufficient variation within a realistic range around the calibrated settings. We performed 10,000 Monte Carlo simulations randomizing all parameters listed in Table 2.1. Both model-related uncertainties are illustrated in Figure 2.2. Third, we estimated uncertainties in the remote sensing data. Here, the procedures on field- and pixel-level are different. On the field level, we used a set of all CC pixel values in a given field at the observation date; on the pixel-level, we used only values in the 3×3 pixel neighborhood (see Figure 2.3).

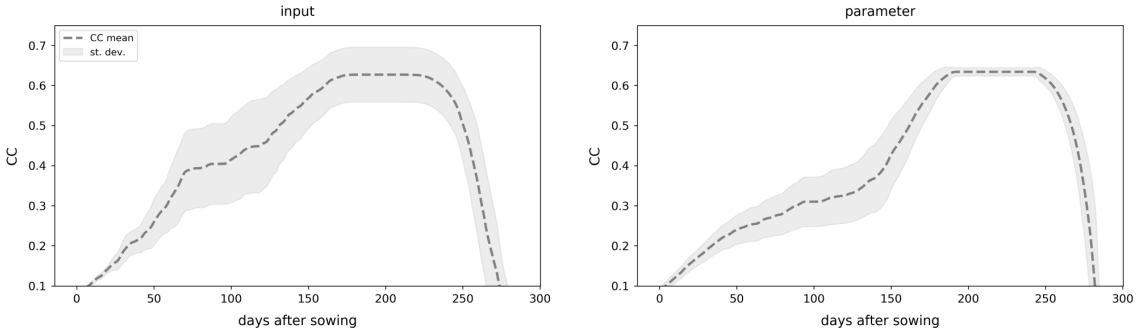

Figure 2.2. Visualization of observed uncertainties throughout the growing season. Uncertainty from Monte Carlo simulations weather inputs (left) and perturbing parameters (right).

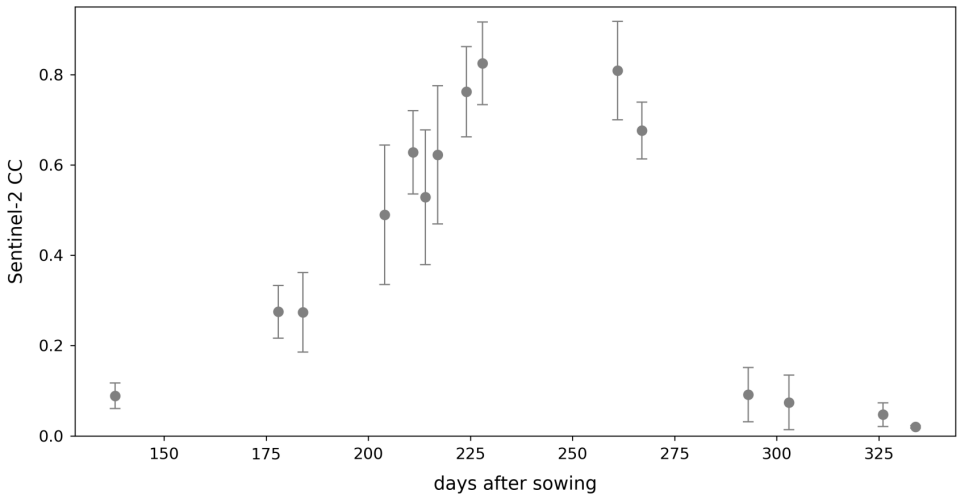

Figure 2.3. Example of uncertainties in Sentinel-2 CC values in a field. Dots refer to the mean of all CC observations (i.e., pixels) in the field, error bars indicate standard deviations.

Using these datasets, we created probability density functions (PDF) representing the probability of all possible CC values (between 0 and 1) of each uncertainty source. We employed kernel density estimation with a symmetric Gaussian kernel. Tests showed that a narrow bandwidth of 0.02 yielded the best results. Finally, we represented the uncertainty in the current simulation by a Gaussian distribution around the currently simulated CC value using a bandwidth of 0.2.

2.2.3 Updating Methodology

2.2.3.1 Simple Updating

The simple updating scheme we employed replaced simulated CC values directly with remote sensing observations without any additional processing and with no consideration of uncertainties. As a result, the simulated CC values remained unconsidered and errors in the remote sensing data were directly transferred to the model.

2.2.3.2 Extended Kalman Filter Updating

Since its first description in Kalman (1960), the Kalman filter has become one of the most common data assimilation techniques (Reichle, 2008). It iteratively updates an estimated value by incorporating information from incoming measured values, taking into account the uncertainty associated with both the measurement and the estimated value. The Kalman filter assumes a linear model. Its extension to non-linear models is the EKF. Here, a linearization of the original non-linear model function is used to update the uncertainty (i.e., the covariance matrix) of the estimate of the model state (Kalman and Bucy, 1961; Grewal and Andrews, 2015).

In our case, we have a non-linear model, but we assimilated the scalar state variable CC directly. Thus, no additional observation is present. Both facts simplify the EKF procedure and make the updating computationally very efficient. Assuming we have the estimates of the state variable x_k and its uncertainty P_k at time step instant t_k. We now obtain a new observation value y_{k+1} at the next time instant t_{k+1}. Then, the EKF performs a predictor step for the model state

$$\hat{x}_{k+1} = f(x_k) \tag{2.10}$$

using the original non-linear model. Additionally, the uncertainty is predicted as:

$$\hat{P}_{k+1} = F_k P_k F_k \tag{2.11}$$

Here, we assume that the model has no error and uses an approximation $F_k \approx f^{'}(x_k)$ for the derivative of the model function. In our case, this derivative is also a scalar. Now, the Kalman gain is computed as:

$$G_{k+1} = \frac{\hat{P}_{k+1}}{\hat{P}_{k+1} + R_{k+1}} \tag{2.12}$$

where R_{k+1} is the uncertainty in the measurement y_{k+1}. Now, the correction step computes new estimates of the state and its uncertainty as:

$$x_{k+1} = \hat{x}_{k+1} + G_{k+1}(y_{k+1} - \hat{x}_{k+1}) \tag{2.13}$$

$$P_{k+1} = (1 - G_{k+1})\hat{P}_{k+1} \tag{2.14}$$

We computed the derivative approximation needed in Equation 2.11 by a finite difference formula:

$$F_k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = \frac{\hat{x}_{k+1} - \hat{x}_k}{x_k - x_{k-1}} \tag{2.15}$$

This approximation uses only already computed quantities. In the first assimilation step ($k = 0$), a modification is needed to replace the value x_{k-1} and $f(x_k - 1)$.

As mentioned previously, after accounting for clouds and cloud shadows, the remaining observations were not too frequent. In case of frequent observations, the updated uncertainty can be propagated continuously throughout the whole EKF process. In our case, however, we often encountered large time gaps in between observations. This implies that the assimilation cannot take place in every time step of the model. Thus, the function f in Equation 2.10 represents not just one model step, but rather summarizes a concatenation of model steps between subsequent time instants t_k and t_{k+1} where measurements are available. As a consequence, the derivative approximation in Equation 2.15 is an average of the derivative of the model in the interval $[t_k, t_{k+1}]$. As indicated in Section 2.2.2.1, AquaCrop-OS varies significantly in its simulation procedures depending on growth stages and environmental influences. Thus, this kind of averaging of the derivative seems to be reasonable. As noted above, the derivative for the first assimilation step has to be approximated in a slightly different way. Here, we used a state at a time instant in the interval $[t = 0, t_1]$ instead of x_{k-1}.

The uncertainty in the state is initially assumed to be 0.2. The uncertainty in the measured values was estimated as the standard deviation of all CC values at the observed location and time (i.e., all pixels of a field on the field-level and pixels in the 3×3 neighborhood on the pixel-level).

2.2.3.3 New Updating Scheme

Figure 2.4 demonstrates the main processing steps for our new method. Preparation of CC data (green), weather data (blue), and yield maps (yellow) were discussed in Section 2.2.1

and uncertainty quantification (dark blue) was covered in Section 2.2.2.4. In this section and the next, we will explain the details of the actual updating process and accuracy assessment (grey).

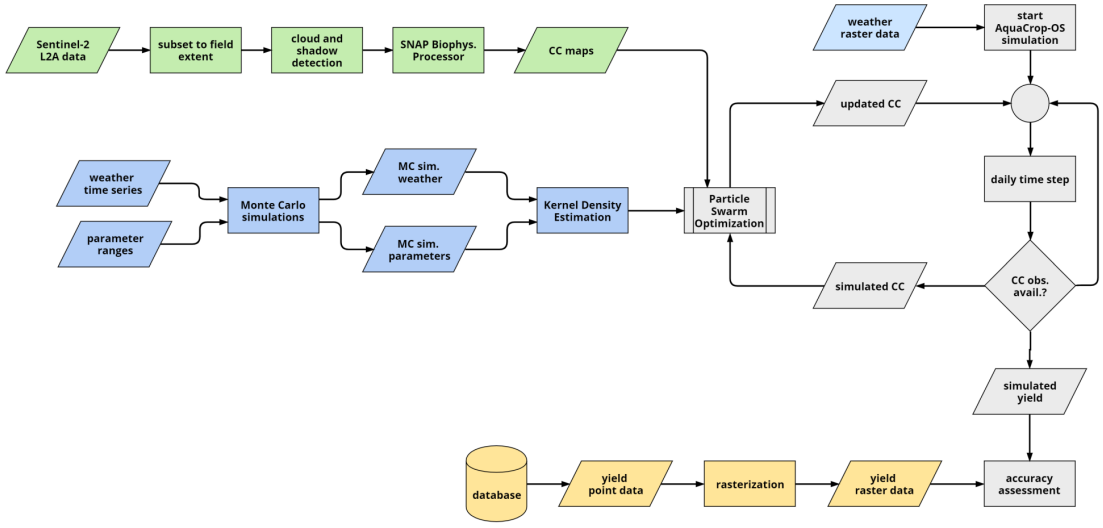

Figure 2.4. Flowchart of pre-processing steps and the new updating algorithm. Colors refer to different aspects: Sentinel-2 CC data (green), weather input data (blue), yield data preparation (yellow), uncertainty quantification (dark blue), and the actual simulation and updating process (grey).

The fundamental idea behind our approach is to balance all uncertainties (relating to the model and the CC observations) to obtain the updated value. To do so, we represented all uncertainties as PDFs (see Section 2.2.2.4). We then obtained a hypothetical optimal Gaussian distribution, as described by a mean μ and a standard deviation σ, that balances all uncertainties in terms of statistical distance (see Figure 2.5). In other words, we assumed that the mean of a distribution minimizing statistical distances to all PDFs will provide us with a better estimate given the available information. We employed PSO to search for the mean and standard deviation of this optimal Gaussian distribution.

The central element of this technique is the representation of distance or similarity between probability distributions or their respective PDFs. There are a number of statistical distance and divergence metrics proposed in the literature. Some of the most common ones are the Hellinger distance, the Kullback–Leibler divergence, and Bhattacharyya distance (Hellinger, 1909; Kullback and Leibler, 1951; Bhattacharyya, 1946).

When comparing calculations measuring the statistical distance of the optimal Gaussian distribution to a set of uncertainty PDFs, the different metrics behaved similarly. Figure 2.6 demonstrates this with example cases where three PDFs at different locations and with different standard deviations are considered. The values shown for the different

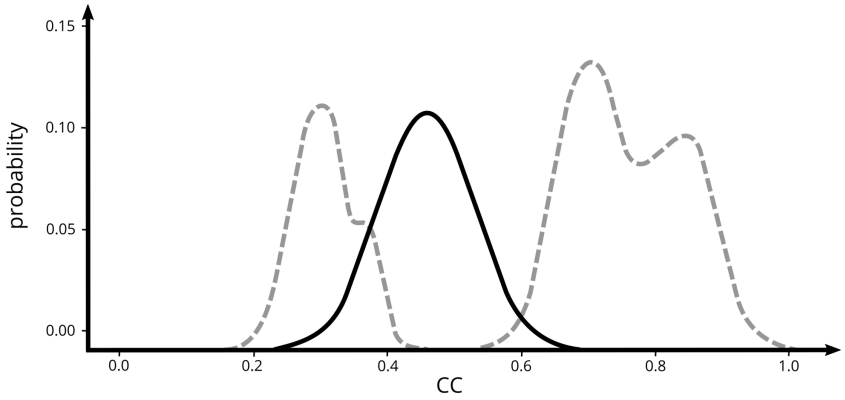

Figure 2.5. Idealized representation of optimal Gaussian distribution (solid dark grey line) and summed uncertainty probability density functions (solid light grey lines). Maximum likelihood estimators (MLE) of the uncertainty probability density functions indicated by vertical dotted lines, updated CC value (i.e., location of optimal Gaussian distribution) indicated by vertical dashed line. This example represents a common case in our tests with model-related uncertainties and remote sensing observations indicating different value ranges for optimal CC.

distance metrics were obtained using a brute force algorithm moving an optimal distribution of standard deviation 0.05 from 0.01 to 0.99 through the search space. These demonstrative cases are, however, drastically simplified as they assume all PDFs to be perfectly symmetric Gaussian distributions and keep the standard deviation of the optimal distribution fixed. Additionally, this example case only assumes three PDFs while the situation logically becomes much more heterogeneous when more are considered.

Figure 2.6a,b show that in cases where the PDFs are relatively far apart, the three distance metrics behave similarly. Although magnitudes may differ significantly, the general shapes (number and location of local optima) are quite similar. If we consider the case where the three PDFs are located closely to one another, however, problems arise. Here, as illustrated in Figure 2.6c, all three distances failed to establish a clear minimum within the search range and instead produced single peaks or plateaus. This produced a situation with two potential minimum solutions at the extremes. In searching for the minimum value, the algorithm would run off to either side. In this special case of only Gaussian normal distributions, this is equivalent to choosing either the upper or the lower bound as the updated value. This may be mitigated, as attempted in a previous implementation (Wagner et al., 2019), by using the mean squared distance of the Maximum Likelihood Estimator (MLE) of the optimal distribution to the MLEs of all uncertainty PDFs. This approach, however, has two major drawbacks. First, using the MLE as an indicator for the position of a distribution is only representative if it is a unimodal, approximately normal distribution. Second, it necessitates an additional processing step to determine the MLEs. Although obtaining MLEs is a straightforward optimization problem (maximizing the sum

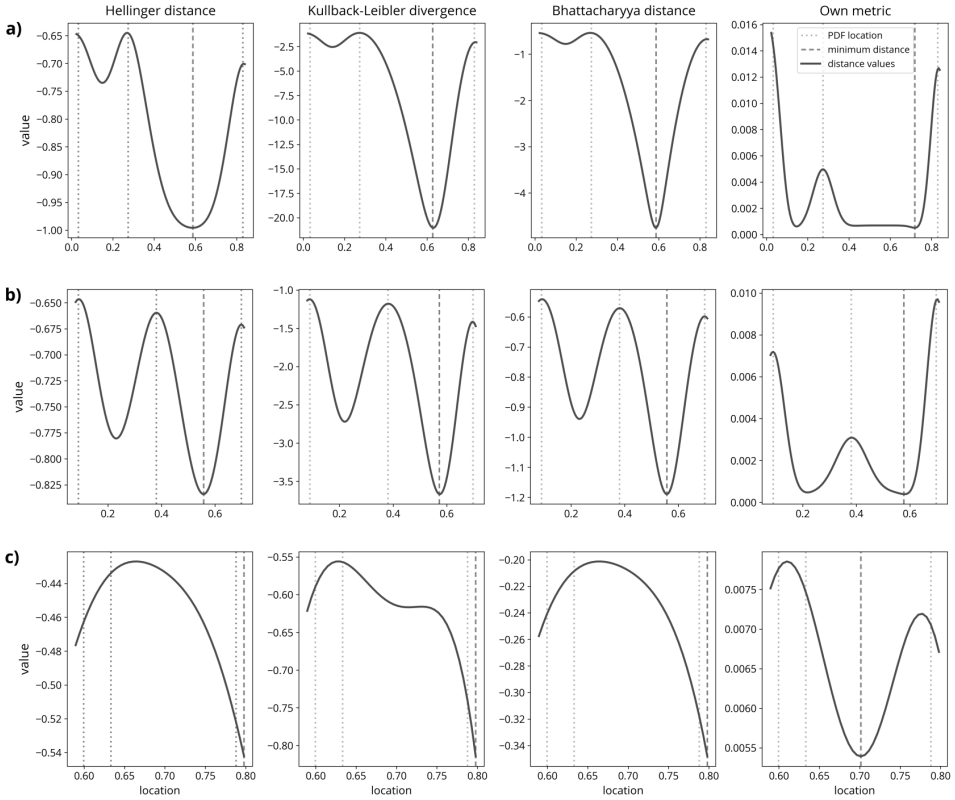

Figure 2.6. Three simplified example cases (a–c) of Hellinger distance, Kullback–Leibler divergence, Bhattacharyya distance, and the distance metric of 2.16 used in this study (scales on the first three inverted). Solid black lines represent the distance of an optimal distribution moving through the search space according to three uncertainty PDFs (locations indicated by dotted grey vertical lines). Vertical dashed lines indicate the minimum value the optimizer would obtain.

of probabilities in all PDFs), it adds to the processing time and introduces a potential error source. Due to this drawback, we decided to employ a different metric described as:

$$D(f_{opt}(\mu, \sigma), f_{sum}) = \sum_{i=1}^{N} \left(q_i \cdot \frac{(p_i - \mu)}{\sigma} \right)^2 \tag{2.16}$$

where $f_{opt}(\mu, \sigma)$ and f_{sum} are the optimal Gaussian distribution defined by μ and σ and the sum of all uncertainty PDFs, respectively. Furthermore, q_i is the probability value of the summed uncertainty PDFs. Although this definition violates at least two important criteria of a metric as it is neither symmetric nor is it limited to the range (0, 1), we may still refer to it as such for simplicity. Fundamentally, this metric weights the probabilities of the uncertainty PDFs based on their distance to the optimal distribution measured in standard deviations (z-scores). As shown in Figure 2.6, using this metric, we can ensure a

clear minimum within the search space, even in the case of PDFs located very closely to one another.

Additionally, we addressed the possibility that not all PDFs may have the same relevance for the update. We therefore introduced a weighting, following the assumption that PDFs that are more similar to the currently simulated CC value of the model contain less new information than those that were more dissimilar. To represent this, we employed the Hellinger distance to introduce a weighting of the individual PDFs. We obtained weights by calculating the Hellinger distance for each uncertainty PDF to a narrow Gaussian distribution around the currently simulated CC value, following the equation:

$$H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{N}(\sqrt{p_i} - \sqrt{q_i})^2} \tag{2.17}$$

where P and Q are two distributions with p_i and q_i describing the probabilities of the two distributions at point i. The Hellinger distance ranges between 0 (identical) to 1 (no overlap). We used the resulting distance values to establish normalized exponential weights:

$$w_i = \frac{exp(\alpha H(f_i, f_{sim}))}{\sum_{j=0}^{n} exp(\alpha H(f_j, f_{sim}))} \tag{2.18}$$

where f_i is the PDF representing the respective uncertainty and f_{sim} is the distribution around the simulated CC value. The denominator represents the sum of all Hellinger distances, ensuring summation to unity. This further allows us to introduce α, a simple multiplicative factor that determines the magnitude of weights with higher values of α leading to more emphasis on dissimilar PDFs (i.e., those with larger Hellinger distance). Combining Equation 2.16 with the weights determined in Equation 2.18 results in an optimization problem of the form:

$$min \sum_{i=1}^{N} (D(f_{opt}(\mu, \sigma), f_{sum}) \cdot w_i) \tag{2.19}$$

This objective function, however, may lead to the process of finding an optimum with a very large standard deviation σ. Although such an extremely flat distribution would indeed minimize statistical distances, it is not a useful solution for our approach. If the distribution is essentially flat, the mean value may be positioned anywhere in the CC range without having any significant effect on statistical distance. In other words, a flat distribution would allow for any CC value to be an optimal solution. To avoid this,

we penalized σ of the optimal Gaussian distribution. This leads to our final optimization problem:

$$min \sum_{i=1}^{N} (D(f_{opt}(\mu, \sigma), f_{sum}) \cdot w_i) + \sigma \tag{2.20}$$

We then searched for optimal values of μ and σ to minimize Equation 2.20. A drawback in terms of computation, however, is that values around the optimum tend to be generally very small and the minimum is not as distinct in cases where PDFs are far apart, like in Figure 2.6a,b. This puts a larger emphasis on the proper settings and reliability of the optimizer used. The theoretically unbounded, but in our observations generally small value range of this metric, however, facilitated the introduction of constraints compared to, for example, the Kullback–Leibler divergence with observed values between 0.02 and over 30.

2.2.3.4 Performance Analysis

The comparison of updating schemes involved three test situations: field-level, pixel-level, and pixel-to-field aggregated level estimations where we simulated yield on a pixel basis and compared the mean of these individual estimates to the observed mean yield of the field. We performed this analysis on the pixel-level validation dataset and therefore did not use all pixels. Still, the large size of 40% of all pixels ensured a representative sample of pixels for all fields. As stated earlier, we did not have in situ measurements of canopy cover or regular samples of biomass in the field. It was therefore not our goal to obtain realistic CC time series or closely recreate biomass development. Instead, our comparison relies primarily on the capacity for improving yield predictions.

We considered two versions of our method: one with a fixed (user-defined) value for the weighting factor α and an adaptive one letting PSO automatically determine α in the process. Previous tests on the calibration data showed quite high values of α around 5–10 were advantageous for field-level simulations, while on the pixel-level, values around 1–2 were preferable. In our comparison, we compared these settings with the results obtained by the adaptive weighting within the continuous range 1–10. We also evaluated the capability of our method to incorporate different uncertainties by adding uncertainty sources one at a time and observing the effect on performance. First, we introduced remote sensing data into the update, then parameter-related and, finally, weather-related uncertainty.

We used three metrics to analyze the results. The main metric for overall performance in the results was the RMSE.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2.21}$$

where y_i describes the reference yield value and $\hat{y}_i$ is the predicted value. To determine bias in the results, we used the mean percentage error (MPE).

$$MPE = 100 \cdot \frac{1}{N}\sum_{i=1}^{N}\frac{\hat{y}_i - y_i}{y_i} \tag{2.22}$$

Furthermore, we calculated the R^2 scores.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{2.23}$$

where $\bar{y}$ is the mean of the reference data. To incorporate the inherent uncertainty that is present in any sort of yield measurement, we also employed a percentage match metric (i.e., counting the predicted values that fall within a certain range) around the respective reference value (in this case $+/-20\%$).

$$pmatch = 100 \cdot \frac{1}{N}\begin{cases} |y_i - \hat{y}_i| \leq 0.2 \cdot y_i & 1 \\ else & 0 \end{cases} \tag{2.24}$$

2.3 Results

In this section, we describe the performance of yield predictions of AquaCrop-OS on the field-level, pixel-level, and pixel-to-field aggregated level. The very low values of R^2 we observed throughout all experiments did not benefit the interpretation and were therefore omitted from descriptions. We analyze and address this topic specifically in Section 2.3.4 and the discussion (Section 2.4).

2.3.1 Field-Level Yield Estimation

The results in Table 2.3 show that without assimilation, AquaCrop-OS produced rather poor results with a RMSE of 1.32 t/ha and a quite significant bias, expressed in an MPE of 15.2%, which indicates a tendency of the model to overestimate yield. The simple updating scheme had no effect in terms of accuracy, but inverted the bias to an underestimation.

The EKF performed better with a significant reduction of RMSE to 1.20 t/ha and a slightly smaller bias.

Table 2.3. Accuracies of the AquaCrop-OS field-level yield estimation on the validation dataset using different updating schemes. Uncertainties refer to the PDFs considered. RS: remote sensing; pars: parameter-related; weather: weather-related.

Updating	**α**	**Uncertainties**	**RMSE [t/ha]**	**MPE [%]**	**R^2**	**Pmatch [%]**
none	-	-	1.32	15.2	0.01	70.0
simple	-	-	1.37	−15.6	0.09	70.0
EKF	-	-	1.20	−14.8	0.35	70.0
new method	5	RS	1.14	−10.5	0.08	85.0
new method	5	RS, pars	1.09	4.3	0.04	75.0
new method	5	RS, pars, weather	1.04	5.7	0.06	75.0
new method	10	RS	1.23	−7.5	0.13	80.0
new method	10	RS, pars	1.11	4.20	0.10	85.0
new method	10	RS, pars, weather	1.05	4.0	0.11	80.0
new method	auto	RS	1.25	−8.4	0.12	75.0
new method	auto	RS, pars	1.09	4.5	0.15	75.0
new method	auto	RS, pars, weather	0.90	5.90	0.21	80.0

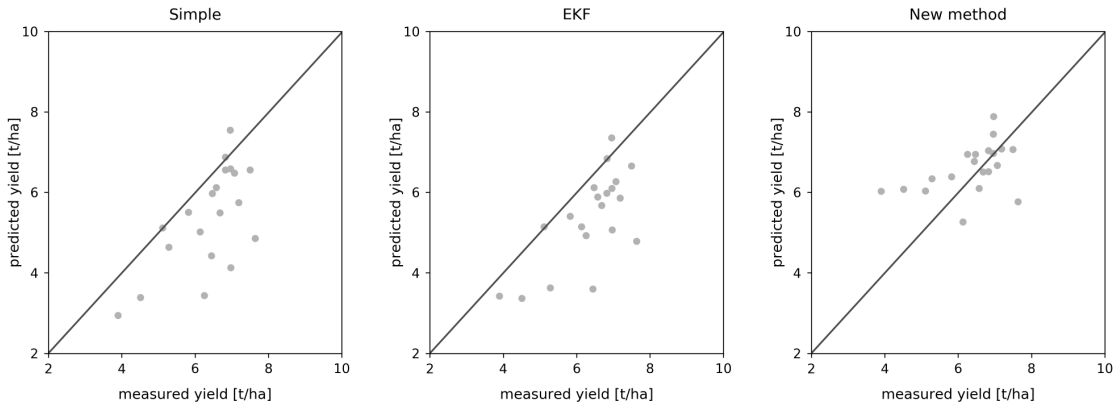

Figure 2.7. Scatter plots of measured vs. predicted yield on the field-level for (left to right) simple update, EKF update, and the new method (adaptive; all three uncertainties).

Our new PSO method performed similar to or better than the EKF with the two α values of 5 and 10 when using all three uncertainties. Using only remote sensing inputs led to small changes in RMSE, but inverted the bias from positive to negative in all setups, similar to what we observed in the simple and EKF updates. Adding parameter-related uncertainties led to an improvement in both RMSE and MPE, while the subsequent addition of weather-related input affected results only slightly, and sometimes slightly deteriorated MPE. The adaptive version performed comparably and even outperformed

the non-adaptive ones occasionally. Overall, when using all three uncertainties, model results improved by up to 0.42 t/ha in the RMSE, 11.2% in MPE, and 15% in terms of pmatch. All versions of our approach were particularly successful in reducing bias. This is also apparent in the scatter plot in Figure 2.7. It also indicates a tendency of the new method to reduce the range of predictions by avoiding low results < 5 t/ha.

2.3.2 Pixel-Level Yield Estimation

Table 2.4 shows that on the pixel-level, the model with no updating again produced a high RMSE and showed a bias of 12.7%. Both simple updating and the EKF inverted the bias to an underestimation of -14.9 % and -13.3 %, respectively. Interestingly, both techniques increased the inaccuracies.

Table 2.4. Accuracies of AquaCrop-OS pixel-level yield estimation on the validation dataset using different updating schemes. Abbreviations as in Table 2.3.

Updating	$\boldsymbol{\alpha}$	**Uncertainties**	**RMSE** [t/ha]	**MPE** [%]	$\mathbf{R^2}$	**Pmatch** [%]
none	-	-	1.52	12.7	0.08	65.6
simple	-	-	1.88	−14.9	0.07	43.8
EKF	-	-	1.79	−13.3	0.08	45.6
new method	1	RS	1.80	−12.9	0.06	43.6
new method	1	RS, pars	1.43	1.8	0.07	64.4
new method	1	RS, pars, weather	1.47	2.5	0.07	64.9
new method	2	RS	1.72	−12.7	0.05	49.0
new method	2	RS, pars	1.47	0.2	0.06	61.3
new method	2	RS, pars, weather	1.50	3.2	0.07	64.9
new method	auto	RS	1.82	−12.8	0.05	46.2
new method	auto	RS, pars	1.52	−2.9	0.07	59.7
new method	auto	RS, pars, weather	1.48	1.1	0.09	62.1

The new method performed better when using remote sensing and parameter-related uncertainties, while adding weather-related uncertainty did not improve the results consistently. Still, even the best results only reduced the RMSE by about 0.09 t/ha. Despite that, it again managed to reduce the bias significantly. The adaptive version seemed to incorporate the different uncertainties more consistently, as indicated by decreasing RMSE and MPE with each added uncertainty. The scatter plot in Figure 2.8 supports these findings. As previously mentioned, the new method reduced the bias through slightly higher predictions with a slightly smaller range.

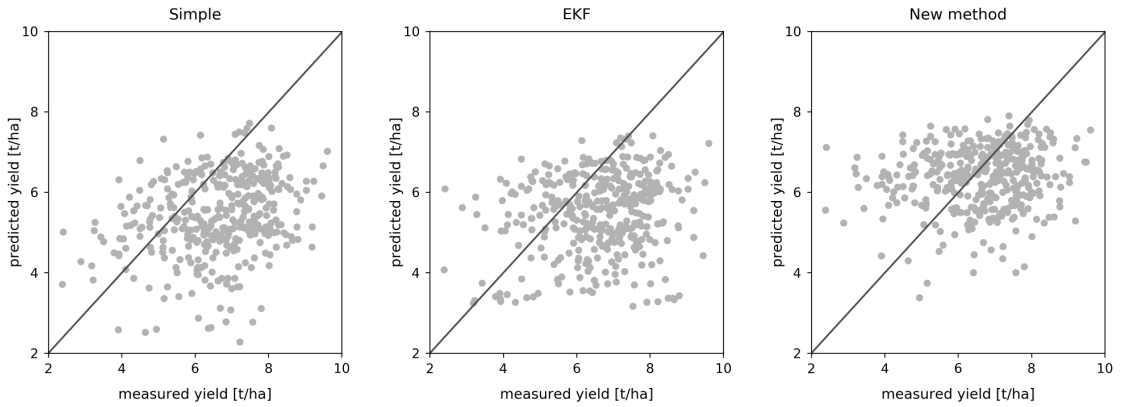

Figure 2.8. Scatter plots of measured vs. predicted yield on the pixel-level for (left to right) simple update, EKF update, and the new method (adaptive; all three uncertainties). To improve visual interpretation, we displayed only a subset of 400 data points.

2.3.3 Pixel-To-Field Aggregated Yield Estimation

Table 2.5. Accuracies of AquaCrop-OS yield estimation on the validation dataset aggregating pixel-level simulations to the field-level using different updating schemes. Abbreviations as in Table 2.3.

Updating	α	**Uncertainties**	**RMSE** [t/ha]	**MPE** [%]	**R²**	**Pmatch** [%]
none	-	-	1.12	11.6	0.09	72.2
simple	-	-	1.36	−12.5	0.11	69.4
EKF	-	-	1.33	−13.3	0.09	61.1
new method	1	RS	1.27	−10.9	0.10	72.2
new method	1	RS, pars	0.92	1.6	0.11	86.1
new method	1	RS, pars, weather	0.96	3.2	0.11	80.6
new method	2	RS	1.26	−10.5	0.10	72.2
new method	2	RS, pars	0.95	0.6	0.05	83.3
new method	2	RS, pars, weather	0.97	1.3	0.07	80.6
new method	auto	RS	1.28	−10.4	0.08	69.4
new method	auto	RS, pars	0.96	−2.4	0.09	86.1
new method	auto	RS, pars, weather	0.92	1.5	0.07	86.1

In general, the aggregated results (Table 2.5) are better than those on the pixel-level, indicating a benefit from aggregation. When compared to field-level results, however, differences become apparent. Without an update, the model produced better results on the aggregated than on the field-level, while the simple update showed no significant difference. The EKF, however, performed worse on aggregated scales than on field-wise runs, indicated by a higher RMSE. The new method, in comparison, provides similar

or better results on the aggregated compared to the field-level and bias tended to be a bit smaller. Results for the different uncertainty setups generally behaved in accordance to the observations in previous sections. Compared to the model without updating, we observed only small improvements for all updating schemes. Again, the scatter plot in Figure 2.9 shows a smaller range, but also a reduction in bias in the predictions using the new method compared to simple updating and EKF.

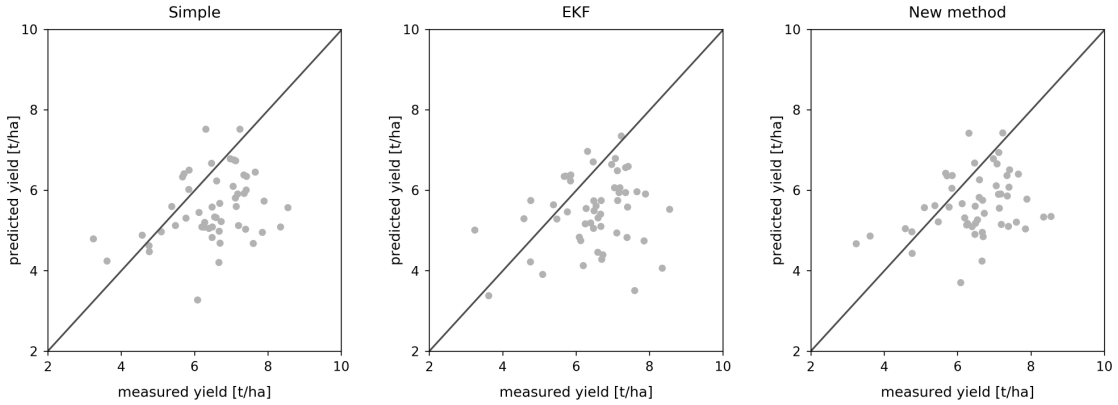

Figure 2.9. Scatter plots of measured vs. predicted yield on the pixel-to-field aggregated level for (left to right) simple update, EKF update, and the new method (adaptive; all three uncertainties).

2.3.4 R^2 Performance

As mentioned previously, we generally observed low R^2 values in all yield estimations. This is particularly surprising, since even good RMSE and MPE values were associated with low R^2 values. The scatter plots in previous sections also indicate low correlations between the predicted and measured yield values.

The time series plot in Figure 2.11 may hint at causes behind different behavior of the updating techniques as well as the Sentinel-2 CC observations. As mentioned earlier, CC observations were often very different from what the AquaCrop-OS model simulated by default. In this example, observations were much lower than the CC predicted by the model (without update), especially in the earlier growth stages. The EKF updates were therefore particularly low in many cases, while the new method often performed less dramatic corrections.

An explanation could be a bias toward drastic underestimation in some cases and overestimation in others. This kind of bias would not manifest itself in a metric like MPE, but may become obvious when looking at outputs individually. We therefore investigated the results of the fields with the worst predictions. Figure 2.10 shows an example for extreme over- and underestimations in one of the fields. Overall results using the EKF

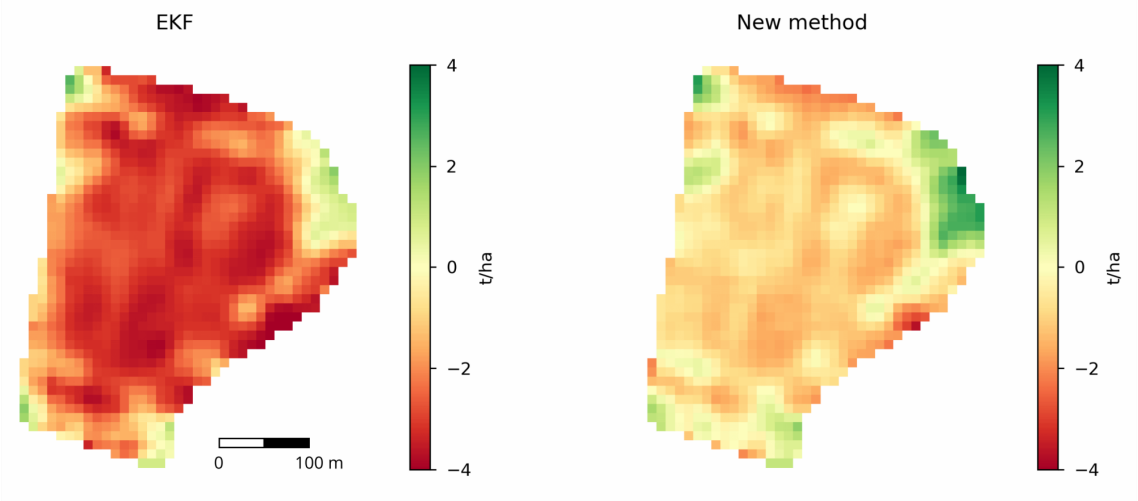

Figure 2.10. Example map of the pixel-wise differences between the simulated and in situ yield values (< 0 underestimation, > 0 overestimation) for EKF updating (left) and adaptive PSO updating (right). Maps are smoothed with a 3×3 mean filter to reduce noise.

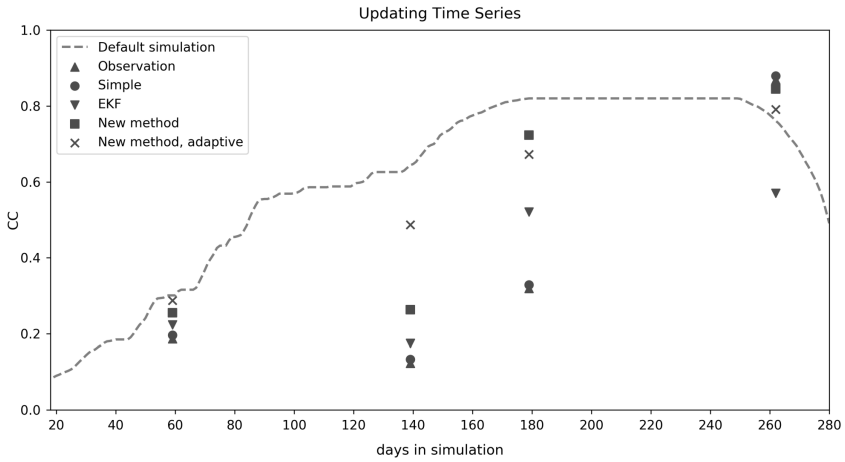

Figure 2.11. Example for time series of CC updating on the field-level showing the default simulation without updating and the corresponding updated CC values from simple updating (equivalent to observed CC), EKF updating, and the new method using $\alpha = 5$ and the adaptive version.

update had a more obvious bias towards underestimations (≤ -3 t/ha). The new method also produced significant errors, but they tended to be slightly more evenly distributed and less dramatic in most cases. Nevertheless, both results demonstrate similar trends with regions of overestimation near the top left and right as well as the bottom left boundaries in Figure 2.10. The remaining parts were mostly underestimated, in particular near the center and at the right border of the field. These issues may be explained by mixed pixels and often observed low yields near the boundaries of the field that are not captured by the model.

2.4 Discussion

In this study, we introduced a new updating method based on PSO to simulate wheat yield using the AquaCrop-OS model. As mentioned previously, our comparison addressed the results of different updating schemes applied to the same yield prediction scenario. We compared performances on different scales.

Overall, the new method performed better than a simple update and similar or better than the EKF update approach. It was particularly successful in reducing the bias in the estimation and outperformed both the simple update and the EKF update in this respect. However, the EKF is designed to correct random errors in the model state rather than systematic errors, so state augmentation or bias correction may improve its performance in this respect (De Lannoy et al., 2007). Furthermore, other techniques such as the ensemble Kalman filter or particle filter updates may better address non-linearities in the model.

Settings chosen for the update largely determined performance. The weighting factor α had a significant impact and the optimal value was scale-dependent. This would pose the need for prior testing and calibration by the user, which is not ideal for most applications. Further research may reveal a best practice for the setting of α, which may depend on the number and/or type of uncertainty PDFs or scale. The adaptive version, however, showed promising results via self-adjusting α. It performed comparably to other settings on all scales, even though in most cases, it was not the best-performing updating scheme. Instead of guidance for the manual adjusting of α, it may be preferable to improve the way is involved in the objective function. Another observation in this context is that higher α (values of five and higher) seemed to be advantageous on the field-level. One may argue that remote sensing observations are superior to the simulation and therefore a higher weighting automatically led to a significant improvement in predictions. This interpretation, however, contrasts with the poor results obtained when using remote sensing-related uncertainty alone or in the simple update. It is more likely that the reason lies in model-related uncertainties, which often are distributed (skewed) normally around or close to the simulated CC value, while remote sensing observations differed significantly. Therefore, a larger weight on remote sensing-related uncertainty is required to "outweigh" the two similar model-related distributions. This interpretation is further supported by the fact that the field-level required higher α values than the pixel-level. In our pixel-level implementation, we used the same model-related uncertainty PDFs (based on 10,000 simulations each), but considered only the immediate neighborhood of the pixel. Naturally, heterogeneity in such a small, spatially constrained sample is much smaller. Therefore, the distribution of the remote sensing-related uncertainties in the pixel neighborhood tends to be narrower than that of the entire field, consequentially leading to a smaller probability of an overlapping with the current simulated CC of AquaCrop-OS. The result is

that most remote sensing predictions are assigned a Hellinger distance of 1 and therefore receive higher weights than on the field-level, even with a comparatively small α. This situation may be addressed by altering the kernel bandwidth or analyzing different scales, numbers of pixels, etc. to possibly observe a relationship between the number of PDFs or sample size and the optimal α values to choose. The other approach may be the use of an adaptive version that seemed to adjust to the different levels quite effectively, which was demonstrated in its comparable performance to the fixed versions.

Regarding uncertainties, all versions of the new method managed to incorporate different PDFs successfully. Using only remote sensing uncertainty led to poor results, probably because of the way the weighting was handled: in case of having only one uncertainty input, that distribution will always be assigned a weight of 1 (see Equation 2.18). Results are expected to be close to those of a simple replacement update because the remote sensing-related PDF is the only one considered. An obtained optimal distribution would then obviously be located closely to the observed pixel or field mean CC value. In case of an actual normal distribution of values, it must be identical.

As expected, adding model-related uncertainty always improved results, except for adding the weather PDF, which was unable to enhance the performance consistently. A possible reason could be the similarity of the two distributions as indicated in Figure 2.2. Adding parameter-specific uncertainty PDFs rather than an all-in-one PDF may prove a better alternative in the future. This also highlights the importance of choosing the relevant type of uncertainty and its correct quantification. However, further research may be necessary to ensure a proper incorporation of all PDFs. This task would be closely linked to the improvement in optimizing α, but may also involve increasing the 3×3 pixel neighborhood with a larger rectangular or circular one.

Another observation was that R^2 values were generally poor throughout all analyses, methods, and scales. Our investigation revealed that many fields contained regions of significant over- and underestimations. These frequent outliers may cause the low R^2 values, even in cases with relatively good RMSE and MPE values. The new method seemed to be less likely to underestimate yield than EKF updating, but showed similar overall trends.

In this context, we also cannot exclude the possibility of significant bias in the Sentinel-2 CC data itself. The low errors mentioned in the original report on the algorithm suggest good performance (Weiss and Baret, 2016). However, the datasets used for both training and validation of the artificial neural networks in the algorithm were obtained from radiative transfer simulations rather than in situ experimental data may therefore not be representative for all situations. The fact that low R^2 values were present in all updating results further indicates that they are probably unrelated to the updating

methods themselves, but are connected to issues in the CC inputs or in situ yield data. This is further supported by the facts that pixel-level predictions were much worse and the simple update performed particularly poorly. Data on the pixel-level may therefore be unreliable. By increasing the detail in the process (i.e., the number of individual simulations per field), the broader trend is captured, but individual pixel-wise yield predictions are not matched. Nevertheless, as no external in situ measurements were available in our study area, we could not validate the quality conclusively.

Finally, AquaCrop-OS may also introduce unknown uncertainties. Even with careful calibration, it is difficult to scale such a model to represent conditions in a number of fields distributed over a large area, let alone different spatial scales. On top of performance-related questions, pre-processing requirements are important for applications. The approach in the implementation described here requires extensive pre-processing. Our technique, however, is flexible regarding the type of uncertainty representation, future applications would not need to rely on a computation-intensive approach like Monte Carlo simulations and kernel density estimation. Instead, if previous knowledge about the characteristics of the model is available, simple functional relationships may represent uncertainties, for example.

2.5 Conclusions

We presented a method for updating model variables during simulation, taking into account uncertainties in both the model and the assimilated data. We described the method for assimilating remote sensing data into a dynamic crop model for improving yield estimation. The method proved to be comparable to other existing updating techniques, but was particularly capable of reducing bias in the estimations and managed to incorporate different sources of uncertainties.

We described the process specifically for the application in our study. The principle is, however, easily transferrable to other models or model variables. Its flexibility regarding the representation of uncertainties would also allow an adaptation to different situations where Monte Carlo simulations may not be feasible. Previous knowledge about the model in question would allow a representation of uncertainties by a simple functional relationship or a set of distributions.

Further research may be needed to analyze the behavior of the technique regarding different numbers of uncertainties and potential improvement in incorporating them into the updating scheme. Additionally, improvements to the distance metrics, objective function, and weighting may also be of interest as well as the applicability to different remote sensing datasets and pixel neighborhood sizes. It is also possible to analyze improve-

ments regarding the optimal distribution, for example, by adding skewness or kurtosis. Furthermore, additional comparisons to other non-linear updating techniques such as the ensemble Kalman filter may provide valuable insights.

Acknowledgments

We thank the German Weather Service (DWD) for delivering the station and raster weather datasets. We further thank the anonymous farmers that provided in situ yield data.

Chapter 3

Extracting Agricultural Fields from Remote Sensing Imagery Using Graph-Based Growing Contours

M. P. Wagner, N. Oppelt
Remote Sensing, 2020, 12(7), 1205
Special Issue "Remote Sensing for Crop Mapping"
DOI: 10.3390/rs12071205

Received: 18 March 2020; Accepted: 6 April 2020; Published: 8 April 2020

Abstract

Knowledge of the location and extent of agricultural fields is required for many applications, including agricultural statistics, environmental monitoring, and administrative policies. Furthermore, many mapping applications, such as object-based classification, crop type distinction, or large-scale yield prediction benefit significantly from the accurate delineation of fields. Still, most existing field maps and observation systems rely on historic administrative maps or labor-intensive field campaigns. These are often expensive to maintain and quickly become outdated, especially in regions of frequently changing agricultural patterns. However, exploiting openly available remote sensing imagery (e.g., from the European Union's Copernicus programme) may allow for frequent and efficient field mapping with minimal human interaction. We present a new approach to extracting agri-

cultural fields at the sub-pixel level. It consists of boundary detection and a field polygon extraction step based on a newly developed, modified version of the growing snakes active contours model we refer to as graph-based growing contours. This technique is capable of extracting complex networks of boundaries present in agricultural landscapes, and is largely automatic with little supervision required. The whole detection and extraction process is designed to work independently of sensor type, resolution, or wavelength. As a test case, we applied the method to two regions of interest in a study area in the northern Germany using multi-temporal Sentinel-2 imagery. Extracted fields were compared visually and quantitatively to ground reference data. The technique proved reliable in producing polygons closely matching reference data, both in terms of boundary location and statistical proxies such as median field size and total acreage.

3.1 Introduction

Throughout the second half of the 20th century, the growing demand for agricultural products was largely met by either an expansion of agricultural lands or an intensified production. Both may cause environmental concerns (Graesser and Ramankutty, 2017; Tilman et al., 2001). While cropland expansion slowed down due to the limited available land and improving technology, the intensification of agricultural production went along with increasing emissions, negative effects on soil and groundwater, and changing land use patterns (Graesser and Ramankutty, 2017; Tilman et al., 2001; Barrett et al., 2001).

Therefore, the timely and accurate monitoring of agricultural landscapes and production systems is crucial. In national cadasters such as the Land Parcel Identification Systems or the Integrated Administration and Control Systems promoted by the European Union, land use changes have been addressed for a long time (García-Pedrero et al., 2017; European Court of Auditors, 2016; Sagris and Devos, 2008; Inan et al., 2010). These systems may offer valuable and accurate datasets, but they often rely on historic administrative maps, manual field delineations based on satellite or airborne imagery, and in-situ mappings via Global Positioning Systems (GPS) tracking (García-Pedrero et al., 2017). However, manual approaches are time-consuming, costly, and often subjective, and therefore of limited use in regions with frequently changing field structures or cropping patterns (García-Pedrero et al., 2017; Tiwari et al., 2009; Rahman et al., 2019; Da Costa et al., 2007).

The accurate mapping of the location, size, and shape of individual fields is important for the implementation of precision agriculture, crop yield estimations, resource planning, and environmental impact analysis (García-Pedrero et al., 2017; Rahman et al., 2019; Turker and Kok, 2013). It enables insights into the degree of mechanization, agricultural practices, and production efficiency in a region, and allows better implementation

of administrative policies such as subsidy payments and insurance (García-Pedrero et al., 2017; Yan and Roy, 2016). Additionally, extracting individual fields is a crucial preliminary step for many analytical and mapping applications, including the monitoring of field management and calculating agricultural statistics (Graesser and Ramankutty, 2017; García-Pedrero et al., 2017; Turker and Kok, 2013). Especially crop type and land use mapping benefit from knowledge of precise field locations, as object-based classification techniques usually outperform pixel-based ones (Turker and Kok, 2013; Rydberg and Borgefors, 2001; Watkins and van Niekerk, 2019).

The timely and accurate land use monitoring of agricultural landscapes and production systems is therefore crucial, especially in highly fragmented landscapes with high spatial heterogeneity due to diversity in sizes, shapes, and crops and a high temporal heterogeneity due to changes of field patterns, management practices, and crop rotation (García-Pedrero et al., 2017; Gonzalo-Martín et al., 2016).

Given the increasing availability of satellite remote sensing data, Earth observation has the capacity to aid in the setup of cost-effective automatized mapping systems. Agricultural monitoring in general has a long history in remote sensing, dating back to the original Landsat mission and the Large Area Crop Inventory Experiment (LACIE) (Yan and Roy, 2016; MacDonald and Hall, 1980). In recent decades, various applications have been explored, including large-scale yield prediction, land use mapping, crop type classifications, plant health monitoring, and precision farming (Atzberger, 2013; Mulla, 2013; Dorigo et al., 2007). With the increasing availability of remote sensing data through open data policies, agricultural services and automatized parcel-level monitoring are becoming more prevalent (Barrett et al., 2001; Borgogno Mondino and Corvino, 2019). Still, remote-sensing-based boundary detection and field extraction have seen relatively little research interest in comparison, and robust techniques for automatic field delineation are still rare (Watkins and van Niekerk, 2019; Yan and Roy, 2014).

Many previous field extraction approaches are concerned with the location and extent of fields within the image rather than extracting precise polygons. Examples include the use of edge detection and morphological decomposition to segment fields in Landsat imagery or training random forest classifiers on image features such as mean reflectance values (Yan and Roy, 2014; Debats et al., 2016). Rahman et al. (2019) followed a different approach by attempting to delineate fields using statistical data on crop rotation patterns. Other studies intended to extract field boundaries. For example, García-Pedrero et al. (2017) trained a classifier to determine delineation between adjacent superpixels near potential field boundaries. Tiwari et al. (2009) used color and texture information to first segment the image using fuzzy logic rules and then refined the field boundaries using snakes. Turker and Kok (2013) attempted to detect sub-boundaries within known fields via perceptual grouping using Gestalt laws. Recently, Watkins and van Niekerk (2019)

compared multiple edge detection kernels as well as watershed, multi-threshold, and multi-resolution segmentations in their capacity for field boundary detection. Their results indicated that the combination of the Canny edge detector and the watershed segmentation algorithm produced the best results. While most of these studies rely on high-resolution satellite data, da Da Costa et al. (2007) explored the potential to delineate vine parcels in very-high-resolution data from unmanned aerial vehicles. They used texture metrics and specific patterns in the vine fields to extract the precise locations of field boundaries. Recently, the use of convolutional neural networks was explored in Masoud et al. (2019).

In this study, we present a new approach to agricultural field extraction that allows for both the detection of boundaries and the automatic extraction of individual fields as polygons on a sub-pixel level. We merge image enhancement and edge detection techniques with a new modified active contour model and graph theory principles to obtain smoother field boundaries than possible in regular image segmentation alone. Further, the new approach provides a largely unsupervised contour extraction tracking the spatial relationships and interconnectivity of a complex boundary set and makes it possible to produce an interconnected network of contours rather than a set of separate segments.

As natural boundaries like hedges are easier to detect than differences due to crop types, field boundaries are often difficult to separate depending on the current growth stage or types of crops (Rydberg and Borgefors, 2001). Following previous studies, we therefore combine multi-temporal remote sensing observations to enhance boundaries that may only become visible during certain growth stages or management steps. The structure of the methodology can be regarded as two-fold, consisting of the pre-processing and edge detection steps on one hand and separate field boundary detection and polygon extraction steps on the other.

The structure of this paper is as follows: Section 3.2 describes the study area and the data we used. Section 3.3 introduces the methodology, while Section 3.4 and Section 3.5 present and discuss the results for our study area. Lastly, Section 3.6 summarizes the most important findings and provides a brief outlook on further research.

3.2 Data and Materials

3.2.1 Study Area

Our study area is located in northern Germany in the state of Schleswig-Holstein. The landscape is hilly but with low elevations. The climate of the region is temperate/oceanic (warm summers, wet winters; Cfb in Koeppen-Geiger climate classification) (Kottek et al., 2006). Soils are predominantly para brown earths, podzol-brown earths, and pseudogleys

originating from glacial deposits (Bundesanstalt für Geowissenschaften und Rohstoffe, 2018).

The region is dominated by agricultural use. It is rather heterogeneous with a mixture of small to mid-sized towns, forests, and agricultural areas with varying intensities of use. Farming in the region is highly industrialized and purely rain-fed. Main agricultural crops are cereals such as wheat, barley, and rye as well as maize and rapeseed (Ministerium für Energiewende, Landwirtschaft, Umwelt, 2020). Field shapes are irregular, with no predominant patterns. Field characteristics and distribution are heterogeneous, with sizes ranging from below 1 to above 10 ha. As seen in Figure 3.1, the landscape consists of a mixture of fields of different sizes and shapes. The whole study area had an extent of roughly 11.5 × 7.0 km². Within it, we selected two 2.5 × 2.5 km² regions of interest (ROIs) for detailed analysis (see Figure 3.1). We created a reference dataset for each of them representing all fields that should be extracted from the subset alone (i.e., all fields contained entirely within the subset).

Figure 3.1. Sentinel-2 greyscale image of the study area and the two regions of interest (ROIs).

3.2.2 Satellite Imagery

For our study, we used Sentinel-2 Level-2A atmospherically corrected imagery. To cover different growth stages and maximize the differentiation of neighboring fields (see Section

3.3.2), we explored scenes throughout the 2019 crop growing season (March until late October). Considering only observations without cloud cover over the study area, our dataset consisted of six scenes in March, April, June, July, August, and September. To provide sufficient spatial detail, we focused only on the 10-m-resolution red, green, and blue (RGB) bands of Sentinel-2. This further allows us to explore the potential applicability of our method to other sensors with a more limited selection of wavelength bands or even a standard RGB setup.

3.2.3 Reference Data

State cadastral data containing administrative plots served as a reference (Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland, 2016). Although the boundaries in this dataset sometimes aligned with actual field boundaries of 2019, manual updating was required based on field surveys and recent satellite imagery. Considering that the 10 m spatial resolution of Sentinel-2 limited the capacity to extract very small plots, we excluded all polygons below a size of 0.5 ha.

Interpretation of field conditions proved difficult. On some occasions, fields appeared as separate at one point in time (e.g., due to separate tilling) but merged in later images. In other cases, the delineation of two adjacent plots was difficult (see also Section 3.5).

3.2.4 Land Cover Mask

As we were only interested in agricultural areas, we masked all non-agricultural landscapes using land cover maps. We considered two products: the CORINE land cover data of 2018 and the Land Cover DE product of the German Aerospace Center (DLR) (Weigand et al., 2020; Büttner et al., 2017). Previous investigation demonstrated superior quality of the DLR product in our study area, but we still occasionally observed some urban and other non-agricultural areas as being misclassified. We therefore decided to use a combination of the two products to create a conservative land cover mask.

3.3 Methodology

The flowchart in Figure 3.2 provides an overview of the workflow of our approach. After providing some background on pixel value transformation and active contours, we first describe the image pre-processing and enhancement (green) in Section 3.3.2, then proceed with the edge detection (blue) in Section 3.3.3 before describing seed point selection (yellow) as well as contour and polygon extraction in Section 3.3.4 and Section 3.3.5.

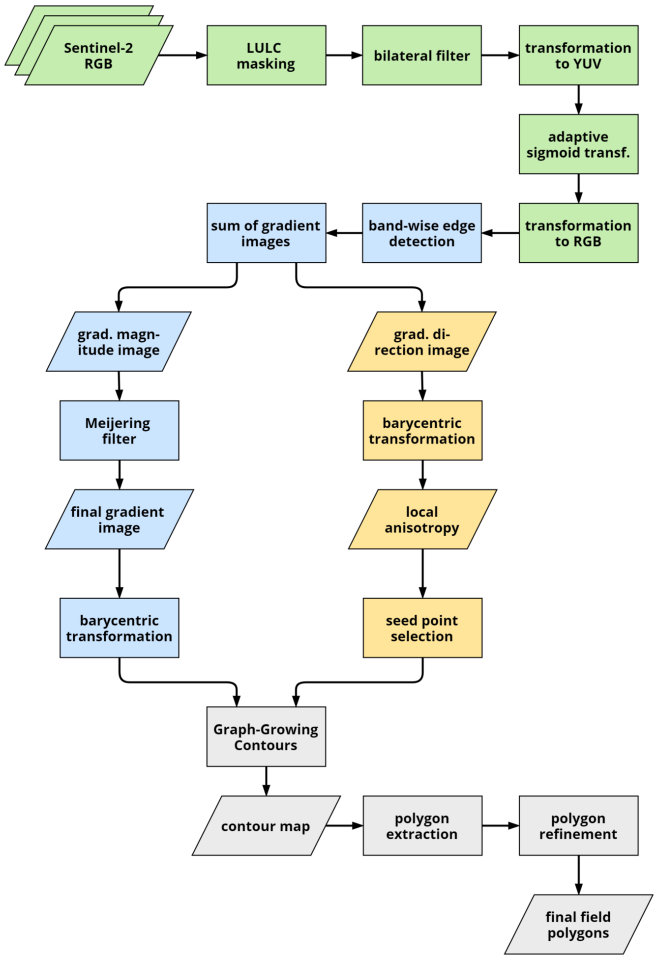

Figure 3.2. Flowchart of the boundary detection and field extraction procedures (LULC: land use/land cover map). Stages of the process are highlighted in different colors: pre-processing in green, edge detection in blue, seed point selection in yellow, and contour and subsequent polygon extraction in grey.

3.3.1 Background

3.3.1.1 Sub-Pixel Image Transform

To perform analyses on the sub-pixel level, we dynamically transformed the pixel grid data of the images using a barycentric transform, which represent the location of a point within a triangle (or more generally a simplex) as the center of mass (i.e., the barycenter) in relation to its vertices (Yiu, 2000).

Barycentric coordinates have proved to be useful in finite element analysis and computer graphics applications, and are superior to simple distance-weighted transforms. To obtain the value of a given point, the three closest grid points in the image are selected

as vertices of the triangle. The value is then computed as a linear combination of these vertices based on the following equations:

$$P_x = W_{v1}X_{v1} + W_{v2}X_{v2} + W_{v3}X_{v3} \tag{3.1}$$

$$P_y = W_{v1}Y_{v1} + W_{v2}Y_{v2} + W_{v3}Y_{v3} \tag{3.2}$$

$$W_{v1} + W_{v2} + W_{v3} = 1 \tag{3.3}$$

$$W_{v1} = \frac{(Y_{v2} - Y_{v3})(P_x - X_{v3}) + (X_{v3} - X_{v2})(P_y - Y_{v3})}{(Y_{v2} - Y_{v3})(X_{v1} - X_{v3}) + (X_{v3} - X_{v2})(Y_{v1} - Y_{v3})} \tag{3.4}$$

$$W_{v1} = \frac{(Y_{v3} - Y_{v1})(P_x - X_{v3}) + (X_{v1} - X_{v3})(P_y - Y_{v3})}{(Y_{v2} - Y_{v3})(X_{v1} - X_{v3}) + (X_{v3} - X_{v2})(Y_{v1} - Y_{v3})} \tag{3.5}$$

$$W_{v3} = 1 - W_{v1} - W_{v2} \tag{3.6}$$

where $v1$, $v2$, and $v3$ denote the three vertices of the triangle, X_{vi} and Y_{vi} are the x- and y-coordinates of the vertex v_i, and P_x and P_y are the x- and y-coordinates of the interior point. W_{v1}, W_{v2}, and W_{v3} are the weights assigned to each vertex. As long as a point is located within the triangle, all three weights are ≥ 0 and sum up to 1.

3.3.1.2 Active Contours and Growing Snakes

Active contours (also called snakes) were first introduced by Kass et al. (1988) for object delineation. The concept is based on the energy minimization of a deformable spline or polyline. The spline is initiated either via user input (e.g., a rough sketch around the object of interest) or automatically in the object's vicinity. The snake then follows two influencing factors: the external and internal energy. The external energy given by the image (e.g., the gradient magnitude) directs the snake towards the contour of interest. The internal energy determines the shape of the snake and usually consists of two parts: continuity and smoothness or curvature, often represented by the first and second derivative. They limit the curvature of the contour and the distance between individual points. This ensures that the snake does not expand or contract indefinitely or become discontinuous. A traditional

snake is described as a parametric curve $f(s) = [x(s), y(s)], s \in [0, 1]$, following the energy given by the equation below (Kass et al., 1988; Xu and Prince, 1998):

$$E = \int_0^1 \frac{1}{2} \left[\alpha ||f^{'}(s)||^2 + \beta ||f^{''}(s)||^2 \right] + E_{ext}\left(f(s)\right) ds \tag{3.7}$$

where $f^{'}(s)$ and $f^{''}(s)$ describe the first and second derivatives of the parametric curve $f(s)$ and α and β are weighting parameters that determine the influence on continuity (first derivative) and curvature (second derivative). $E_{ext}\left(f(s)\right)$ is the external energy. One major drawback of the original model (and many subsequent variants for that matter) is the difficult choice of initial placement due to high sensitivity regarding initialization (Velasco and Marroquin, 2003).

Many variants of the active contour model have been proposed to address various drawbacks or expand the model for use in new applications, including geodesic contours using level sets and various different external energies such as edge distance transform or gradient vector flow (Xu and Prince, 1998; Caselles et al., 1997). The aim of our use case was not to detect single objects, but rather to extract a complex network of contours that may not even be interconnected and conserve contour relationships. Given the irregular sizes and shapes of fields in our study area, it should further be independent of shape or size priors.

We therefore focused on the so-called growing snakes model introduced by Velasco and Marroquin, which was designed particularly for complex topologies where traditional active contours may struggle (Velasco and Marroquin, 2003). Growing contours are initialized on seed points along the contour itself and grow (or "move") in steps perpendicular to the local gradient direction (i.e., along the contour). Movement in each step is determined by selecting the position of highest gradient magnitude out of a set of possible locations. The process proceeds until the snake approaches an existing contour point. In this case, the growth process stops, forming either a T-junction or a closed line. Although capable of tracking heterogeneous boundaries, this method simply follows local maxima. Therefore, it has a tendency to create discontinuous contours, as it does not consider the smoothness of the curve and requires an additional smoothing step.

We observed that growing contours had a tendency to get "off track" easily when the image gradient was heterogeneous, especially at corner points of multiple edges. Following the local maximum of gradient magnitude, the contour moves away from the actual boundary. Once this happens, it will create false boundaries along any local maxima it encounters. To address this, we experimented with user-defined constraints, for example, allowing only a certain number of steps with low magnitudes before terminating the snake. However, this introduced another level of complexity, as the threshold may be dependent on the strength of the edge currently being explored. If the snake is initiated on a very

strong edge, a conservative threshold may ascertain good results; however, the process may stop prematurely if the explored edge is rather weak.

Another issue we encountered was the already mentioned tendency to produce discontinuous lines, either failing to represent sharp 90° turns and crossing edges or producing false turns. To avoid this, we limited the maximum amount of turns allowed and experimented with different movement schemes. Again, these modifications produced unreliable results and added complexity to the parameter settings. Hence, we developed a new graph-based growing contour model, described in Section 3.3.4.

3.3.2 Image Pre-Processing

The initial stage in our procedure is image pre-processing, serving multiple purposes: (a) removing non-agricultural areas; (b) reducing noise; (c) removing undesired effects due to sub-field features such as crop rows; and (d) enhancing actual field boundaries. The process comprised four parts. First, we mask non-agricultural areas based on the land use mask and created an RGB composite using bands 4, 3, and 2 of each scene. This particular band arrangement is not a prerequisite for further processing steps, and any three-band composite would suffice.

Second, we perform noise reduction and image smoothing. Particularly in edge detection, noise reduction is important to reduce the amount of spurious edges. In the context of this study, we define noise as any kind of non-boundary feature that may interfere with the boundary detection steps (e.g., soil and plant growth patterns). Although convolution of the image with a Gaussian kernel is a common approach, it has the negative side effect of blurring out potentially relevant edges as well. As we are particularly interested in edges corresponding to field boundaries, it is desirable to conserve stronger edges. To achieve this, we employ bilateral filtering. The bilateral filter combines a Gaussian distribution in the spatial domain (as regular Gaussian smoothing), defined by a standard deviation σ_s , with a Gaussian distribution in the value domain, defined by a standard deviation σ_r (Tomasi and Manduchi, 1998). While the former determines the weighting of neighboring pixels based on their distance from the center pixel, the latter determines how values are weighted based on similarity to the center pixel. Edges in the image refer to sharp changes in image reflectance. Therefore, a smaller σ_r will conserve even weaker edges, while a larger σ_r will lead to a stronger blurring. The definition of the two distributions should be chosen carefully to ensure that weak, potentially irrelevant, edges are removed while strong edges are conserved.

Third, we transform the RGB image into the YUV color space. The YUV color space, similar to the closely related YCbCr color model, transforms the RGB information such that luminance is stored in the Y component while U and V refer to chrominance

information. This enables image enhancement with reduced color information distortion. Fourth, we apply a sigmoid transform to the Y component to increase contrast of the image (Braun and Fairchild, 1998). The sigmoidal function $S(x)$ is defined as follows:

$$S(x) = \frac{1}{1 + e^{\alpha_s(x_i - x_0)}} \tag{3.8}$$

where x_i is the reflectance of a pixel, α_s is the slope of the sigmoidal curve (gain), and x_0 controls the centering of the sigmoid curve (offset).

As illustrated in Figure 3.3, luminance images in our observations were usually bimodal, with the two peaks representing dark areas such as fully grown crops and bright areas such as dry open soil, respectively. The cutoff location of the sigmoid function can be adaptively positioned between the two peaks to increase contrast between dark and bright parts in the image. In the last step, we transform the YUV data back to RGB color space for subsequent steps.

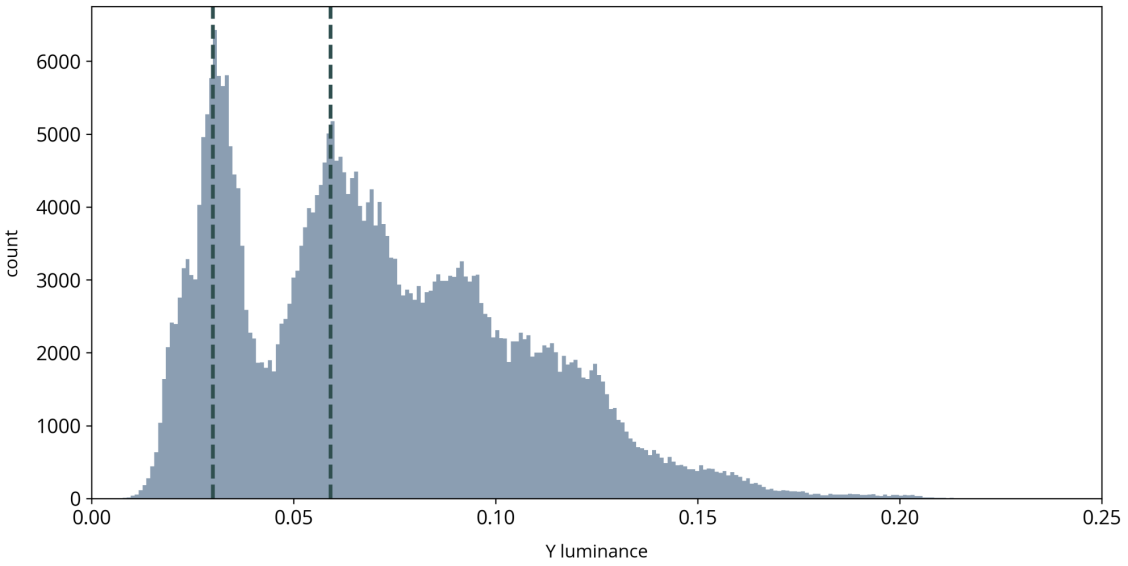

Figure 3.3. Example histogram of a luminance image. Vertical dashed lines illustrate the location of the two peaks used for positioning the cutoff location of the sigmoid transformation.

3.3.3 Edge Detection and Enhancement

We convolve each of the three bands of each scene with the Sobel operator in both x- and y-directions:

$$K_x = \begin{Bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{Bmatrix}, \; K_y = \begin{Bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{Bmatrix} \tag{3.9}$$

We sum up the resulting gradient images to highlight recurring edges and use the summed horizontal and vertical gradient images to obtain overall gradient magnitude g and gradient direction θ following the formulas:

$$g = \sqrt{I_x^2 + I_y^2} \tag{3.10}$$

$$\theta(x, y) = arctan\left(\frac{I_y}{I_x}\right) \tag{3.11}$$

where I_x and I_y are the gradient images in x- and y-directions.

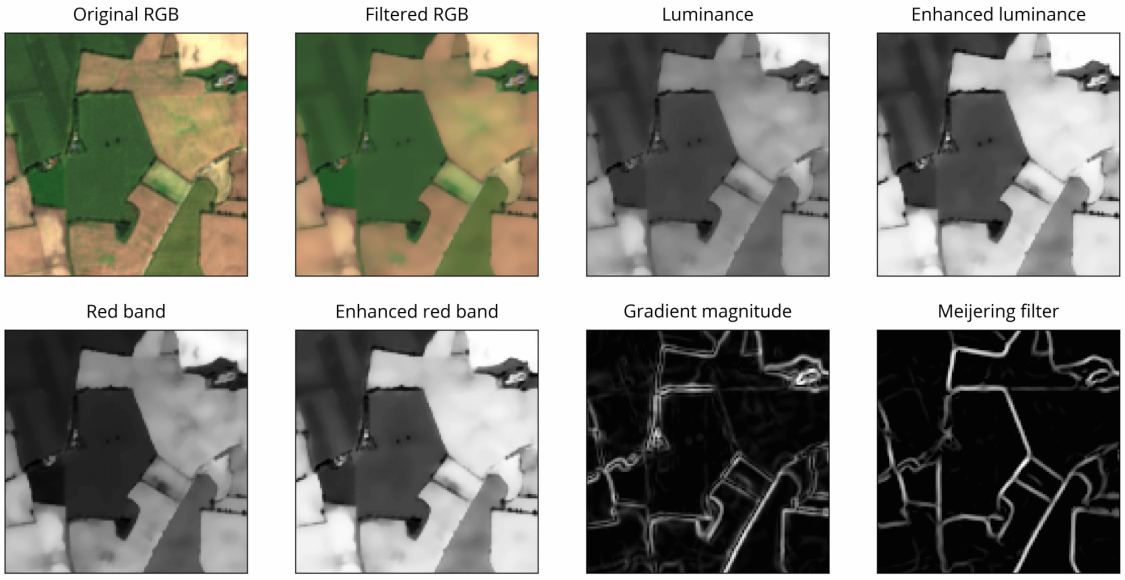

Figure 3.4. Examples of pre-processing steps.

Despite smoothing, the gradient magnitude still contains spurious edges caused by crop rows and other features within some fields. Some of these edges are rather strong and consistent (appearing in multiple scenes) and could not be removed by filtering alone. To overcome this and further emphasize the actual field boundaries, we exploit the shape and

appearance of boundaries by applying a Meijering filter to the magnitude image (Meijering et al., 2004). This filter was originally designed for medical purposes to extract neurites from microscopic images. It uses the eigenvectors and eigenvalues of the second-derivative (Hessian) matrix of the image to determine the "ridgeness" of a pixel. By applying this filter, we can exploit the curvilinear structure of the field boundaries in the image (see Figure 3.4).

As seen in Figure 3.4, many remaining spurious edges disappeared or were at least substantially reduced. Further, the Meijering filter provided a good representation of the actual contour shape compared to the sometimes uneven and blurred appearance of the gradient magnitude image. The resulting image was normalized and was the basis for most subsequent procedures, referred to as (gradient) magnitude.

3.3.4 Graph-Based Growing Contours

We overcome the limitations observed in the growing snakes model (see Section 3.3.1.2) with a newly developed modified growing contours model called graph-based growing contours (GGC). The goal of our GGC technique is not only to locate potential field boundaries but also to facilitate the extraction of the actual fields. The main improvement lies in a newly designed movement based on graph theory concepts. We aimed to address five goals: (a) improve the quality of extracted contours and reduce the need for repeated smoothing; (b) introduce adaptive behavior to accurately trace the contours of different shapes (smooth curves, sharp corners, straight lines etc.); (c) increase the capacity to extract complex, irregular networks with multiple crossing edges and T-junctions via automatic branching; (d) improve the handling of "dead ends"; (e) automatically create an interconnected network of contours rather than a set of separate segments.

Parts of this concept were inspired by the methodology by Aganj et al., who used a Hough transform and orientation distribution functions for fiber tracking in diffusion-weighted magnetic resonance images (Aganj et al., 2011). In a similar fashion, we employ principles of graph theory to extract the most likely boundary paths in a local neighborhood.

3.3.4.1 Seed Point Selection

The technique is rather flexible with respect to initialization, but ideally starts at one or multiple points on the boundaries of interest. In case of an interconnected network of boundaries, even a single initial seed point is often sufficient to extract all boundaries in the image. However, to ensure coverage over a larger area, multiple potential seeds are

preferable. This also ensures that weakly connected parts of the boundaries are correctly extracted.

Our observations showed that initializing the method on individual seed points consecutively provided the best results. The process starts on one seed point and proceeds until termination before being re-initialized on the next seed point. We therefore generate multiple seed points by selecting one point per 50 × 50 pixel image tile of the scene. For each tile, we limit the choice to those pixels located on strong edges. To determine potentially useful starting points, we assume that points near changing gradient direction such as corners or crossings of multiple edges are most suitable. These points allow the quick exploration of several branches and usually do not run the risk of producing only dead ends. As seen in Figure 3.5, regions along a single boundary are usually areas of similar gradient direction, while regions near corners, curves, and crossings contain gradients of different directions (Gregson, 1993). To transfer this to an easy-to-use metric, we approximate the local anisotropy. Isotropy is defined as being direction-invariant (Perona and Malik, 1990). Following this definition, pixels along edges are more anisotropic (low isotropy), while those near corners and crossings of multiple boundaries are less anisotropic (high isotropy).

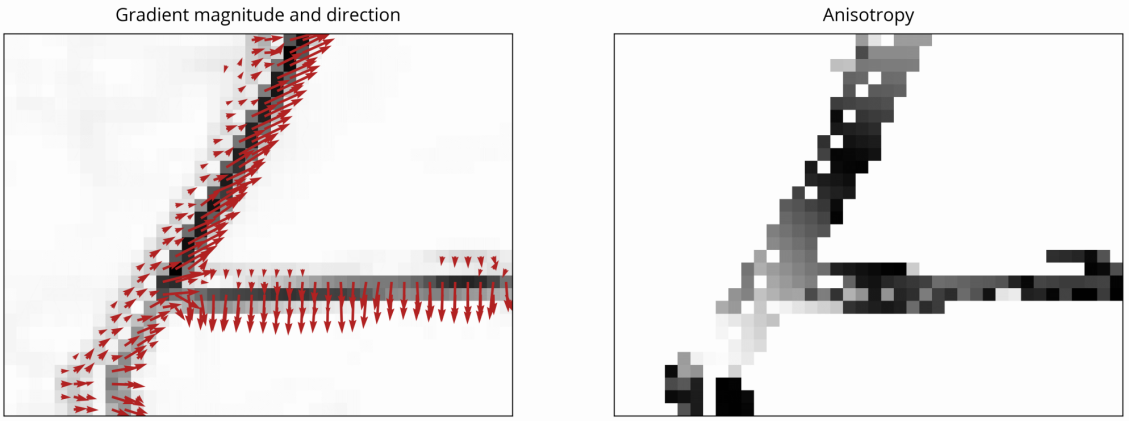

Figure 3.5. Visualization of gradient magnitude (greyscale) and gradient direction (arrows) on the left and anisotropy on the right. Darker levels of grey indicate higher values.

We approximate local anisotropy by sampling the gradient direction in a pattern around each pixel (similar to Section 3.3.4.2). We then determine the predominant or primary gradient direction by assigning the vectors to 16 bins between $-\frac{1}{2}\pi$ and $+\frac{1}{2}\pi$. The bin containing the largest number of vectors defines the predominant direction. We then project all vectors onto the main directional unit vector as well as the unit vector

normal to it. We calculate the total magnitude of projected vectors in both directions and obtain anisotropy via:

$$I_{aniso} = 1 - \frac{min\left(\sum_0^I ||proj_{v_{main}} v_i||, \sum_0^I ||proj_{v_{norm}} v_i||\right)}{max\left(\sum_0^I ||proj_{v_{main}} v_i||, \sum_0^I ||proj_{v_{norm}} v_i||\right)} \tag{3.12}$$

where v_{main} and v_{norm} are the main directional and normal vectors, respectively, v_i describes a given local gradient direction vector, and I is the number of all sampled vectors. If the total magnitude in both directions is similar, anisotropy is low. If the magnitude in one direction is much larger than the other, we observe a high anisotropy (see Figure 3.5). This information is used to select points of high isotropy as seed points.

3.3.4.2 Generating a Local Graph

At each end point, the model starts by creating a directed, weighted graph. Firstly, a set of points around the end point is sampled on the sub-pixel level using a concentric circular pattern, as seen in Figure 3.6. The pattern is defined by an inner radius r_{min}, an outer radius r_{max}, the number of points on the initial (innermost) circle n_i, and the number of evenly spaced circles n_c. Moving outwards, the number of points doubles with each circle, creating a homogeneous pattern. This ensures that each point on a circle has the same number of neighbors on the next circle. As seen in Figure 3.6, the pattern also ensures straight connections from each point of the inner circle to the outermost one.

The sampled points represent nodes in a local directed, unweighted graph $G = (V, E)$ constructed based on a set of rules:

- The center point and all sampled points are nodes V in the graph.
- The center node is linked via edges to all nodes of the initial circle with equal weights.
- Each subsequent node is connected to the same number n_e of nearest nodes on the next larger circle via directed, outgoing edges E.
- In the direction of the previous contour location, nodes of the graph are masked to avoid backtracking.

All edges (except the ones originating at the center) are weighted based on:

$$w_{i,j} = \frac{D_{i,j}}{g_j} \tag{3.13}$$

where $w_{i,j}$ is the weight of the edge $E(v_i, v_j)$, $D_{i,j}$ is the Euclidian distance between the nodes i and j, and g_j is the gradient magnitude at node j. This weighting allows

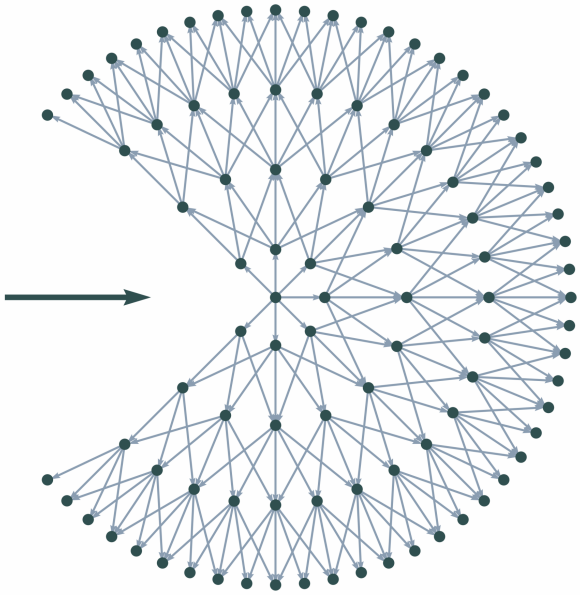

Figure 3.6. Example of a local graph with number of circles $n_c = 4$, initial circle size $n_i = 8$, and allowed connections per node $n_e = 5$. The bold horizontal arrow indicates the current direction of movement.

for consideration of both distance between the points (favoring shorter paths in terms of Euclidian distance) and regions of higher magnitude (favoring locations along an edge in the image).

In constructing the graph, n_c may be regarded as a metric of accuracy or smoothness because, given the same radius, more points will tend to create smoother curves (see Figure 3.7). Similarly, the number of allowed connections from each point to the next circle n_e will influence both the smoothness and the capacity for sharper directional changes (see Figure 3.7). We presume the number of points on the initial circle n_i as a parameter of potential directionality, which may influence the capacity to consider different directions. The parameter r_{max} may be regarded as a "step size".

As illustrated in Figure 3.7, different paths may lead to each point along the outer circle. This ensures the capacity to represent different shapes of contours, including sharp corners or smooth curvatures. However, the distance component in the weighting Equation 3.13 ensures that smoother curves are preferred. The complexity of the graph and, as a result, processing time, increase with larger values of n_i, n_e, and n_c.

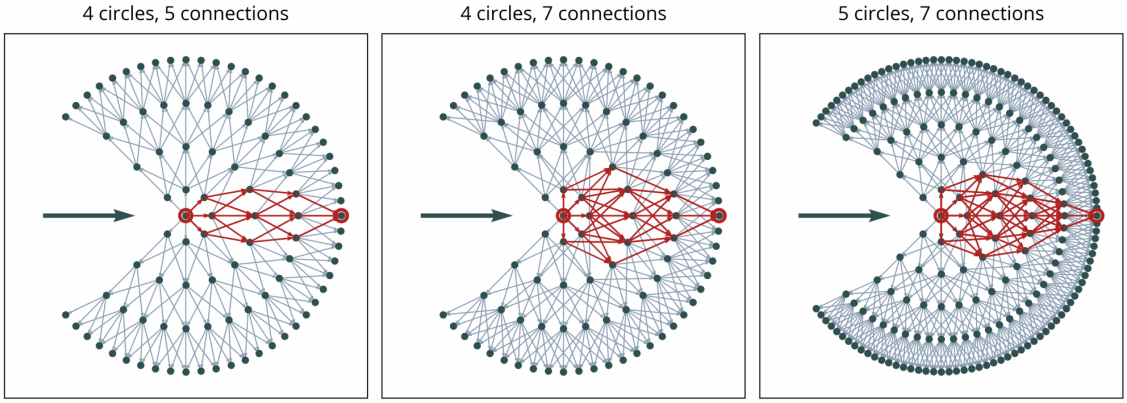

Figure 3.7. Illustration of possible paths from the center point of the local graph to a sink point on the outermost circle (indicated in red) for three different graph setups.

3.3.4.3 Movement

To determine the branches (i.e., directions of movement) at a current point, a shortest-path search starting at the center node (source) of the local graph to all nodes of the outermost circle (sinks) is initiated using Dijkstra's algorithm (Dijkstra, 1959). There are various potential ways of deciding subsequent directions. One may simply search for local minima or select a certain number of shortest paths. However, boundary features in the image usually expand over multiple end nodes on the circle. This may lead to multiple selected paths corresponding to the same image feature while other less-pronounced features remain unconsidered.

To avoid this, we first select the overall shortest path as the first branch. We take the location of the first path as a reference and split the graph into subsections around $-\frac{1}{2}\pi$ and $+\frac{1}{2}\pi$ from the main direction. Subsequent paths are selected only within these sections of the graph. The shortest paths in each of the sections are selected as potential branches (see Figure 3.8, right panel). If there are two edges branching at a very small angle, the two boundary features usually diverge quickly. This can still be represented in the graph (see Figure 3.8, right panel).

Any selected paths that are longer than a certain threshold L_{max} are omitted. This ensures superfluous branches to die off quickly rather than continuing indefinitely and leads to a dynamically changing number of end points and automatic termination of "dead ends".

An end point of the contour is terminated if no valid paths are found, if the current point is near the image boundaries, or if it is near an already existing contour point. In the latter case, the two points are connected.

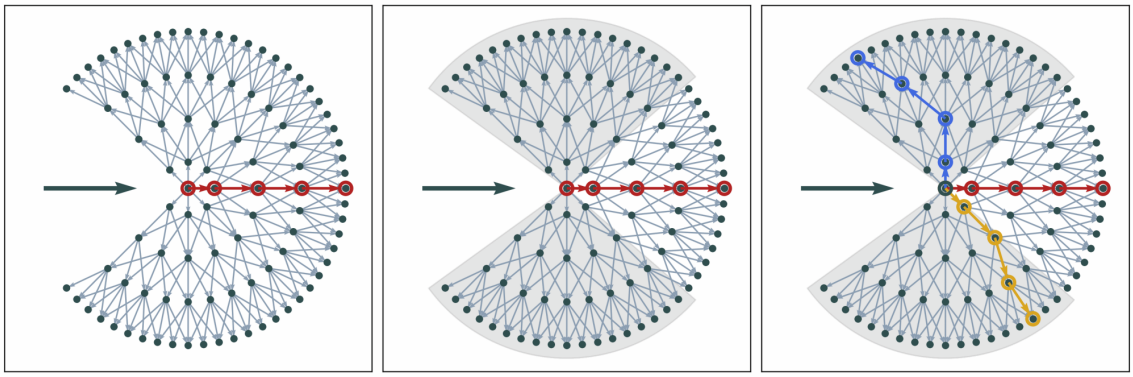

Figure 3.8. Main steps of movement in local graph (left to right): local graph with overall shortest path indicated in red; subsections in which to search for other paths; three selected paths to proceed with. The bold horizontal arrows indicate the current direction of movement.

3.3.5 Polygon Creation and Post-Processing

The output of GGC is a large undirected network (graph) of interconnected contour points. The next challenge is to transform this network of field boundaries into a set of individual field polygons. Simple clustering or associating points with certain shapes in the image are unreliable because points are usually part of multiple neighboring fields, and field shapes may be quite heterogeneous. We did not intend to introduce any assumptions about field shapes or contour arrangements.

Again, we make use of graph theory concepts. In theory, this is as simple as finding cycles in the graph, but in practice some problems may arise. On the one hand, the network may become very large, especially if small step sizes are used (i.e., small r_{max}), making the process of detecting cycles computationally demanding. On the other hand, the process requires a correctly interconnected network with no false edges or gaps.

To improve performance and increase reliability, we decided to create local polygons using a subset of boundary points. To achieve this, we transform the set of edges in the output graph to a binary contour image by drawing a binary edge map. We then employ a flood fill algorithm creating a segment representing the rough extent of a given field as defined by the extracted contours. Then, we extract the nodes of the network located within a certain distance from the boundary pixels of the flood fill segment. A new field boundary graph is created where each node is connected to its four nearest neighbors. This may produce an over-connected graph. The polygon representing the field is extracted by finding the longest cycle in this local network. To refine the polygons, we further employ adaptive Gaussian filtering and the Ramer–Douglas–Peucker algorithm (Deng and Cahill, 1993; Ramer, 1972; Douglas and Peucker, 1973).

3.3.6 Selecting Optimal Parameters

The pre-processing steps are particularly crucial for the performance of our approach, especially the correct filtering of the images. Regarding the parameters of the GGC model, we decided to set constant values for many of them to reduce complexity of the optimization task. In the two ROIs, we chose four parameters for optimization to obtain representative results: the two standard deviations defining bilateral filtering σ_r and σ_s, the gain α_s in the sigmoid transformation, and the maximum allowed path length L_{max} in GGC. For other settings we found $n_i = 8$, $r_{max} = 6$, and $n_e = 7$ to be good default values in both ROIs.

We obtained an optimum parameter setting for both ROIs using differential evolution in a "rand-best/1/bin" strategy with a population size of 20, differential weight of 0.3, and crossover rate of 0.8 (Das and Suganthan, 2011; Charalampakis and Dimou, 2015). We visually compared the obtained extraction results to a representative image and the reference datasets. The target variable of optimization was the mean Jaccard index (see Section 3.3.7). In both ROIs, the differential evolution converged within 20 iterations and determined an optimal parameter set as given in Table 3.1. Optimal settings for both ROIs were overall similar in terms of a quite large difference between σ_r and σ_s and relatively high values of α_s.

Table 3.1. Optimal settings obtained from optimizing polygon extraction for the two regions of interest.

	σ_r	σ_s	α_s	Ł$_{max}$
ROI 1	0.18	1.98	41.7	237.4
ROI 2	0.20	2.14	49.3	209.4

3.3.7 Performance Evaluation

We used a quantitative metric to evaluate the difference between extracted polygons and those in the reference set. We employed the frequently used Jaccard Index as our main reference metric because it is a common metric performing well for automatic segmentation (Jozdani and Chen, 2020). In a pre-cursor step, extracted and reference polygons need to be matched (Jozdani and Chen, 2020). Considering the set of reference polygons $X = \{x_i : i = 1, ..., n\}$ and the set of extracted polygons $Y = \{y_j : j = 1, ..., n\}$, we can define multiple subsets. Following Clinton et al. (2010), we can distinguish:

$$Y_{a_i} = \{y_j \,:\, the\, centroid\, of\, x_i\, is\, in\, y_j\}$$

$$Y_{b_i} = \{y_j \,:\, the\, centroid\, of\, y_j\, is\, in\, x_i\}$$

$$Y_{c_i} = \{y_j \,:\, area\,(x_i \cap y_j)\,/area\,(y_j) > 0.5\}$$

$$Y_{d_i} = \{y_j \,:\, area\,(x_i \cap y_j)\,/area\,(x_i) > 0.5\}$$

$$Y_i^* = Y_{a_i} \cup Y_{b_i} \cup Y_{c_i} \cup Y_{d_i}$$

Based on these matchings, we calculated the Jaccard Index (McGuinness and O'Connor, 2010; Ge et al., 2007):

$$JAC_{ij} = 1 - \frac{area\,(x_i \cap y_j)}{area\,(x_i \cup y_j)}, y_j \in Y_i^* \tag{3.14}$$

Resulting values range from 0 to 1, with 0 indicating an optimal segmentation. If the set of matched polygons is empty, JAC_{ij} is set to 1. All values calculated for individual polygons were averaged. In addition, we used statistical information in the form of the number of fields, median and standard deviation of field sizes, and total acreage as proxies for performance of field extraction. We focused our detailed analysis on the two ROIs but also briefly discuss the performance on the whole study area using both parameter settings.

3.4 Results

As shown in Figure 3.9, the overall appearance of the extraction results in both ROIs was close to a visual interpretation of the image. The general arrangement of extracted fields was matched and most shapes were correctly delineated, closely following the visible boundaries in the images. Spatially isolated phenomena such as patches of wet soil, grass growing within bare fields, or uneven growth stages within a field did not confuse the extraction as long as they were located at some distance from the field boundaries (see Figure 3.9, fields near the top of ROI 1 and at the center-right in ROI 2). However, if they were located close to or directly on a boundary, they could cause irregular field shapes or incorrect indents in the outline (e.g., Figure 3.9, ROI 2 center).

Uncertainties may also occur in case of temporally isolated distinctions, for example, when adjacent fields are only distinguishable at a specific point in time but remain homogeneous otherwise. Our approach was successful in detecting field boundaries, even if they were just visible in a single image. For example, the two elongated fields in the top

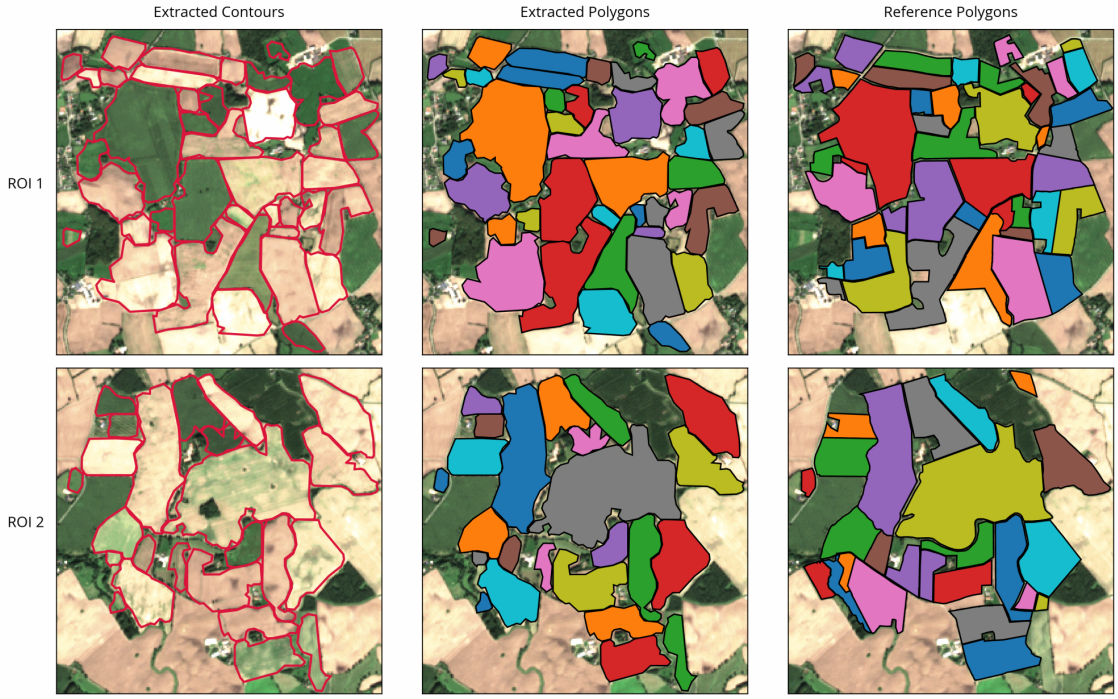

Figure 3.9. Results of field extraction for the two ROIs showing extracted contours, corresponding extracted polygons, and the reference dataset for comparison.

left in ROI 1 appear to be one large field in all but the one observation pictured, where only the northern half is ploughed (see Figure 3.9). In such cases, the strength of the delineating features is crucial (Section 3.5).

The method further detected some additional fields near the boundaries of the image that were not part of the reference dataset. This was due to the actual fields extending slightly outside the ROI, while the extraction still detected a large portion of the field.

Table 3.2 presents the statistics for the two ROIs. In general, results were close to those of the reference datasets. The total acreage of all fields showed a high agreement for both ROIs. Median field sizes in the two ROIs were quite different, with ROI 1 containing mostly smaller fields (median size 5.08 ha in the reference data) than ROI 2 (median size of 8.67 ha). Delineated field contours and sizes in ROI 1 matched accurately with a difference in median field size of just 0.52 ha. In ROI 2, however, the offset in median field size was more significant (1.39 ha). The number of fields extracted was off by -4 and +2 in ROIs 1 and 2, respectively. This was due to some larger fields being mistaken for two separate plots or vice versa, as well as the above-mentioned extraction of additional fields near the image boundaries.

Table 3.2. Field statistics in the two regions of interest.

		Field Count	**Median Field Size [ha]**	**St.Dev. Field Size [ha]**	**Total Acreage [ha]**
ROI 1	extracted	41	5.60	8.45	362.60
	reference	45	5.08	8.04	359.14
	difference	−4	0.52	0.42	3.46
	perc. diff.	−8.9 %	10.2 %	5.2 %	1.0 %
ROI 2	extracted	26	10.05	12.21	390.16
	reference	24	8.67	12.72	393.69
	difference	2	1.39	−0.51	−3.53
	perc. diff.	8.3 %	16.0 %	−4.0 %	−0.9 %

Results on the whole study area are shown in Figure 3.10 for both parameter setups. Both showed mixed results. In some areas (especially near the respective ROIs), results were good but performance deteriorated further away from the regions for which the settings were optimized.

3.5 Discussion

The presented GGC method proved reliable in extracting complex networks of boundaries with minimal supervision. Once initiated, it automatically detected relevant boundaries in the image, often only requiring a single seed point. Automatic branching at diverging or crossing edges worked reliably, and "dead ends" (i.e., undesired branches following spurious edges) were avoided for the most part. This ensured the successful extraction of many of the irregularly shaped fields in the reference data.

The field extraction approach produced good results in terms of statistical characteristics of the field structure in the two ROIs. The total acreage was matched very closely. Overall, ROI 2 proved more challenging than ROI 1, possibly due to a more heterogeneous appearance because of wet soils and uneven growth patterns. This became apparent in the overestimated median field size. The total number of fields extracted in the two ROIs also reflected some issues. The algorithm missed some very small fields, while merging some larger, heterogeneous ones. However, as mentioned before, interpretation of actually separated fields vs. separately managed parts of one larger field is difficult, and may explain some of these issues.

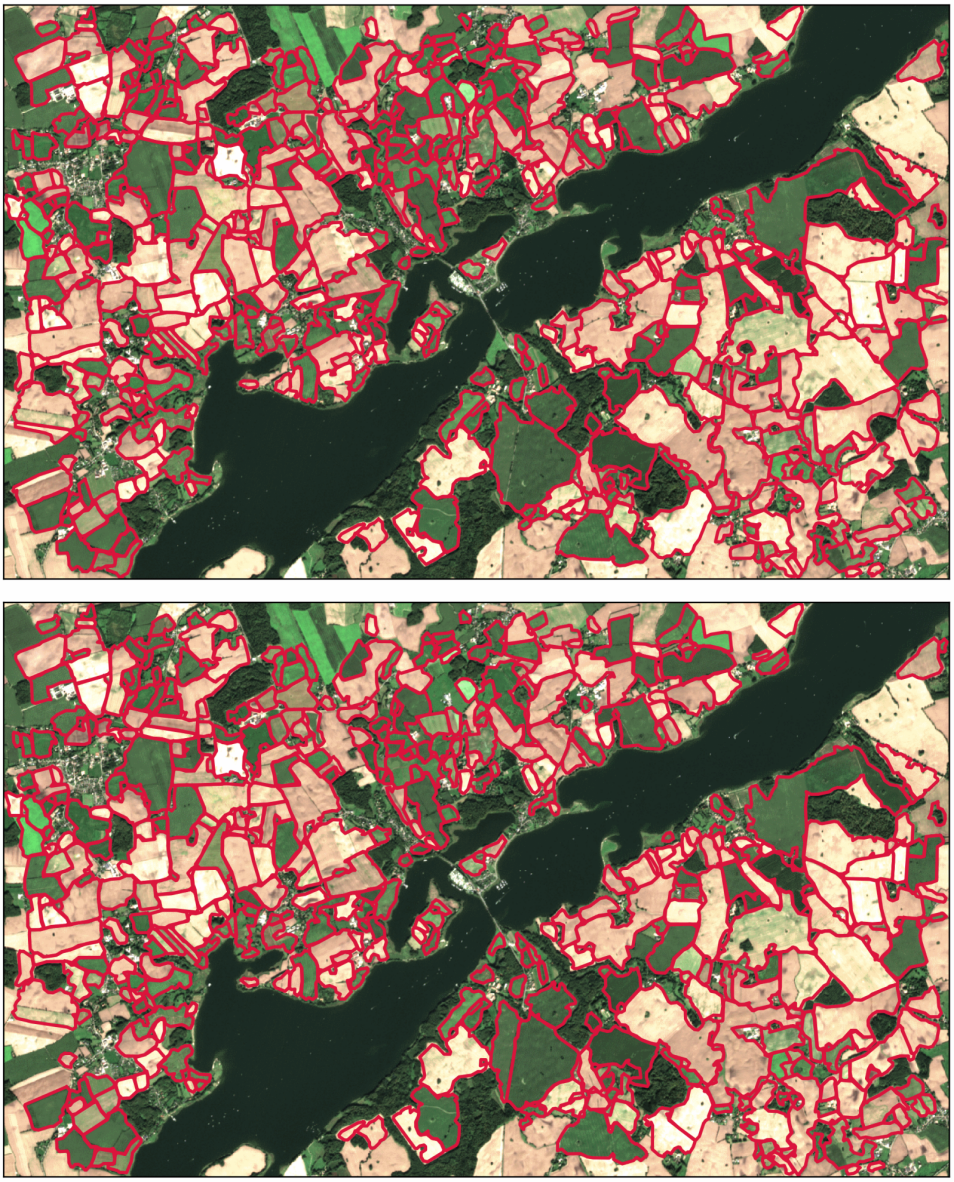

Figure 3.10. Results of field extraction for the whole study area using parameter settings of ROI 1 (top) and ROI 2 (bottom) as listed in Table 3.1.

The proposed method further proved reliable in extracting even narrow field strips and many smaller plots, but was occasionally confused by inconsistencies near urban areas or single man-made structures. This may be addressed through better masking of undesired objects, but would require much better knowledge of conditions in the area or significantly more pre-processing time. For such cases, further research is needed to ensure higher reliability.

We also observed uncertainties regarding temporally confined features like temporarily wet soils or field management, which can introduce heterogeneities within a single field.

In such cases, delineation is may be ambiguous. In part, the use of multi-temporal data sets helped to enhance weaker but consistent field boundaries. In most cases, our technique was capable of detecting weak but consistent boundaries. Problems occurred, however, if the separation only appeared in a single observation: if the boundary is pronounced, the algorithm is capable of correctly delineating the fields, even if a separation is only visible in one observation. However, if groups of adjacent fields are separated by a weak border, they may be detected as a single large field (see Figure 3.9, ROI 1 bottom left). The point at which different appearances or spatially or temporally constrained boundaries actually justify a separation into multiple fields is debatable. However, by adjusting L_{max}, the user can influence the degree to which the model considers weaker boundaries. Since there is a trade-off between detecting weak boundaries and reducing the risk of extracting false boundaries, more research is needed to address this issue – for example, by adapting pre-processing steps such as contrast stretching or edge detection to the individual images or by introducing a dynamic weighting.

Overall, we observed that the pre-processing had the greatest influence on the performance of the whole process. The filtering and edge enhancement steps we employed proved for the most part effective in reducing background noise and highlighting actual field boundaries. Especially, the use of multi-temporal Sentinel-2 data to emphasize weaker structures and the Meijering filter to reduce spurious edges proved useful. In general, the better the prior boundary detection step, the more reliable and efficient the GGC contour extraction process can be. With a more reliable and homogeneous input, a larger radius r_{max} can be chosen without the need to increase the number of circles n_c too much. This would speed up the process and reduce the number of iterations required.

The spatial resolution of Sentinel-2 also limits the approach. Although 10 m resolution is sufficient in most cases, it may hamper the precise extraction of very small fields. Overall, we found the spatial resolution to be suitable for this task, but we expect that higher resolutions would provide better results. We would further speculate the "sweet spot" resolution to be between 1 and 5 m for this type of application. At this scale, delineation would become even more accurate and would make it possible to distinguish smaller objects such as single man-made structures, narrow hedges, or trees. However, with sub-meter resolutions, separating meaningful information from "noise" like soil texture or individual plants may become difficult. Moreover, processing time would increase dramatically and the benefit from additional detail would probably not outweigh the drawbacks. Future research may explore the performance of our technique on higher-resolution imagery.

It became apparent that different parameter settings are required for reliable field extraction in the two ROIs, in both the pre-processing and the actual extraction steps. This was particularly obvious in applications to the whole study area, and may have

been aggravated by the heterogeneous structure of the agricultural landscape. Different circumstances in different parts of the region led to inconsistent results when using the same settings for the entire area. As mentioned previously, this was mainly due to different requirements in the boundary detection step. In the future, it would be desirable to explore an adaptive procedure that chooses parameters based on characteristics such as structure, morphology, and statistics of the input imagery. Judging from the results of this study, subsets even smaller than 2.5 × 2.5 km² may be advantageous to allow local adaptations, even at field scales. As mentioned above, further improvements to the algorithm may also include pre-processing and boundary detection steps. Considering not only different scene characteristics but also the date of image acquisition in a time series may improve delineation results, for example, via dynamic image filtering and enhancement procedures depending on the characteristics of individual images. Further, analysis of image stacks may be employed to dynamically select desirable features (e.g., by giving higher weights to images with more pronounced boundary features). The polygon refinement may also be adapted to criteria such as predominant field shapes and required detail.

It is also desirable to ensure that higher values of L_{max} do not result in more dead ends along extracted boundaries but only prolong movement along relevant but weak boundaries.

3.6 Conclusions

We presented a new method for field boundary detection and subsequent field polygon extraction based on image enhancement, edge detection, and a new version of the growing snakes active contours model called graph-growing contours. The approach succeeded in extracting field boundaries on the sub-pixel level from a set of Sentinel-2 RGB images. The method is very flexible in its application, as it is not restricted to imagery of a certain sensor, resolution, or wavelength, but can utilize any combination of bands as an RGB input. The boundary extraction step using the new GGC model also requires little supervision. Once initialized, it automatically extracts even large networks of complex interconnected boundaries.

There were some issues with respect to weak field boundaries, urban structures, or temporary disturbances such as wet soil patches in fields. However, most of these are probably best addressed in the pre-processing steps and are not inherent flaws of the extraction procedure. The flexibility of the presented contour extraction allows the use of any kind of image-like data representing field boundaries. The polygon extraction method may therefore be used in combination with other field boundary detection algorithms. We are currently exploring the potential to improve the process with respect to small-scale image features, inconsistencies at edges, and better highlighting of relevant vs. irrelevant

boundary features. Further, we plan to generalize and scale up the extraction procedure to larger areas – possibly even landscape scales. This may require further work regarding automatic adaptation to local conditions and available imagery or improvements to the boundary detection step.

Acknowledgments

We thank the students and student assistants who carried out in-situ mapping campaigns and helped to produce and update the reference data.

Chapter 4

Deep Learning and Adaptive Graph-Based Growing Contours for Agricultural Field Extraction

M. P. Wagner, N. Oppelt
Remote Sensing, 2020, 12(12), 1990
Special Issue "Deep Neural Networks for Remote Sensing Applications"
DOI: 10.3390/rs12121990

Received: 20 May 2020; Accepted: 18 June 2020; Published: 21 June 2020

Abstract

Field mapping and information on agricultural landscapes is of increasing importance for many applications. Monitoring schemes and national cadasters provide a rich source of information but their maintenance and regular updating is costly and labor-intensive. Automatized mapping of fields based on remote sensing imagery may aid in this task and allow for a faster and more regular observation. Although remote sensing has seen extensive use in agricultural research topics, such as plant health monitoring, crop type classification, yield prediction, and irrigation, field delineation and extraction has seen comparatively little research interest. In this study, we present a field boundary detection technique based on deep learning and a variety of image features, and combine it with the graph-based growing contours (GGC) method to extract agricultural fields in a study area in northern Germany. The boundary detection step only requires red, green, and blue (RGB) data and is therefore largely independent of the sensor used. We compare

different image features based on color and luminosity information and evaluate their usefulness for the task of field boundary detection. A model based on texture metrics, gradient information, Hessian matrix eigenvalues, and local statistics showed good results with accuracies up to 88.2%, an area under the ROC curve (AUC) of up to 0.94, and F_1 score of up to 0.88. The exclusive use of these universal image features may also facilitate transferability to other regions. We further present modifications to the GGC method intended to aid in upscaling of the method through process acceleration with a minimal effect on results. We combined the boundary detection results with the GGC method for field polygon extraction. Results were promising, with the new GGC version performing similarly or better than the original version while experiencing an acceleration of 1.3× to 2.3× on different subsets and input complexities. Further research may explore other applications of the GGC method outside agricultural remote sensing and field extraction.

4.1 Introduction

Field mapping is a topic of increasing relevance. Knowledge of field sizes, shapes, and distributions is valuable for a wide variety of topics, ranging from precision agriculture to biodiversity mapping, agricultural policy, yield estimation, and cropland classification (García-Pedrero et al., 2017; Rahman et al., 2019; Turker and Kok, 2013). Today, monitoring is mostly based on field campaigns, administrative maps, and manual airborne or satellite imagery interpretation (García-Pedrero et al., 2017). However, such approaches are time-consuming and costly, and require manual interpretation. Regularly updated and standardized national cadasters, such as the Land Parcel Identification Systems or the Integrated Administration and Control Systems promoted by the European Union, can mitigate this but are still prone to subjective interpretation in the mapping process and difficulty of evaluating information (García-Pedrero et al., 2017; Sagris and Devos, 2008; Inan et al., 2010; European Court of Auditors, 2016). Furthermore, financial and labor limitations often hamper timely updates.

Remote sensing promises to alleviate these issues through regular large-scale observations of agricultural landscapes and much lower cost thanks to increasing amounts of freely accessible data. As a result, earth observation data has seen extensive use in agricultural applications over the years, covering topics such as crop type classification, plant health monitoring and stress detection (Jégo et al., 2012; Verrelst et al., 2014; Liu et al., 2010; Belgiu and Csillik, 2018; Kussul et al., 2016; Peña et al., 2014). The topic of field boundary detection and field extraction, however, has received relatively little interest so far.

Field boundary detection is commonly tackled through edge detection, clustering, and segmentation techniques, as well as machine learning classifications such as random

forest (Watkins and van Niekerk, 2019; Yan and Roy, 2014; Tiwari et al., 2009; Debats et al., 2016). Other approaches include the use of image characteristics such as texture and statistical modeling (Rahman et al., 2019; Da Costa et al., 2007). Frequently, separation of individual fields is performed via segmentation of pixels or clusters of pixels, ultimately limiting the accuracy of extracted fields.

Simultaneously, artificial intelligence methods, such as artificial neural networks (NNs), have frequently proven their merits in the context of edge detection, classification, and segmentation (Peña et al., 2014; El-Sayed et al., 2013; Kemker et al., 2018; Längkvist et al., 2016). Recently, research has focused on convolutional NNs that were also applied to field boundary detection (Masoud et al., 2019; Waldner and Diakogiannis, 2020; Persello et al., 2019; Kamilaris and Prenafeta-Boldú, 2018). Neural networks have the advantage of being capable of extracting highly non-linear relationships and have proven to be highly capable in different classification tasks. However, a drawback often noted is the difficulty of transferring results obtained in one region or on a specific dataset to another.

In a previous publication, we presented a full workflow combining agricultural field boundary detection and polygon extraction steps (Wagner and Oppelt, 2020). Starting from image pre-processing and enhancement, we combined edge detection, contour detection, and, finally, field polygon extraction. We used a newly developed, modified growing contours method, called graph-based growing contours (GGC), to extract complex networks of contours present in agricultural field boundaries. This method is flexible regarding the input information used and proved to be promising in extracting field boundaries, even in rather complex environments.

However, this process is naturally dependent on the quality of the field boundary map used as an input. Despite extensive pre-processing and image enhancing, contours in the previous study were often heterogeneous in strength, and it proved to be difficult to create a reliable boundary map for the subsequent field polygon extraction steps. This necessitated optimization of local parameter settings to achieve comparable results in different parts of an image. Further, we had to resort to small step sizes in the extraction procedure, resulting in longer processing time and hampering scale-up to larger scenes.

To address these drawbacks and enable large-scale applications of our methodology, in this study we explore an improved boundary detection approach. We decided to use multilayer perceptron neural networks (MLP-NNs) and aimed to develop a structure with a high potential of performing well under different environments and in different locations (Bishop, 2006). Following an extensive literature review, we selected a large number of potentially useful input features for successful agricultural boundary detection using MLP-NNs. We explored the usefulness of different features based on various image char-

acteristics and developed a boundary detection model based on study areas located in the east of the state of Schleswig-Holstein, Germany. By limiting our analysis only to the red, green, and blue (RGB) bands of Sentinel-2 (S-2), we wanted to demonstrate the use of only the most basic imagery, making our approach largely independent of the sensor.

We further introduce an improved version of the previously presented GGC algorithm with scalability in mind. We apply both versions of the GGC algorithm to the obtained boundary maps and analyze results of the boundary detection and the extracted polygons, as well as processing speed of the different GGC algorithms.

The paper is structured as follows. In Section 4.2, we describe the study area and the data we used. Section 4.3 presents the methodology, including image feature preparation, input selection, and model training. Section 4.4 shows results of the boundary detection and performance of subsequent polygon extraction. Section 4.5 discusses results and Section 4.6 summarizes the most important findings.

4.2 Data and Materials

4.2.1 Study Areas

We obtained data from two study areas in the state of Schleswig-Holstein, Germany (see Figure 4.1). Both were predominantly used for agricultural purposes. Field sizes were rather heterogeneous in both regions, ranging from below 1 ha up to about 75 ha. Fields, especially in study area 1, were often irregular in shape. The landscape structure is a mixture of agricultural areas with small forests, grasslands, and smaller urban agglomerations.

The study areas are located in the so-called “Schleswig-Holsteinisches Hügelland”, a hilly but low-elevation region along the coast of the Baltic Sea in northern Germany that forms one of three large landscape zones in the state of Schleswig-Holstein. The climate is temperate/oceanic with warm summers and wet winters (Cfb in the Koeppen–Geiger climate classification) (Kottek et al., 2006). Farming in the region is rain-fed and highly industrialized. Common agricultural crops include cereals such as wheat and barley but also maize and rapeseed (Statistisches Amt für Hamburg und Schleswig-Holstein, 2019).

4.2.2 Satellite Imagery

We used Sentinel-2 Level-2A atmospherically corrected imagery. For each study area, we considered observations over the 2019 growing season (March to October) and selected those with no cloud cover over the respective study area. We considered three observation times in early (March), mid (June), and late (September) growing season. To allow for

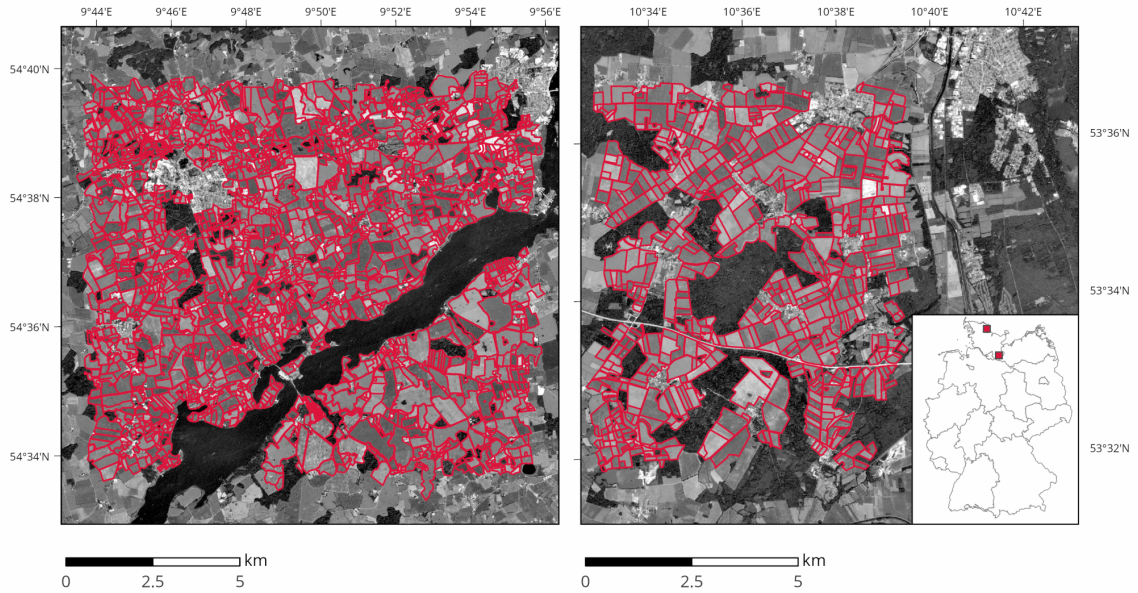

Figure 4.1. Maps of study areas 1 (left) and 2 (right).

high spatial detail and independence of very specific wavelength bands, we used only 10-m-resolution red, green, and blue (RGB) bands. Pre-processing included filtering of each band to reduce noise and remove undesired features. Considering our interest in field boundaries, we opted for using bilateral filtering rather than traditional Gaussian smoothing. The bilateral filter is an edge-preserving filter balancing similarity in the spatial and spectral domains to smooth the image while still conserving strong edges (Tomasi and Manduchi, 1998). The filtered images were then transformed from RGB to CIELAB color space to separately observe information on luminosity (l) and color (a, b) in the images.

4.2.3 Reference Data

We used cadastral data as a reference (Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland, 2016). This data, however, represented purely administrative borders and needed manual updating based on field surveys and recent satellite imagery to accurately represent conditions in 2019. We omitted fields that were not clearly discernable in S-2 imagery, e.g., very small or thin and elongated fields. Table 4.1 contains some general information about the study areas.

Table 4.1. Field statistics in the two regions of interest based on the updated reference datasets.

Study Area	Total Extent	Field Sizes	Field Count
1	~13.5 × 12.4 km²	~0.9 to 75 ha	~1400
2	~0.2 × 7.5 km²	~0.8 to 50 ha	~600

4.3 Methodology

The flowchart in Figure 4.2 demonstrates the structure of our methodology. The pre-processing (green) was discussed in Section 4.2.2. In this chapter, we start with describing the image features we obtained in Section 4.3.1 (blue), followed by the boundary detection model development, including feature selection and hyperparameter tuning (yellow), in Section 4.3.2. Finally, we describe the modified GGC method and the field polygon extraction process (grey) in Section 4.3.3.

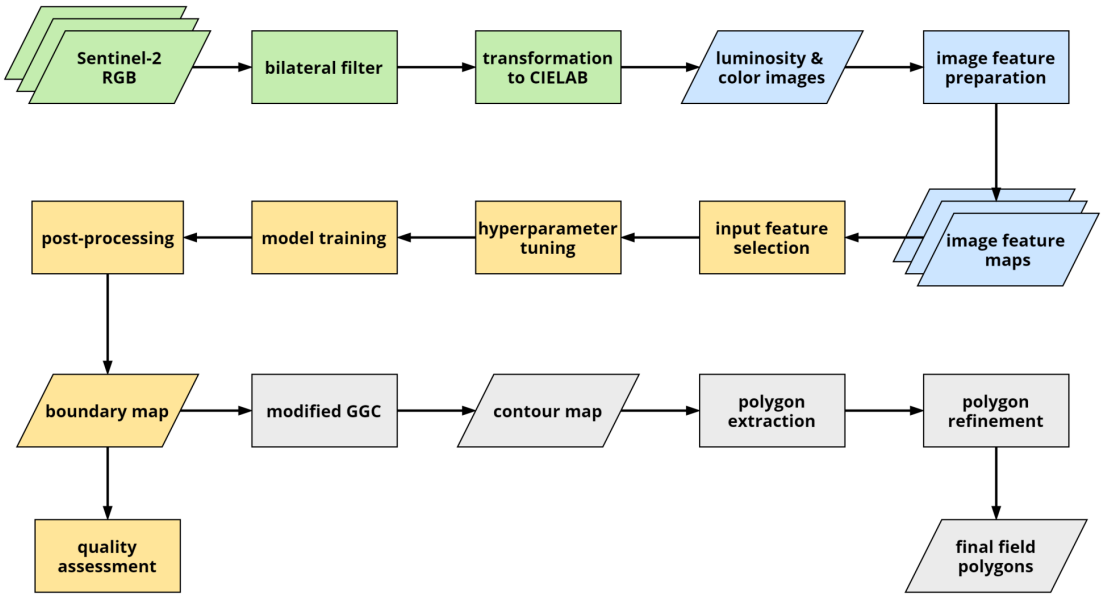

Figure 4.2. Flowchart of the methodology. Different stages highlighted in colors: pre-processing in green, image feature preparation in blue, boundary detection model development in yellow, and contour detection and field polygon extraction in grey.

4.3.1 Image Feature Preparation

We undertook an extensive literature review on topics of edge detection, boundary detection, and segmentation, and selected a set of potentially useful input features for field boundary detection. We distinguished applications on the luminosity (l) and the color channels (a, b) in the CIELAB color space. Table 4.2 provides an overview of all inputs

we considered. In the following sub-sections, we briefly describe the concepts behind the features and how we obtained them. In the text, we also refer to features and feature groups by their numbers in Table 4.2 as F1, F2, G1, G2, etc. All inputs were scaled linearly to a range of 0 to 1 based on the overall minima and maxima observed.

Table 4.2. Overview of features and respective feature groups considered for boundary detection.

Feature Group		**Feature**	
G1	image gradient (l)	F1	gradient magnitude (l)
G2	image gradient (a, b)	F2	gradient magnitude (a, b)
G3	local statistics (l)	F3	variance (l)
		F4	skewness (l)
		F5	kurtosis (l)
G4	local statistics (a, b)	F6	variance (a, b)
		F7	skewness (a, b)
		F8	kurtosis (a, b)
G5	Hessian matrix (l)	F9	eigenvalue along ridge (l)
		F10	eigenvalue across ridge (l)
G6	Hessian matrix (a, b)	F11	eigenvalue along ridge (a, b)
		F12	eigenvalue across ridge (a, b)
G7	texture metrics (l)	F13	contrast (l)
		F14	correlation (l)
		F15	asm (l)
		F16	homogeneity (l)
G8	texture metrics (a, b)	F17	contrast (a, b)
		F18	correlation (a, b)
		F19	asm (a, b)
		F20	homogeneity (a, b)
G9	angular dispersion (l)	F21	modified angular dispersion (l)
G10	angular dispersion (a, b)	F22	modified angular dispersion (a, b)
G11	homogeneity measures (l)	F23	PEG (l)
		F24	QEG (l)
G12	homogeneity measures (a, b)	F25	PEG (a, b)
		F26	QEG (a, b)
G13	local cues (l)	F27	brightness gradient (l)
		F28	texture gradient (l)
G14	local cues color (a, b)	F29	color gradient (a, b)

4.3.1.1 Image Gradient

The image gradient is the first derivative of the image and highlights discontinuities. To obtain gradient magnitude (F1, F2), we first applied the Sobel operator in both x- and

y-directions:

$$K_x = \begin{Bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{Bmatrix}, K_y = \begin{Bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{Bmatrix} \tag{4.1}$$

We then obtained gradient magnitude g based on the following formula:

$$g = \sqrt{I_x^2 + I_y^2} \tag{4.2}$$

where I_x and I_y are the gradient images in x- and y-direction. For the application on color layers (F2), we summed up the individual gradient images I_x and I_y before calculating magnitude and direction from the result.

4.3.1.2 Local Statistics

We calculated local statistics for each pixel based on a 5 × 5 kernel neighborhood (F3-F8). We considered three statistical measures: variance and the third and fourth standardized moments, skewness and kurtosis (NIST, 2020). The variance is defined as the spread of values in a (normal) distribution:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \tag{4.3}$$

where $x_i, ..., x_n$ are the sampled values, μ is the mean of the sample, and N is the sample size.

Skewness refers to the asymmetry of a distribution. A negative skewness means the tail of the distribution to the left is longer and the distribution "leans to the right"; a positive skewness represents a longer tail to the right and the distribution "leans to the left". We used the Fisher–Pearson coefficient formula to obtain skewness in the local sample (NIST, 2020):

$$\tilde{\mu}_3 = \frac{\sum_{i=1}^{N} \frac{(x_i - \mu)^3}{N}}{\sigma^3} \tag{4.4}$$

where σ is the standard deviation.

Kurtosis describes the shape of a distribution in terms of its "curvedness", i.e., how convex or concave the shape of the underlying distribution is. We used the following

formula to calculate kurtosis (NIST, 2020):

$$\tilde{\mu}_4 = \frac{\sum_{i=1}^{N} \frac{(x_i - \mu)^4}{N}}{\sigma^4} \tag{4.5}$$

In the case of application to color channels, we considered the maximum of the two results (F6-F8).

4.3.1.3 Hessian Matrix

Following concepts described in Meijering et al. (2004), Ando (2000), and Sato et al. (1998), we used the Hessian matrix of the image to obtain information on ridge-like structure strength and orientation.

The Hessian matrix of a 2D image $I(x)$ is given by:

$$\Delta^2 I(x) = \begin{bmatrix} \frac{\delta^2}{\delta x^2} & \frac{\delta^2}{\delta x \delta y} \\ \frac{\delta^2}{\delta y \delta x} & \frac{\delta^2}{\delta y^2} \end{bmatrix} \tag{4.6}$$

where $\frac{\delta^2}{\delta x^2}$, etc. are the partial second derivatives of the image I. Following the descriptions by Sato et al., we obtained the eigenvectors $e_1(x)$ and $e_2(x)$ of $\Delta^2 I(x)$ and their corresponding eigenvalues λ_1 and λ_2 (Sato et al., 1998). The eigenvector with the higher eigenvalue represents the direction in which the second derivative is maximal. In the case of a line-like structure, the eigenvalue across the line (or across ridge) is high and the eigenvalue along the line is low.

We use the eigenvalues along and across the ridge (F9–F12) as potential inputs to the model.

4.3.1.4 Second-Order Texture Metrics

The concept of the grey-level co-occurrence matrix (GLCM) and the derived second-order texture metrics was first introduced by Haralick et al. (1973) as a means to quantify the texture of imagery for classification. The GLCM refers to a matrix capturing the spatial relationship of values in a raster by counting how often two values occur together ("co-occurrence") considering different directions (horizontal, vertical, two diagonals) and different distances (steps apart).

At first, a quantization is performed in which the image values are assigned to a fixed number of gray levels. We performed quantization error minimization to reduce the data to 32 grey levels. We considered four second-order texture metrics: contrast, correlation, angular second momentum (ASM), and homogeneity ("inverse difference moment" in the

original paper) (F13–F20):

$$Contrast = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j} \, (i-j)^2 \tag{4.7}$$

$$Correlation = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j} \left[\frac{(i-\mu_i)(j-\mu j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \tag{4.8}$$

$$ASM = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j}^2 \tag{4.9}$$

$$Homogeneity = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2} \tag{4.10}$$

where i and j are two quantization values (or coordinates in the GLCM), $P_{i,j}$ is the probability value of the cell (i, j), N is the number of columns and rows in the GLCM (i.e., the number of grey levels), and μ_i, μ_j, σ_i, σ_j are the means and standard deviations of the marginal distributions of $P_{i,j}$.

We considered all four possible angles (0°, 45°, 90° and 135°) with a maximum of two steps in the 5 × 5 neighborhood. Final texture metrics for each pixel were obtained as the mean of all outputs of the process. In case of color layers, we took the mean of the resulting metrics in the two channels.

4.3.1.5 Angular Dispersion

Following the concept of angular dispersion introduced by Gregson (1993) and the fractional anisotropy described in Aganj et al. (2011), we calculated a modified angular dispersion metric similar to the local anisotropy we used for seed point selection in our previous study (Wagner and Oppelt, 2020). The concept is based on the observation that gradient direction tends to be homogeneous along edges or boundaries in an image while they are heterogeneous ("dispersing") if no clear boundary is present or boundary directions are changing, for example, at crossings of multiple edges or along sharp curves.

To approximate this, we consider the mean direction of gradient angles and the normal to it. By projecting all gradient direction vectors onto the two main directions, we calculate angular dispersion as follows:

$$I_{disp} = 1 - \left(\frac{min\left(\sum_0^I ||proj_{v_{mean}} v_i||, \sum_0^I ||proj_{v_{norm}} v_i||\right)}{max\left(\sum_0^I ||proj_{v_{mean}} v_i||, \sum_0^I ||proj_{v_{norm}} v_i||\right)} \right) \tag{4.11}$$

where v_{mean} and v_{norm} represent the mean directional and normal vectors, v_i describes a given local gradient direction vector, and I is the number of all sampled vectors. If the total magnitude of vectors projected in the mean and the normal direction is identical, we obtain a value of 0 (low dispersion); if the two total magnitudes differ largely, we obtain a value close to 1 (high dispersion).

We consider gradient directions in a 3 × 3 pixel neighborhood (F21-F22).

4.3.1.6 Homogeneity Measures

Ando presented an approach for edge and corner detection based on gradient covariance (Ando, 2000). The cross-correlation matrix is given as:

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} \begin{bmatrix} f_x & f_y \end{bmatrix} = \begin{bmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{bmatrix} \tag{4.12}$$

where S_{xx}, S_{xy}, etc. refer to the cross-correlations of the image derivatives f_x and f_y. Based on these cross-correlations, he defines two homogeneity measures:

$$P_{EG} = \frac{(S_{xx} - S_{yy})^2 + 4S_{xy}^2}{(S_{xx} - S_{yy})^2 + \sigma_{EG}^4} \tag{4.13}$$

$$Q_{EG} = \frac{4(S_{xx}S_{yy} - S_{xy}^2)}{(S_{xx} - S_{yy})^2 + \sigma_{EG}^4} \tag{4.14}$$

with σ_{EG}^4 being a small constant to avoid division by zero at completely flat areas in the image. The two measures show different behaviors: while P_{EG} reaches 1 where grayness varies one-dimensionally, i.e., near edges or ridges in the image, Q_{EG} reaches 1 where grayness variation is omnidirectional, i.e., at centers of circular symmetry or corners (F23–F26). For color images, we used the maximum of the two color results.

4.3.1.7 Local Cues

Martin et al. (2004)presented a set of local cues based on brightness, color, and texture information to observe boundaries in images. Their reasoning is that different types of boundaries exist in images. Traditional image gradient approaches only observe sharp changes in image intensities, so their techniques aim to provide gradient information on color and texture as well (F27–F29).

They achieve this by transforming RGB images to the CIELAB color space and separating the luminosity (l) from the color bands (a, b). To obtain local gradient information,

luminosity values in a local disk centered on each pixel and divided into two halves along a diameter are sampled. A dissimilarity between the two halves indicates the presence of a strong gradient and thereby a possible boundary. First, histograms of values in each disk half are created by sampling a kernel density estimate of the sampled values. The gradient is then obtained as the χ^2 distance between the two histograms:

$$\chi^2(P,Q) = \frac{1}{2}\sum_{i=1}^{N}\frac{(p_i - q_i)^2}{p_i + q_i} \tag{4.15}$$

with P and Q referring to the two distributions and p_i and q_i being the sampled probabilities. Local color gradients are obtained in the same fashion by applying the same procedure to both color channels and then adding the two marginal gradients to a joint gradient.

In the case of texture, the approach is based on what the authors refer to as "textons". At first, a filter bank of 13 different filters is applied to the neighborhood of each pixel. The filters contain a difference of Gaussian filters, as well as a set of elongated, oriented even- and odd-symmetric Gaussian second derivative filters. The responses of all filters are then clustered via k-means to obtain the so-called textons. The rest of the procedure is similar to the brightness and color gradients: textons are assigned to bins in a histogram and the local texture gradient is derived as the χ^2 distance between the two histograms of the disk halves.

In all three cases (brightness, color, and texture) the final gradient is obtained by taking the maximum gradient observed in 16 orientations in the range $[-\pi, \pi]$. As we did not train a separate classifier as in the original paper, we computed textons on individual image subsets. Due to the limited amount of data, we observed that a reduced number of 32 clusters (versus 64 in the original paper) was preferable.

4.3.2 Boundary Detection

4.3.2.1 Dataset Preparation

To create the training data, we considered areas in proximity to our reference data as a possible input. Boundary samples were taken from the rasterized polygon boundaries of the reference data and non-boundary samples were considered from a distance of up to 8 pixels from the boundaries. We avoided selecting non-boundary samples in the vicinity of the boundaries within a two-pixel-wide buffer to enhance the sample set. Datasets were balanced to contain 50% boundary and 50% non-boundary samples and then randomly split into calibration (60%, 310,928 samples), validation (20%, 103,642 samples) and test (20%, 103,642 samples) sets.

4.3.2.2 Model Development Setup

We used fully-connected MLP-NNs for boundary detection. We considered the task of boundary detection as a binary classification problem, distinguishing any type of field boundary pixel as represented by the reference data from any type of non-boundary pixels. This results in a rather heterogeneous non-boundary class, as agricultural fields, nearby forests, urban agglomerations, streets, etc. may end up being selected as non-boundary examples. Using additional classes by distinguishing different kinds of land cover may have been advantageous in terms of classification accuracy but was not considered in order to achieve a result that can serve as a direct input to the subsequent field extraction steps. The output node was scaled using a sigmoid function.

We used the stochastic gradient descent optimizer (SGD) with momentum of 0.9 to train the model using the log loss function (Hastie et al., 2009; Robbins and Monro, 1951):

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{4.16}$$

where $p(y_i)$ is the predicted probability (output value between 0 and 1) and y_i is the class label of the given sample point (0 or 1). The smaller the resulting log loss value, the better the prediction. Theoretically, log loss can become infinitely large when the prediction is worse than a purely random guess.

4.3.2.3 Input Feature Selection

As seen in Table 4.2, we obtained a total of 29 input features. We categorized them into 14 groups based on concept (see Section 4.3.1) and on image information used (luminosity vs. color). For each feature, we considered values from the three observation times in April, June, and September.

Feature selection was performed through quality ranking. We first trained a default model of topology x-30-1 with each set of input feature groups for 10,000 epochs and ranked results from best (1) to worst (max) based on validation loss. The overall rank of each input feature group was obtained by summing up the ranks of all test runs involving the respective feature group. For example, if the test run ranked 1 was based on groups 2, 3, and 7, then the total score of all three groups would be increased by 1. As a result, we obtained an overall ranking that represents the relative performance of each group in all test runs with the lowest total rank number representing the best-performing group.

We started with single groups as inputs, then pairs, triplets, quadruplets, and, finally, quintuplets. At each stage, we dropped the two lowest ranking feature groups. Results are presented in Section 4.4.1.1 as relative importance, i.e., the ratio of the lowest

total score achieved in the corresponding step to the scores of each group, resulting in importance scores between 0 and 1, where the best-performing group of each step achieves an importance of 1.

4.3.2.4 Hyperparameter Tuning and Model Training

We considered five hyperparameters for tuning: number of hidden layers, number of nodes per hidden layer, type of activation function in hidden layers, learning rate and dropout rate (random dropout probability) per hidden layer. Table 4.3 lists the considered settings.

Table 4.3. Hyperparameters considered (selected parameters in bold text).

Hyperparameter	**Considered Values**
Number of hidden layers	2, 3, **4**, 5
Number of nodes per hidden layer	15, 20, 25, 30, **35**, 40, 45, 50
Type of activation function	Sigmoid, **ReLU**
Learning rate	0.01, 0.001, **0.0001**, 0.00001
Dropout rate per hidden layer	20%, 30%, 40%, 50%, 60%, 70%, **80%**, 90%

We first performed a manual pre-testing to obtain some insights into the model behavior and the effects of different hyperparameter values to reduce the number of settings that need to be considered in the tuning stage. Our pre-testing showed that a single hidden layer was insufficient to obtain best results, although larger numbers of hidden layers beyond five also showed no advantage. Results for the number of hidden nodes were inconclusive, with no clear tendency towards larger or smaller numbers. Learning rates of 0.01 and below proved beneficial, as well as dropout rates of 20 % or higher, especially to avoid overfitting late in the training process.

We performed the tuning via a random search. We trained each model for 10,000 epochs and selected the one with the lowest validation loss as the parameter set for final model training. We trained and validated the model on the full calibration and validation datasets, respectively. The final model (four hidden layers of 35 hidden nodes each, rectified linear unit (ReLU) activation function, learning rate of 0.0001, and dropout rate of 80%) was trained for 100,000 epochs (Nair and Hinton, 2010).

4.3.2.5 Boundary Map Post-Processing

To further improve quality of the output, we transformed the probability map into a binary boundary map by setting a threshold of 0.5. We further excluded non-agricultural areas using CORINE land cover data of 2018 and the Land Cover DE product of the German Aerospace Center (DLR) and highlighted the boundaries of those areas to allow for the

contour extraction to detect them (Weigand et al., 2020; Büttner et al., 2017). To further refine the boundaries, we calculated the boundary strength as the share of boundary pixels in a 3×3 pixel neighborhood. This helps guide the contour extraction towards the center of the boundaries (see Section 4.3.3).

4.3.2.6 Quality Assessment

For final model evaluation, we employed multiple accuracy metrics. First, we used sensitivity (true positive rate), specificity (true negative rate), accuracy, and F_1 score as general indicators of model performance, calculated as follows (Hastie et al., 2009):

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.17}$$

$$Specificity = \frac{TN}{FP + TN} \tag{4.18}$$

$$Accuracy = \frac{TP + TN}{N} \tag{4.19}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{4.20}$$

where TP is the number of samples correctly classified as positive (true positive), TN is the number of samples correctly classified as negative (true negative), FP is the number of samples falsely classified as positive (false positive), and FN is the number of those falsely classified as negative (false negative). N represents the whole sample size. All four metrics range from 0 to 1 and 0% to 100%, respectively. A score of 1 or 100% indicates a perfect result.

Further, we obtained the area under the receiver operating characteristic (ROC) curve as a probability-based indicator of model performance. The ROC curve represents the trade-off between the true positive rate and false positive rate by plotting the two measures against each other for multiple thresholds. The area under the curve (AUC), i.e., its integral, is used as an indicator for a model's capacity to separate two classes in a binary classification (Hastie et al., 2009). An AUC of 1 represents a perfect distinction, while an AUC of 0.5 indicates a purely random classification.

4.3.3 Field Extraction

4.3.3.1 The Graph-Based Growing Contours Method

For field polygon extraction, we employ an active contours algorithm based on growing contours (or snakes) that we presented recently (Wagner and Oppelt, 2020; Velasco and Marroquin, 2003). Active contours in general were introduced for object delineation and are based on energy minimization (Kass et al., 1988). An initial spline or polyline follows an external and an internal energy, representing information from the image (e.g., gradient magnitude or local intensities) and the shape of the contour. While the former guides the snake towards the contour of interest, the latter influences curvature and the size of the snake, preventing it from creating discontinuous shapes and collapsing or expanding indefinitely.

However, agricultural field boundaries are rather complex networks of contours that are sometimes only loosely interconnected and can be quite irregular in shape. Therefore, using shape priors or any previous assumptions may limit performance of an algorithm in different environments with varying characteristics (field sizes, shapes, locations, etc.).

The growing snakes concept introduced by Velasco and Marroquin addressed common drawbacks of many active contour models regarding sensitivity to initiation, i.e., initial position and shape priors, as well as difficulties in extracting discontinuous contours of many separate or highly irregularly shaped objects (Velasco and Marroquin, 2003). To achieve this, the growing snakes are initiated on seed points on the contour of interest and grow along it. As the growth follows local maxima, it tends to create irregular shapes and requires repeated smoothing.

In our applications to field boundaries, however, we observed that growing snakes tended to get "off track" easily. As they follow local maxima of the input image, any irregularities, especially near corner points or intersections of multiple boundaries, may easily move them away from the actual boundary. Further, we observed limitations in representing sharp turns.

Following these insights, we developed a modified growing contours model called graph-based growing contours (GGC). Our main goals were to (a) improve the quality of extracted contours and reduce the need for repeated smoothing; (b) introduce adaptive behavior allowing the tracing of contours of different shapes (straight lines, curves, sharp changes in direction, etc.); (c) enable automatic branching to explore multiple parts of a boundary network automatically and represent crossings as well as T-junctions; (d) reduce "dead ends"; and (e) extract an interconnected network rather than separate contour segments.

To achieve this, we replaced the growth step of the original growing snakes with a more sophisticated approach based on graph theory principles. Instead of simply searching for a local maximum in a set of possible steps, we generate a local graph by sampling around each current end point at each growth step. The GGC method works at sub-pixel accuracy. We used a barycentric transform to obtain values at a sub-pixel level (Yiu, 2000).

The graph is defined by an inner radius r_{min}, an outer radius r_{max}, the number of vertices on the innermost circle n_i, and the number of evenly spaced circles n_c. The parameter n_e defines the number of connections of each vertex to vertices on the next-larger circle. Starting from the innermost circle, the number of points doubles with each circle to obtain a homogeneous pattern, in which each vertex on a circle has the same number of neighbors on the next circle (see Figure 4.3). Vertices in the direction of the previous step are masked to avoid retracing of contours.

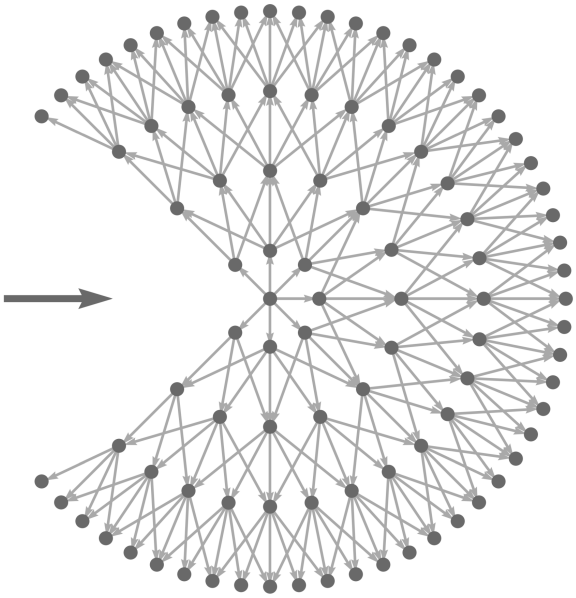

Figure 4.3. Example of local graph in graph-based growing contours. The large horizontal arrow indicates the direction of movement.

The sampled points are used as vertices in a directed, weighted graph $G = (V, E)$ as shown in Figure 4.3, where each edge (except for those originating at the center point) is weighted based on:

$$w_{i,j} = \frac{D_{i,j}}{g_j} \tag{4.21}$$

where $w_{i,j}$ is the weight of the edge connecting vertices v_i and v_j, $D_{i,j}$ is the Euclidian distance between the two vertices, and g_j is the gradient magnitude (or any other boundary strength indicator used) at vertex v_j. The resulting weights favor shorter paths (smaller Euclidian distance) and paths along boundaries in the image (higher boundary strength).

The different settings in constructing the graph (r_{min}, r_{max}, n_i, n_c, n_e) may be regarded as parameters for adjusting step size, precision, capacity to represent sharper directional changes, and complexity of the graph.

At each iteration, each current end point is explored for possible branches. Shortest path searches from the center vertex (source) to each vertex of the outermost circle (sinks) are initiated. Firstly, the single shortest path defines the first branch. The graph is then separated into two subsections at $+\frac{1}{2}\pi$ and $-\frac{1}{2}\pi$ from the first branch (see Figure 4.4). The shortest path in each segment is then considered as a further branch. Any branch exceeding a certain maximum length L_{max} is ignored to quickly omit possible "dead ends". If no valid path is found or the end point is near an already existing contour point or the image boundaries, the end point is terminated. As such, the number of end points changes with each iteration, dynamically exploring various contours.

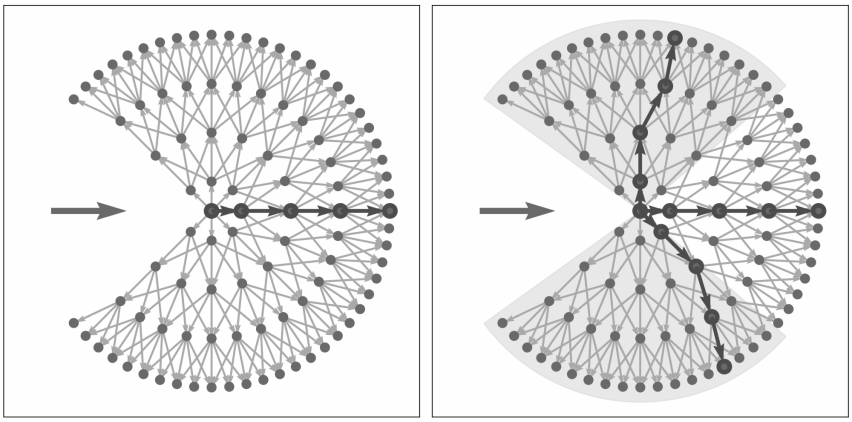

Figure 4.4. Main steps of movement in local graph: determine overall shortest path in graph (left); select shortest paths in subsections (right).

4.3.3.2 Modifications and Adaptive Masking

In our previous study, we observed two main drawbacks of the GGC method. Firstly, there is a significant trade-off between precision of the contour extraction and complexity of the local graph as expressed by the number of circles n_c per unit distance in a given step size r_{max}. Higher complexity dramatically increases the number of vertices and edges considered and consequently slows down the process. Secondly, different quality of boundary inputs to the GGC method requires a different behavior. If the boundary

map provided is inconsistent or accuracy is low, following the provided magnitude values may result in "zig-zag" curves and irregular shapes, even when considering distances (see Equation 4.21).

To address the complexity issue, we acknowledge that it is not necessary to consider the entire local graph at a given position. In fact, only those parts of the graph that cover actual boundary structures are of interest. Therefore, it would be desirable to exclude certain parts of the graph and thereby reduce the number of possible paths to explore in the movement step. We achieve this by first sampling the local magnitudes at all points of the graph. We then apply Otsu's method to obtain an optimal threshold, separating high and low values in the local neighborhood histogram (Otsu, 1979). By removing all sample points below the threshold, we effectively drop those parts of the graph that do not correspond to any boundary structures (see Figure 4.5).

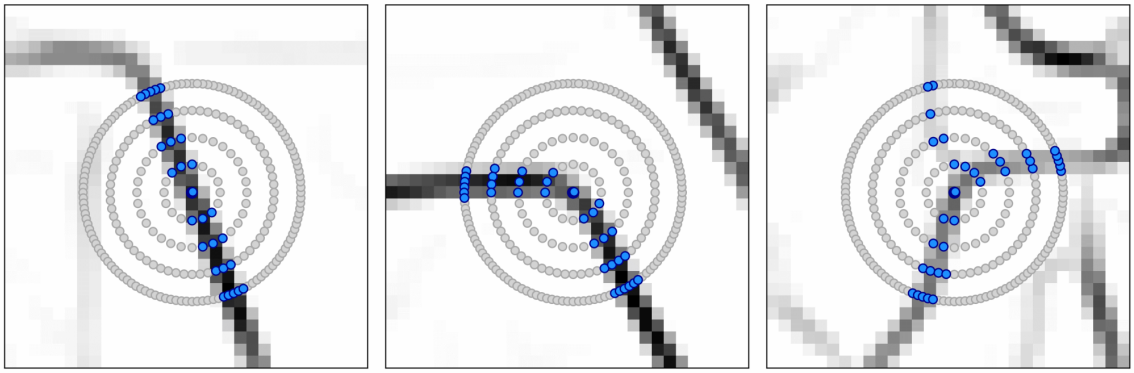

Figure 4.5. Examples for adaptive masking of vertices in the graph. Light grey points indicate removed points of the local graph, blue points represent those retained after masking.

The aspect of input quality can be addressed by modifying the weighting in Equation 4.21. If the input map is less reliable, a higher emphasis on shorter (and smoother) paths would be advantageous. If the input map is very accurate, however, the model may more closely follow the provided boundaries to create a more accurate result. Therefore, we introduce the factor β that can be used to increase or decrease the relevance of magnitudes relative to the distance between vertices:

$$w_{i,j} = \frac{D_{i,j}}{\beta \cdot g_j} \tag{4.22}$$

To ensure consistency, L_{max} is scaled by $\frac{1}{\beta}$ to allow using comparable values of Lmax, regardless of the β used.

4.3.3.3 Field Polygon Extraction

Using the output of the GGC model as an undirected network or graph of contour points, we intend to produce actual field polygons. Theoretically, finding cycles in the graph would be a possible approach to distinguishing different fields. The large size of the resulting graph, however, makes this task computationally demanding. Further, even a few missing connections may result in incorrectly created polygons or cycles not found.

Following our previous methodology, we therefore transform the contour output of the GGC method into a binary edge map (Wagner and Oppelt, 2020). We use a flood fill algorithm initiated on local maxima of a Euclidian distance transform from the binary boundaries to create a segment representing the rough extent of a field as described by the extracted contours. We then only consider those vertices close to the segment and a new over-connected unweighted and undirected graph is produced in which each vertex is connected to its four nearest neighbors. The field polygon is then obtained using the longest cycle in this network. Finally, polygons are refined using adaptive Gaussian filtering and the Ramer–Douglas–Peucker algorithm (Deng and Cahill, 1993; Douglas and Peucker, 1973; Ramer, 1972).

4.4 Results

We provide statistical and visual results for the boundary detection as well as the subsequent field polygon extraction steps. For the sake of brevity, we focus visual comparisons on study area 2.

4.4.1 Boundary Detection

4.4.1.1 Input Feature Importance

The step-wise feature importance ranking resulted in relative importance values listed in Table 4.4. As step five (quintuplets) did not allow for much further distinction in quality, we ultimately arrived at the six feature groups (in order of importance): G8, G4, G14, G1, G6, and G5 and the corresponding 13 features F17–F20, F6–F8, F29, F1, F11–F12, and F9–F10. Multiplying by the three time steps, this resulted in a total of 39 inputs for the boundary detection model.

This demonstrates texture metrics (G8), local statistics (G4), local cues (G14), gradient magnitude (G1), and Hessian matrix eigenvalues (G5 and G6) to be the most relevant for boundary detection. Overall, it seems that color information was significantly more

Table 4.4. Results of feature importance ranking. Values are given as relative importance values (ratio to best achieved for any feature group in all test runs of the respective step).

	Round 1	**Round 2**	**Round 3**	**Round 4**	**Round 5**
G1	0.20	0.19	0.30	0.49	0.83
G2	0.17	0.19	0.29	0.47	-
G3	0.25	0.19	0.28	-	-
G4	1.00	0.54	0.48	0.69	0.94
G5	0.33	0.22	0.29	0.51	0.75
G6	0.50	0.35	0.39	0.53	0.79
G7	0.10	0.18	-	-	-
G8	0.11	1.00	1.00	1.00	1.00
G9	0.07	-	-	-	-
G10	0.08	-	-	-	-
G11	0.09	0.18	0.26	-	-
G12	0.13	0.22	0.30	0.48	-
G13	0.08	0.17	-	-	-
G14	0.14	0.20	0.28	0.55	0.88

useful than luminosity, as in most feature types, those applied on color channels were preferred over luminosity ones. The only exception in this respect is image gradient.

Texture metrics on color images were particularly useful and continuously achieved the best ranks in all steps except the first one. This initial underperformance may be explained by similar results across all feature groups in the initial tests, except for the two "outliers", G9 and G10 (angular dispersion), which showed very little merit with validation loss barely better than a purely random guess. Further, Hessian matrix eigenvalues proved relevant in terms of both luminosity and color, while local statistics and local cues were again most useful in terms of color.

4.4.1.2 Performance Metrics

We evaluated the boundary detection model on the test dataset: once on the full dataset and once only on those samples that were actually agricultural according to the land cover maps. We made this distinction to analyze potential effects of basic "post-processing" based on prior knowledge of results.

As seen in Table 4.5, accuracies tend to be slightly better in study area 2 compared to 1. Nevertheless, results for both areas are similar, indicating a comparable performance of the model. Overall accuracies are reduced by a lower sensitivity while specificity is generally high. The high AUC values of 0.92 and greater indicate a very reliable distinction

of the classes. Masking of non-agricultural areas slightly increases performance in all metrics, reaching accuracies of up to 88.2% and AUC of 0.94.

Table 4.5. Accuracy metrics of the field boundary detection for the test dataset.

	Full Dataset			**Only Non-Agricultural Areas**		
Study Area	**1 & 2**	**1**	**2**	**1 & 2**	**1**	**2**
Sensitivity	77.9%	77.3%	79.1%	81.6%	81.3%	82.5%
Specificity	92.1%	92.9%	93.2%	93.5%	93.2%	94.4%
Accuracy	83.8%	83.4%	84.8%	87.2%	87.8%	88.2%
F_1	0.85	0.85	0.86	0.87	0.87	0.88
AUC	0.92	0.92	0.93	0.94	0.94	0.94

4.4.1.3 Visual Comparison

Figure 4.6 provides side-by-side comparisons of the predicted boundary probability map, the masked binary map, the boundary strength map, and the reference boundary map for three subsets.

The comparison shows that most boundaries are correctly detected and overall appearance is matched quite well, although extracted boundaries tend to be thicker than those drawn in the reference dataset. Problems, however, occur near urban areas (first example top right, third example top right). The heterogeneous structure of these built-up areas is frequently confused with actual field boundaries when they have not been masked using the two land cover maps. Similar issues occur near groups of small fields where the model struggles to separate multiple boundaries in close proximity. In some cases, the land cover based masking excluded some structures that may have been considered agricultural field boundaries (first example top center).

Further, there are cases where the model originally seemed to detect some weaker boundary patterns within the fields that were not part of the reference set (third example bottom left), or it missed some smaller boundary segments (first example center right).

4.4.2 Field Extraction Results

We compare the default version of the GGC method as presented in our previous publication and the new adaptive version (Wagner and Oppelt, 2020). Settings were selected through manual testing of different settings and visual interpretation of results. According to the results of manual testing, we chose the following setting: $r_{max} = 12$, $n_c = 8$, $L_{max} = 20$, and $\beta = 1.25$.

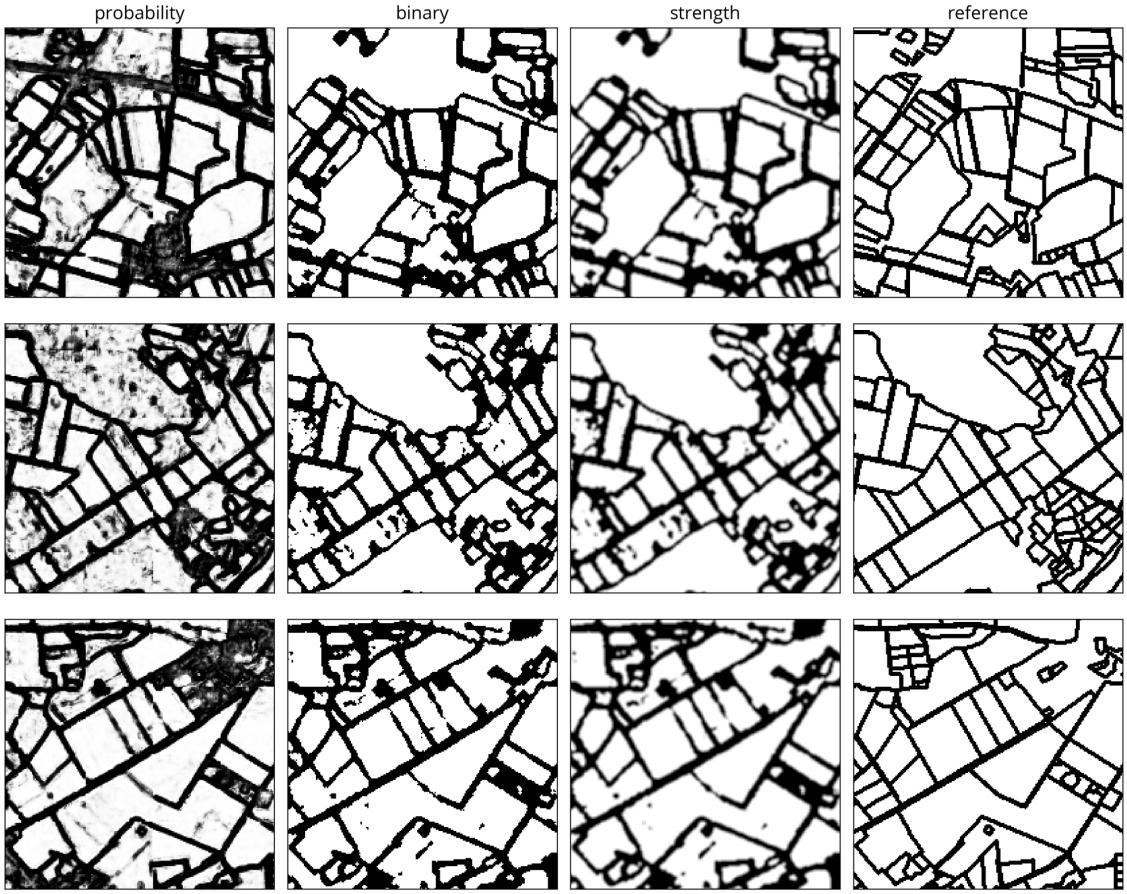

Figure 4.6. Comparison of boundary detection results in study area 2. Columns represent the model probability output, the binary output after land cover masking, the final boundary strength map, and the final reference dataset (boundaries thickened for visualization). Rows represent different subsets.

4.4.2.1 Visual Comparison

As seen in Figure 4.7 and Figure 4.8, distribution and general shapes of extracted polygons often closely resemble those of the reference polygons. However, shapes tend to be a bit smoother and edges more rounded (e.g., Figure 4.7 and Figure 4.8 bottom center). Smaller fields, in particular, were occasionally missed or merged with larger fields (e.g., Figure 4.7 and Figure 4.8 top center). This can be explained in part by issues in the prior boundary detection step, e.g., inconsistent boundaries or some falsely detected boundaries within the fields. Other fields were also missed by the process, again partly due to the boundary detection step (e.g., Figure 4.7 and Figure 4.8 bottom left).

In particular, clusters of small fields pose problems (see Figure 4.7 and Figure 4.8 bottom left). As mentioned in Section 4.4.1.3, boundaries are sometimes difficult to delineate here and the subsequent extraction may be confused if multiple adjacent boundaries

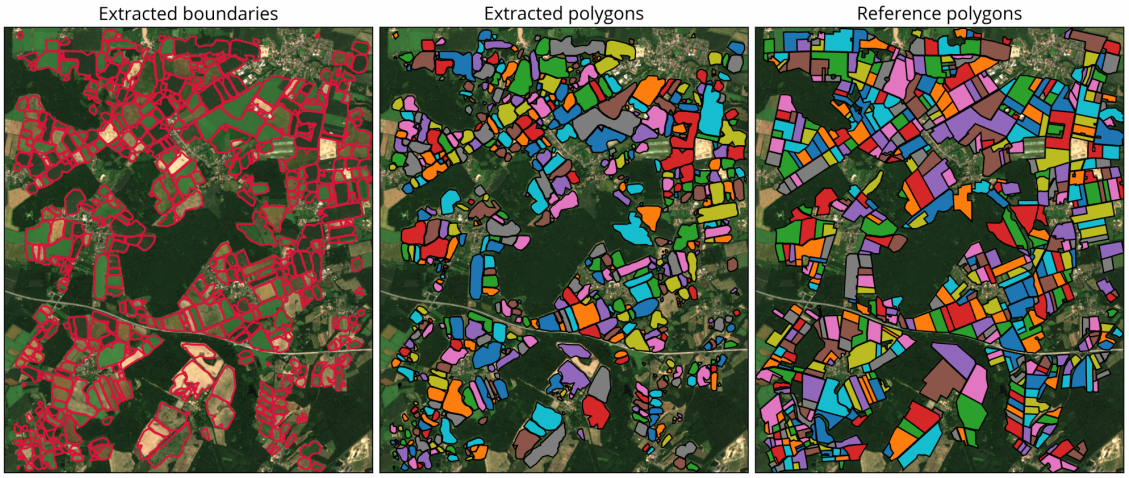

Figure 4.7. Result of the adaptive graph-based growing contours (GGC) field polygon extraction.

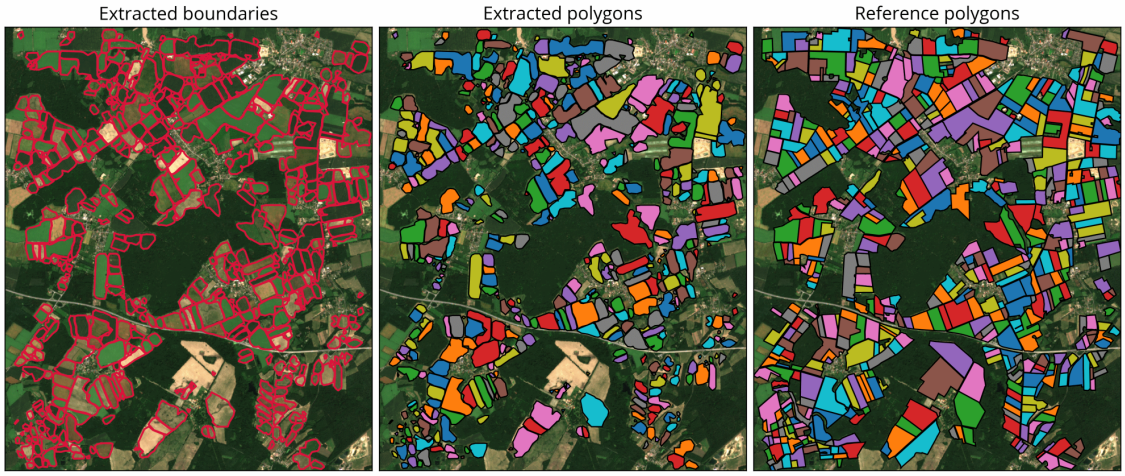

Figure 4.8. Result of the non-adaptive GGC field polygon extraction.

are merged. Additionally, large gaps within fields due to larger non-agricultural areas, such as clusters of trees, occasionally led to errors in the extraction (e.g., Figure 4.7 and Figure 4.8 bottom center), confusing the extraction process.

When comparing the adaptive and non-adaptive version, the results look very similar. Differences are visible in some polygons being detected by only one of the versions (e.g., Figure 4.7 and Figure 4.8 center and bottom center). In many cases, the adaptive version seems to be a bit less likely to miss fields entirely.

4.4.2.2 Processing Speed and Upscaling

Our testing revealed the actual processing time to be highly dependent on the implementation. Therefore, the results in this section are presented as a relative comparison of the adaptive and non-adaptive versions rather than a reference for actual processing time. The test runs were performed on an Intel Core i7-4790K CPU with four cores at 4.0 Ghz (4.4 Ghz turbo boost) in single-core execution and 32 GB of memory. Table 4.6 shows an overview of execution speed and the number of contour points extracted as an indication of the complexity of the given scene.

Table 4.6. Comparison of processing speed and number of extracted contour points in adaptive vs non-adaptive GGC for differently sized areas. Processing acceleration indicates the reduction in processing time of the adaptive version in relation to the non-adaptive version.

Settings	r_{max}**=5,** n_c**=4,** L_{max}**=14**		r_{max}**=12,** n_c**=8,** L_{max}**=20**	
	Acceleration	**Points extracted**	**Acceleration**	**Points extracted**
2.5×2.5 km² subset	1.3×	~5,500	2.2×	~3,800
5.0×5.0 km² subset	1.3×	~20,000	2.2×	~14,000
10.2×7.5 km² subset	1.3×	~55,000	2.3×	~40,000

Results demonstrate the adaptive version achieved a substantial acceleration compared to the non-adaptive version at all subset sizes and settings. The advantage is more substantial for larger graphs.

4.5 Discussion

Analysis of the selected input features revealed that most features obtained from color channels outperformed those based on luminosity. Sequentially dropping the lowest performing groups highlighted texture metrics, eigenvalues of the Hessian matrix, local statistics, local cues, and gradient magnitude as useful. Homogeneity measures, however, showed poorer performance, and angular dispersion did not provide enough information to obtain good boundary detection. It may be interesting for future applications to further explore the effect on performance if the groups dropped were added back into the input set or comparisons were performed on an even more granular level evaluating each individual feature.

The MLP-NN model proved capable of accurately detecting field boundaries. Hyperparameter selection showed multiple layers to be advantageous, combined with a high dropout rate of 80% and a learning rate of 0.0001. Results of the boundary detection model were good, with high accuracies that could be further increased by removing non-

agricultural areas from the output. This is particularly promising when considering the input being solely comprised of RGB channels. For both study areas combined, we obtained an accuracy of 87.2%, F_1 score of 0.87, and AUC of 0.94 after removing non-agricultural areas. These results also seem competitive when compared to another deep learning approach recently presented by Masoud et al. (2019). They proposed a field boundary detection based on convolutional NNs trained on S-2 10-m and 20-m resolution imagery of multiple test sites in the Netherlands. They reported an overall best F_1 score of 0.67 and an average F_1 score of 0.63 for all test sites. Further, the overall best sensitivity and specificity values were 0.69 and 0.66, respectively. However, a direct comparison may be difficult, as they used more bands, including near-infrared channels, and merged different spatial resolutions, but did not consider multi-temporal imagery in their approach. This may partly explain the difference in performance. Another approach based on convolutional NNs was presented by Persello et al. (2019). They reported performance similar to that obtained here with F_1 scores of up to 0.830 when using WorldView-2/3 data for field delineation in smallholder farms in Nigeria and Mali.

Visual comparisons further revealed that the boundary detection sometimes observed boundaries that were not part of the reference dataset, while it missed other parts. In some of these cases, it is debatable if results are objectively wrong, as interpreting certain structures can be difficult if they only occur at some time of the growing season, e.g., due to field management or cropping patterns. The boundaries produced tended to be broader than a precisely hand-drawn delineation and may appear "blurred". This is mostly a result of the appearance of many of the input features, such as gradient magnitude, that do not provide a sharp distinction but rather a gradual highlighting of discontinuities in the image. As the model predictions, however, were intended as an input to the following GGC method, they are not required to be flawless. The β value > 1 selected as best for the extraction step further indicates a high quality of the boundary detection result. For other applications, further post-processing or modifications to the modeling step may be required.

Another advantage of the boundary detection approach presented here is that it is scalable and should be relatively easy to transfer to other regions and landscapes. The inputs to the model are universal image features and exclusively based on regular RGB imagery, enabling an adaptation to other satellite, unmanned aerial vehicle (UAV), or airborne sensors. Further research may also explore the use of higher resolution imagery that offers more accurate field delineations and helps resolve more detail to effectively extend the process to even smaller field sizes.

Field extraction results were also promising and proved to work well when given a consistent, high quality input map. This also allowed the use of a much larger step size than in our previous study. The new adaptive version of GGC obtained very similar

results as the non-adaptive one, while drastically reducing processing time. It is therefore more suited for larger-scale applications. A further advantage could be that settings such as L_{max} may not be as crucial as in the previous version because the automatic masking step removes much of the ambiguity of a heterogeneous environment.

Nonetheless, despite the improved detection step, groups of very small fields or very heterogeneous structures posed problems in field polygon extraction. Fields were occasionally merged or missed entirely. While it also did not extract all fields accurately as represented in the reference set, the adaptive version seemed to suffer slightly less from those issues. One reason for this may be that masking less relevant parts of the graph also avoids following erroneous paths. For example, cases in which there are small spots of high boundary strength within a field due to false detections can result in a path being selected that attempts to connect to this point, even if other vertices on the way should clearly not be considered boundaries. By masking weaker vertices, the adaptive version may effectively avoid any possible path to such falsely detected boundary spots.

Another observation was that extracted fields tended to be more rounded in shape and sharp 90° angles were occasionally missed. This is in part due to the field extraction steps following the GGC application (see Section 4.3.3.3). The over-connected nature of the graph used as a basis for polygon creation can result in rounded shapes when connections "cut out" the actual corner points. Therefore, further research may be needed to improve the transferring of the GGC output to polygons.

Furthermore, it may be interesting to explore other applications of the GGC methodology outside of field delineation. As it is fundamentally a growing active contour technique and independent of the context or the type of input given, its application may extend outside its original use case.

4.6 Conclusions

We demonstrated a new field boundary detection approach based on deep learning and explored a large number of image features obtained from S-2 RGB imagery. We merged it with the GGC method for field polygon extraction and further introduced an improved version of the GGC approach that increases scalability through automatic masking of the local graph to reduce complexity without affecting quality of the output.

The analysis of a large number of possible input features revealed interesting insights regarding their usefulness for this particular task. While some features such as local statistics, texture metrics, image gradient, Hessian matrix eigenvalues, and local cues were valuable inputs, others such as angular dispersion and homogeneity measures were less useful. Color information was more valuable than luminosity in most cases. The use

of these universal image features may also facilitate adapting the methodology to other sensors and resolutions.

The boundary detection model obtained high accuracies but tended to produce broader boundary lines. The quality of the output map benefited from post-processing based on land cover masking. Limitations of the results were, to a large extent, successfully mitigated by the subsequent GGC method. Nonetheless, issues remain regarding performance on non-agricultural areas in case there is no land cover mask available. Further research may be needed, including closer analysis of input features (settings, types of features to use, etc.), as well as exploring different types of classification models. The use of different time steps and the relevance of multi-temporal compared to single image classification may also be further investigated.

The new, adaptive version of GGC proved to be much faster than the original version without compromising quality of the extraction result. A logical next step may be to apply the technique at an even larger scale, e.g., at county or state level. Beyond that, the nature of the GGC approach also allows exploring further applications outside of field extraction and agricultural remote sensing.

Acknowledgments

We thank the students and student assistants who carried out in-situ mapping campaigns and helped produce and update the reference data.

Chapter 5

Synthesis

5.1 Main Achievements

The aim of this thesis was to address three main topics: (1) remote sensing data assimilation in dynamic crop models, (2) agricultural field boundary detection, and (3) field contour and polygon extraction. This was done by developing multiple new methods. The main results are discussed in the following subsections.

5.1.1 Remote Sensing Data Assimilation

> **Research objective 1:** Enabling flexible representation of different measurement- and model-related uncertainties in an assimilation technique for remote sensing data in dynamic crop models.

Chapter 2 addressed the topic of remote sensing data assimilation by incorporating Sentinel-2 derived canopy cover (CC) data into the AquaCrop-OS model to improve yield prediction in winter wheat. The described data assimilation approach proved comparable in performance to the Extended Kalman filter that served as a reference. In particular, it was capable of reducing bias in the model results.

The method successfully incorporated multiple sequentially added uncertainties indicating a capacity for flexible error handling. The weighting factor α was introduced to adjust importance of more dissimilar uncertainty probability density functions in the updating process, allowing for the user to decide on the degree of influence of individual uncertainties and observations considered. This weighting, however, was shown to be dependent on the application level (pixel vs field vs aggregated). This implies the need for prior experimentation or testing by the user to select the appropriate value for optimal results. Furthermore, the implementation presented in the study relying on Monte Carlo

simulations and kernel density estimation required significant prior processing and was therefore computationally quite expensive. This may aggravate application in some cases and make the method less appealing for operational use.

Both these limitations were discussed and addressed in the study. Firstly, the adaptive version in which not only the optimal Gaussian distribution but also the weighting factor α is determined by the optimization process, demonstrated good results on all scales without further user input. Secondly, the flexibility of the optimization procedure allows for use of different kinds of uncertainty representations and does not require the computationally demanding approach chosen for this example study. Especially if more knowledge about measurement and model uncertainties is available, extensive processing can be avoided, for example, by using functional representations.

Overall, yield prediction results in this study were not particularly accurate which may have multiple reasons. Firstly, despite local calibration the AquaCrop-OS model may struggle to accurately represent conditions in the region. Secondly, the lack of in-situ measurements and reliance on coarser scale weather inputs meant that the only source of spatial dynamics in the model was the remote sensing data assimilated on only a few time steps of the simulation. Thirdly, the local accuracy of the input data is largely unknown and may cause an over- or undercorrection of the default model simulation (see below). Results on field-level were better than those on pixel-level which further suggests aggregation effects.

There was some concern about the performance with respect to R^2 scores that were also discussed extensively in chapter 2. The reasons for the poor R^2 performance throughout all applications and assimilation techniques are still not fully understood. It is likely, though, that issues originated from the CC input data that was used because all tested methods were equally affected. Possibly, a different way of representing the uncertainty in these datasets would have been advantageous to better highlight inconsistencies or unreliable data points. However, necessary background information was lacking as no reference measurements in the fields were available to serve as benchmarks to evaluate actual performance on location.

5.1.2 Field Boundary Detection

> **Research objective 2:** Developing field boundary detection methods based on limited image information (RGB imagery) and exploring the usefulness of image features for improved detection performance using deep learning models.

Chapters 3 and 4 addressed the topic of field boundary detection in two different ways. In chapter 3, a workflow was presented that relies primarily on universal image pro-

cessing techniques and RGB inputs (hereafter referred to as “image processing approach”). This was intended to ensure a high flexibility in adapting the method to different sensors and landscapes. The methodology described in chapter 4, however, took a different approach by expanding on the image processing aspect and employing deep learning techniques to train a more powerful classifier for field boundary detection (hereafter referred to as “deep learning approach”).

The combination of enhanced multi-temporal imagery with adaptive noise reduction, edge detection and homogenization in the image processing approach worked well on local levels but required parameter tuning to adapt to conditions within a certain subset. The test area used in this study was quite heterogeneous with very irregular field sizes and shapes and especially temporally and spatially irregular soil patterns (incl. water-logging) making it a difficult example region for such an application. Nevertheless, the fact that local parameter tuning was necessary even when comparing different regions of interest within the same study area, demonstrated the need for a more universal approach.

In general, the inconsistent nature of natural boundary features poses problems. Field boundaries may consist of delineations between adjacent fields or to other land cover types such as forests, rivers and lakes or urban areas and streets. They may comprise of only narrow strips of grass or bare soil, hedges and trees, or man-made objects like fences, roads and buildings. Some boundaries are also difficult to interpret due to changing appearance over time or limited visibility.

The varying characteristics of field boundaries lead to significantly different behavior in the imagery as well. A boundary originating from an adjacent road, for example, shows a similar signature throughout the year, while a management pattern may disappear at later crop growth stages. Moreover, multiple kinds of patterns may occur along a single boundary between two fields. A human observer may easily interpret these as one consistent boundary but a local filtering technique may produce a very heterogeneous result consisting of stronger and weaker boundary parts. These issues raise multiple questions: Should a field that is managed separately but grown with the same crop be considered as two separated or one uniform field? How should adjacent fields that are temporarily distinct but homogeneous at other times be handled? Should weak, barely visible boundaries be treated the same way as more obvious ones? Do different growth patterns suffice as a reason to separate a field into two parts? Unfortunately, there is no definitive answer to these questions introducing a certain ambiguity to the task and the interpretation of results. This proved particularly problematic in the subsequent contour extraction (see chapter 5.1.3).

These experiences led to the development of the deep learning approach based on universal image features extracted from three time steps in the growing season. The

study was intended to not only present a potential method for field boundary detection and its suitability as a pre-step for contour and polygon extraction, but also to allow insights into the usefulness of different image features that may serve as input for other boundary detection techniques. The topics of edge detection, contour extraction and semantic segmentation have been under extensive research in the image processing field (Cheng et al., 2001; Papari and Petkov, 2011; Bansal et al., 2013). By selecting an extensive list of features, separated by concept and type of image information, it was possible to establish the different value of these inputs for the specific task of field boundary detection. The analysis showed that those features based on color information were often superior to those based on luminosity. The best-performing features were those based on second-order texture metrics, local statistics, gradient magnitude, local cues and Hessian matrix eigenvalues.

Results of the presented model were good and comparable to others in recent literature. Nevertheless, it still required some post-processing to achieve best results. Aside from superior performance, the model has the advantage of fast execution and more homogeneous boundary results than the image processing approach. This manifested itself in a more consistent contour extraction performance with no local adaptation needed.

Furthermore, its concept makes it adaptable to other regions as well. Although it has not been tested yet, one can assume that the universal nature of the image features used would work in a similar way in other regions. The fact that the model was trained on two separate areas and performed very similarly in both, further supports this assumption. Nevertheless, when used in a different environment, a new model would have to be trained.

As reference data was only available for one year, it was also not investigated if a model trained on data from one year could perform similarly in following years. In theory, the concept would allow for such a temporal transfer but in practice it may prove challenging. Due to the fact that growing patterns may change and onset and end of the growing season can shift year by year, the multi-temporal nature of the model may limit its transferrability to subsequent years. In addition, remote sensing observations may not always be available around the time needed. It may be interesting to investigate, how much of an impact slight variations in seasonal patterns or different observation dates have on performance.

5.1.3 Contour and Polygon Extraction

Research objective 3: Developing a method for effective automatized extraction of complex, heterogeneous boundary networks and subsequent creation of field polygons at sub-pixel level with minimal supervision.

In chapter 3 the new Graph-based Growing Contours (GGC) method was presented. It is a modification of the growing snakes model by Velasco and Marroquin (2003) and was designed specifically for extracting heterogeneous, interconnected networks of contours which are very common in agricultural landscapes. The adaptive nature of the technique allows automatic branching and exploration of multiple boundaries while conserving spatial relationships and locations of contour points for subsequent polygon extraction steps. The user has extensive control over the behavior of the model by modifying its central element, the local graph used to define movement at contour end points.

The method itself is a generally applicable contour extraction method from the family of active contours. It is not constrained to remote sensing imagery or agricultural applications but may be adapted to different contexts. A major advantage is its flexible, mostly unsupervised execution. Even initiation on a single seed point was often sufficient for automatic extraction of the entire boundary network of a larger image subset. Once started, GGC does not require any manual input and returns a full set of extracted contour points of theoretically arbitray precision (i.e. point spacing).

However, the trade-off between precision and complexity is one of the central issues to be addressed. As became obvious in the first applications, the extraction is sensitive to the input data. Although it managed to handle heterogeneous outputs of the image processing approach to boundary detection, it was incapable of fully compensating for its flaws. The partly inconsistent nature of boundary detection results made the choice of settings difficult and required small step sizes to avoid confusion by local heterogeneities. This was a main reason for the need of local parameter optimization. Given a more reliable input, such as the deep learning results, it proved to be significantly more consistent and did not require extensive parameter tuning.

Performance was further aided by the modifications presented in chapter 4. By adaptively masking areas considered for contour growth, contour extraction got less susceptible of getting “off track”. This also led to a significant speed-up through reducing the number of vertices considered in the movement step allowing for higher precision at a given processing time. As a side effect, the relevance of maximum path length was reduced as many “wrong paths” were already excluded in the prior masking step, effectively reducing the risk of dead ends that previously caused indents and “zig-zag” patterns in extracted polygons.

The subsequent polygon extraction based on a flood fill algorithm and local selection of contour points was fast and reliable. Resulting polygons were sometimes more “rounded” than the actual fields which is often caused either by errors in the boundary detection at corner points or the tendency of GGC to follow shorter paths that do not trace the exact corner but form a “shortcut”. This was partly addressed by adding a

user-defined factor influencing the relative importance of boundary strength compared to shorter Euclidian distance in the weighting of the edges in the local graph.

On top of the already mentioned means of speeding up the contour extraction (local masking), a crucial aspect in processing speed is the choice of seed points for the polygon extraction step. Original implementations used seed points arranged on a regular grid across the image (e.g. 10×10 pixels). However, this poses issues when attempting to cover many fields of different size and shape. A very fine grid ensures that all fields are covered but also dramatically increases the number of seed points to consider and, as a result, processing time. A looser grid risks missing smaller fields or even larger ones if they happen to lie in between the grid spacing which is especially likely when they are irregularly shaped. A good compromise was found by searching for local maxima on a Euclidian distance transform to the extracted boundaries. This effectively locates seed points roughly at the center of a field at the furthest distance from the boundaries and proved reliable in covering even heterogeneous field arrangements.

5.2 Potential Synergies

Many applications in agricultural modeling and remote sensing benefit from an object-based rather than pixel-based application. This was also supported by findings in chapter 2 where field-level yield prediction was significantly better than pixel-level predictions. Therefore, applications like crop type distinction and land cover classification, crop monitoring and agricultural statistics are often performed on clusters or segments rather than individual pixels (Ma et al., 2017; Weiss et al., 2020). Of course, in an ideal case extracted segments would accurately represent individual fields.

One of the original motivations behind the field boundary detection and field extraction studies was to provide functionality for achieving a better segmentation that actually represents separate fields in the landscape. This is particularly relevant for the application of dynamic crop models that are not designed to run efficiently on a large scale, for example, a whole Sentinel-2 scene. If it was possible to correctly delineate individual fields within the agricultural areas, it would dramatically reduce the number of model runs required to adequately cover even large regions with more accurate field-level predictions rather than simulations on randomly sampled pixels or coarsely distinguished clusters.

Existing large-scale applications of dynamic crop models such as the Crop Growth Monitoring System (CGMS) often rely on highly aggregated input on the scale of multiple km^2 (van Diepen et al., 1994). A combination of a dynamic crop model applied on field-level based on extracted fields on a large regional scale may prove an effective way of exploiting high-resolution satellite imagery and allowing for much higher spatial detail

while mitigating the high computational demand and poor scalability of many dynamic crop models.

This idea may be further expanded by incorporating a crop type classification in the process to detect crops grown on the extracted fields and adjust the crop model parametrization accordingly. The flexible nature of the data assimilation technique presented here may also allow to incorporate different uncertainties depending on crop type and environmental characteristics.

Another advantage of field-level analyses is that most agricultural statistics and available information is provided on field- or farm-level. Therefore, incorporating additional information about management, fertilization, soil etc. is much easier on field-level than gaining access to high-resolution sub-field level measurements. The better data basis opens up more research paths and more detailed modeling.

5.3 Conclusions and Future Research Challenges

All presented techniques showed promising results and potential for synergetic use that may be explored in future studies. Multiple opportunities remain, however, to explore new applications as well as methodological changes.

The data assimilation approach may require some further improvements, regarding mainly two aspects: less computationally expensive ways of representing uncertainties and methodological improvements to the underlying statistical techniques and its adaptive behavior.

The adaptive version presented in chapter 2 was already promising but further improvements should be investigated. Application of the technique to different situations, models and variables may reveal more insights into the appropriate incorporation of highly dissimilar input uncertainties and handling of multiple variables at once. Establishing rules for selecting or adjusting the weighting factor α would be useful as well in case user interaction is strictly required or desired.

Further improvements to the method could lie in adding skewness and/or kurtosis to the representation of the optimal Gaussian distribution that is used for balancing the uncertainties. Especially the former may be valuable in a case as presented in chapter 2 where multiple observations/uncertainties show very similar behavior with only one being a drastic outlier (in this case usually the remote sensing input). Allowing for skewness in the optimal distribution could potentially improve the capability of successfully balancing out multiple different and highly unevenly distributed uncertainty probability density functions. An extended comparison of different statistical distance metrics and a more

detailed analysis of their effects on the assimilation may be informative as well. Similarly, modifications of the objective function and its penalties could be further investigated.

As already discussed in chapters 2 and 5.1.1, the uncertainty quantification part of the study may not be a best practice for other applications as it required extensive pre-processing in the form of Monte Carlo simulations and kernel density estimation. It would be interesting to see how the data assimilation technique behaves with different types of uncertainty functions. Although, in theory, it does not put constraints on the type of uncertainty representation being used, it would be important to know what effects this may have. Further research may also address comparisons to other data assimilation methods such as the particle filter or the ensemble Kalman filter that are also commonly used for updating of crop models.

The field boundary detection studies also implied multiple ways for further research and improvement. A first path of improving would be to make them independent of multi-temporal observations. The need for multi-temporal data poses a limitation in regions with few observations due to cloud cover or in applications to other sensors with longer revisit times. An ideal boundary detection would therefore work on single images of arbitrary timing. The limitations discussed in chapters 3 and 4, however, have to be overcome.

First of all, agricultural landscapes change dramatically in appearance not only from one year to the next but also throughout the growing season. Fields that may be easily distinguishable outside or early in the season may be almost indistinguishable late in the season, and vice versa. These effects are further influenced by location, weather, temporary soil and growth patterns, yearly changing growing conditions, as well as predominant crop types in the region of interest. Therefore, if only a single image is considered, a detection algorithm must work on very limited information aggravating not only detection on the current image but also successful transfer of information to following years.

The most straightforward solution to this is the limitation of model development to very narrowly defined constraints, for example, the use of imagery from only one specific time of the year or a specific time in the growth cycle of certain crops. Defining these constraints, however, is in and of itself difficult and ultimately limits the applicability of the model. If no imagery is available for the specified timeframe or the exact timing cannot be precisely determined (e.g. due to unusual growing conditions during exceptionally hot/cold or dry/wet summers), the model may not be applicable at all.

Another, more comprehensive approach may be to allow the model to learn under many different conditions, i.e. creating a training dataset that covers different times of the year and even extreme conditions. However, this makes the model much more complex and training more difficult. Handling the drastically varying inputs may ultimately prevent the model from achieving comparable performance to a more specialized one. It would

also require a large amount of reference data over the course of multiple years which is usually not available.

Instead, it is imaginable that an ensemble of models may be a good solution. By training multiple models on data from different time steps within the growing season, the ensemble as a whole may be more flexible in handling data gaps within the season or changing patterns from year to year.

A second path of improvement would be the distinction of different land cover types. The presented techniques relied on prior masking of non-agricultural landscapes. This is, however, not always possible as recent accurate land cover maps are not always available. It is therefore desirable to make models work largely independently of ancillary data. Introducing a distinction of agricultural from non-agricultural land could also be a possible way to improving performance as a whole, especially for the deep learning approach. As the training datasets so far consisted only of boundary and any type of non-boundary samples, the non-boundary class is very heterogeneous mixing samples from fields, nearby forests, urban areas and possibly even water bodies. Adding detail by separating parts of this large group of samples may faciliate more accurate distinctions.

A third path of improving boundary detection may be to introduce shape and consistency in the training rather than a pixel-by-pixel detection. This may make the detection more "human-like" as it ensures that coherent structures are formed and gaps within boundaries are avoided. Ideally, local discrepancies or heterogeneous boundary strength do not prevent a boundary from being recognized in full (see chapter 5.1.2). Furthermore, this could lead to more precise delineation in contrast to a more "blurred" out appearance as is often the case in the presented approaches. Convolutional Neural Networks may be the better choice in this respect.

The existing approaches could further be explored in the context of other sensors, including different spatial resolutions and spectral wavelength bands. As mentioned in chapter 3, a spatial resolution of 1-5 m may be optimal for this type of application. Lower resolutions do not allow for precise delineation under certain circumstances and suffer from lack of detail in environments with very small fields or varying boundary structures, while higher resolutions would probably introduce mostly unnecessary additional information that would lead to more confusion than added value for the classification. The low requirements for input data (RGB) could enable application of the methods presented in chapters 3 and 4 to a variety of higher-resolution sensors, including UAV and airborne imagery.

Regarding contour extraction, further challenges will lie in improving the trade-off between precision and processing time. As discussed in chapters 3 and 5.1.3, higher accuracy of extracted contours in the Graph-based Growing Contours approach requires

a much more complex local graph. Similarly, less reliable boundary detection inputs require smaller step sizes and higher detail. Although this was already mitigated by the modifications introduced in chapter 4, further research may discover other, more effective ways.

Another important aspect is the polygon creation out of the extracted contours. Issues remain in handling very small fields and the mentioned tendency of producing "rounded" shapes and indents. However, arguably, most of these issues do originate from remaining inconsistencies in the boundary detection results and the overconnected graph that is created in the current implementation of the polygon creation step. Additionally, few remaining dead ends created by the contour extraction step lead to errors in the polygon representation.

Finally, all methods described in previous chapters may be applied in different contexts and research fields. The data assimilation approach does not need to be limited to updating of a single state variable in dynamic crop models but could be applied to incorporating, for example, soil and plant measurements or information on external stresses with the goal of accurate plant health or canopy development simulation rather than biomass and yield prediction. It may also be applied to completely different types of models entirely.

The contour extraction and polygon creation techniques may even be applied to topics outside of remote sensing or agricultural monitoring. The image processing basis of the GGC method allows the application beyond its original scope to any kind of imagery for different purposes, including natural or medical images.

References

Aganj, I., Lenglet, C., Jahanshad, N., Yacoub, E., Harel, N., Thompson, P. M., and Sapiro, G. (2011). A Hough transform global probabilistic approach to multiple-subject diffusion MRI tractography. *Medical Image Analysis*, 15(4):414–425.

Alexandratos, N. and Bruinsma, J. (2012). World Agriculture towards 2030/2050: The 2012 Revision.

Ando, S. (2000). Image field categorization and edge/corner detection from gradient covariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):179–190.

Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (2016). Produktspezifikation für ALKIS-Daten im Format Shape. Technical report, Berlin.

Asseng, S., Ewert, F., Martre, P., Rötter, R. P., Lobell, D. B., Cammarano, D., Kimball, B. A., Ottman, M. J., Wall, G. W., White, J. W., Reynolds, M. P., Alderman, P. D., Prasad, P. V., Aggarwal, P. K., Anothai, J., Basso, B., Biernath, C., Challinor, A. J., De Sanctis, G., Doltra, J., Fereres, E., Garcia-Vila, M., Gayler, S., Hoogenboom, G., Hunt, L. A., Izaurralde, R. C., Jabloun, M., Jones, C. D., Kersebaum, K. C., Koehler, A. K., Müller, C., Naresh Kumar, S., Nendel, C., O'leary, G., Olesen, J. E., Palosuo, T., Priesack, E., Eyshi Rezaei, E., Ruane, A. C., Semenov, M. A., Shcherbak, I., Stöckle, C., Stratonovitch, P., Streck, T., Supit, I., Tao, F., Thorburn, P. J., Waha, K., Wang, E., Wallach, D., Wolf, J., Zhao, Z., and Zhu, Y. (2015). Rising temperatures reduce global wheat production. *Nature Climate Change*, 5(2):143–147.

Asseng, S., Ewert, F., Rosenzweig, C., Jones, J. W., Hatfield, J. L., Ruane, A. C., Boote, K. J., Thorburn, P. J., Rötter, R. P., Cammarano, D., Brisson, N., Basso, B., Martre, P., Aggarwal, P. K., Angulo, C., Bertuzzi, P., Biernath, C., Challinor, A. J., Doltra, J., Gayler, S., Goldberg, R., Grant, R., Heng, L., Hooker, J., Hunt, L. A., Ingwersen, J., Izaurralde, R. C., Kersebaum, K. C., Müller, C., Naresh Kumar, S., Nendel, C., O'Leary, G., Olesen, J. E., Osborne, T. M., Palosuo, T., Priesack, E., Ripoche, D., Semenov, M. A., Shcherbak, I., Steduto, P., Stöckle, C., Stratonovitch, P., Streck, T.,

Supit, I., Tao, F., Travasso, M., Waha, K., Wallach, D., White, J. W., Williams, J. R., and Wolf, J. (2013). Uncertainty in simulating wheat yields under climate change. *Nature Climate Change*, 3(9):827–832.

Atzberger, C. (2013). Advances in remote sensing of agriculture: Context description, existing operational monitoring systems and major information needs. *Remote Sensing*, 5(2):949–981.

Bansal, A., Kowdle, A., Parikh, D., Gallagher, A., and Zitnick, L. (2013). Which Edges Matter? In *IEEE International Conference on Computer Vision Workshops, 2-8 Dec*, Sydney, NSW, Australia. IEEE.

Bansal, S., Gupta, D., Panchal, V. K., and Kumar, S. (2009). Swarm intelligence inspired classifiers in comparison with fuzzy and rough classifiers: A remote sensing approach. *Communications in Computer and Information Science*, 40:284–294.

Barker, D. M., Huang, W., Guo, Y. R., Bourgeois, A. J., and Xiao, Q. N. (2004). A three-dimensional variational data assimilation system for MM5: Implementation and initial results. *Monthly Weather Review*, 132(4):897–914.

Barrett, C. B., Barbier, E. B., and Reardon, T. (2001). Agroindustrialization, globalization, and international development: the environmental implications. *Environment and Development Economics*, 6(2001):419–433.

Basso, B., Cammarano, D., and Carfagna, E. (2013). Review of Crop Yield Forecasting Methods and Early Warning Systems. In *The First Meeting of the Scientific Advisory Committee of the Global Strategy to Improve Agricultural and Rural Statistics, July 18-19*, Rome, Italy. FAO.

Belgiu, M. and Csillik, O. (2018). Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. *Remote Sensing of Environment*, 204(January 2017):509–523.

Bhattacharyya, A. (1946). On a measure of divergence between two statistical populations defined by their probability distributions. *The Indian Journal of Statistics*, 7(4):401–406.

Birch, C. J., Rickert, K. G., and Hammer, G. L. (1998). Improved methods for predicting individual leaf area and leaf senescence in maize (Zea mays). *Australian Journal of Agricultural Research*, 49(2):249–262.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer, New York, 1st edition.

Bondy, J. A. and Murty, U. S. R. (2008). *Graph Theory*. Springer, London, 1st edition.

Boogaard, H. L., de Wit, A. J. W., te Roller, J. A., Van Diepen, C. A., Rötter, R. P., Cabrera, J. M. C. A., and Van Laar, H. H. (2014). User's guide for the WOFOST Control Centre 2.1 and the crop growth simulation model WOFOST 7.1.7.

Boote, K. J., Jones, J. W., White, J. W., Asseng, S., and Lizaso, J. I. (2013). Putting mechanisms into crop production models. *Plant, Cell and Environment*, 36(9):1658–1672.

Borgogno Mondino, E. and Corvino, G. (2019). Land tessellation effects in mapping agricultural areas by remote sensing at field level. *International Journal of Remote Sensing*, 40(18):7272–7286.

Bouman, B. A. M., Van Keulen, H., Van Laar, H. H., and Rabbinge, R. (1996). The 'School of de Wit' crop growth simulation models: A pedigree and historical overview. *Agricultural Systems*, 52(2-3):171–198.

Braun, G. J. and Fairchild, M. D. (1998). Image Lightness Rescaling Using Sigmoidal Contrast Enhancement Functions. In *Proc. SPIE 3648, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts IV, Dec 22*, pages 96–107. SPIE.

Bundesanstalt für Geowissenschaften und Rohstoffe (2018). Bodenübersichtskarte der Bundesrepublik Deutschland 1:200000.

Büttner, G., Kosztra, B., Soukup, T., Sousa, A., and Langanke, T. (2017). *CLC2018 Technical Guidelines*. European Environment Agency, Vienna, Austria.

Caselles, V., Kimmel, R., and Sapiro, G. (1997). Geodesic Active Contours. *International Journal of Computer Vision*, 22(1):61–79.

Charalampakis, A. E. and Dimou, C. K. (2015). Comparison of evolutionary algorithms for the identification of bouc-wen hysteretic systems. *Journal of Computing in Civil Engineering*, 29(3):1–11.

Chaubey, I., Cherkauer, K., Crawford, M., and Engel, B. (2011). Multiscale Sensing and Modeling Frameworks - Integrating Field to Continental Scales. *The Bridge, National Academy of Engineering*, 41(3):39–46.

Cheng, H. D., Jiang, X. H., Sun, Y., and Wang, J. (2001). Color image segmentation: advances and prospects. *Pattern Recognition*, 34:2259–2281.

Clerc, M. and Kennedy, J. (2002). The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73.

Clinton, N., Holt, A., Scarborough, J., Yan, L. I., and Gong, P. (2010). Accuracy assessment measures for object-based image segmentation goodness. *Photogrammetric Engineering and Remote Sensing*, 76(3):289–299.

Da Costa, J. P., Michelet, F., Germain, C., Lavialle, O., and Grenier, G. (2007). Delineation of vine parcels by segmentation of high resolution remote sensed images. *Precision Agriculture*, 8(1-2):95–110.

Das, S. and Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31.

De Lannoy, G. J., Reichle, R. H., Houser, P. R., Pauwels, V. R., and Verhoest, N. E. (2007). Correcting for forecast bias in soil moisture assimilation with the ensemble Kalman filter. *Water Resources Research*, 43(9).

Debats, S. R., Luo, D., Estes, L. D., Fuchs, T. J., and Caylor, K. K. (2016). A generalized computer vision approach to mapping crop fields in heterogeneous agricultural landscapes. *Remote Sensing of Environment*, 179:210–221.

Del Moral, P. (1997). Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 325(6):653–658.

del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J.-C., and Harley, R. G. (2008). Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. *IEEE Transactions on Evolutionary Computation*, 12(2):171–195.

Delécolle, R., Maas, S. J., Guérif, M., and Baret, F. (1992). Remote sensing and crop production models: present trends. *ISPRS Journal of Photogrammetry and Remote Sensing*, 47(2-3):145–161.

Deng, G. and Cahill, L. W. (1993). An Adaptive Gaussian Filter For Noise Reduction and Edge Detection. In *IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference, Oct 31 - Nov 6*, number pt 3, pages 1615–1619, San Francisco, CA, USA. IEEE.

Diestel, R. (2017). *Graph Theory*. Springer, Berlin, Heidelberg, 5th edition.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

Doorenbos, J. and Pruitt, W. O. (1977). Guidelines for predicting crop water requirements. Technical report, Food and Agriculture Organization of the United Nations (FAO), Rome, Italy.

Dorigo, W. A., Zurita-Milla, R., de Wit, A. J., Brazile, J., Singh, R., and Schaepman, M. E. (2007). A review on reflective remote sensing and data assimilation techniques for enhanced agroecosystem modeling. *International Journal of Applied Earth Observation and Geoinformation*, 9(2):165–193.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the Reduction of the Number of Points Required To Represent a Digitized Line or Its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122.

Eberhart, R. C. and Shi, Y. (2000). Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), 16-19 July 2000*, pages 84–88, La Jolla, California, USA. IEEE.

El-Sayed, M. A., Estaitia, Y. A., and Khafagy, M. A. (2013). Automated Edge Detection Using Convolutional Neural Network. *International Journal of Advanced Computer Science and Applications*, 4(10):11–17.

European Court of Auditors (2016). *The Land Parcel Identification System: A useful tool to determine the eligibility of agricultural land – but its management could be further improved*. Number 25. European Union, Luxembourg.

Evensen, G. (2003). The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367.

Finger, R., Swinton, S. M., El Benni, N., and Walter, A. (2019). Precision Farming at the Nexus of Agricultural Production and the Environment. *Annual Review of Resource Economics*, 11(1):313–335.

Food and Agriculture Organization of the United Nations (2015). *Climate Change and Food Systems - Global assessments and implications for food security and trade*. Rome.

Food and Agriculture Organization of the United Nations (2018). *Transforming Food and Agriculture To achieve the SDGs*. FAO, Rome, Italy, Rome.

Foster, T., Brozović, N., Butler, A. P., Neale, C. M., Raes, D., Steduto, P., Fereres, E., and Hsiao, T. C. (2017). AquaCrop-OS: An open source version of FAO's crop water productivity model. *Agricultural Water Management*, 181:18–22.

Fournier, C. and Andrieu, B. (1998). A 3D architectural and process-based model of maize development. *Annals of Botany*, 81(2):233–250.

García-Pedrero, A., Gonzalo-Martín, C., and Lillo-Saavedra, M. (2017). A machine learning approach for agricultural parcel delineation through agglomerative segmentation. *International Journal of Remote Sensing*, 38(7):1809–1819.

Ge, F., Wang, S., and Liu, T. (2007). New benchmark for image segmentation evaluation. *Journal of Electronic Imaging*, 16(3):033011.

German Weather Service (DWD) (2019). DWD Climate Data Center.

Gonzalo-Martín, C., Lillo-Saavedra, M., Menasalvas, E., Fonseca-Luengo, D., García-Pedrero, A., and Costumero, R. (2016). Local optimal scale in a hierarchical segmentation method for satellite images: An OBIA approach for the agricultural landscape. *Journal of Intelligent Information Systems*, 46(3):517–529.

Graesser, J. and Ramankutty, N. (2017). Detection of cropland field parcels from Landsat imagery. *Remote Sensing of Environment*, 201(July):165–180.

Grassini, P., van Bussel, L. G., Van Wart, J., Wolf, J., Claessens, L., Yang, H., Boogaard, H. L., de Groot, H., van Ittersum, M. K., and Cassman, K. G. (2015). How good is good enough? Data requirements for reliable crop yield simulations and yield-gap analysis. *Field Crops Research*, 177:49–63.

Gregson, P. H. (1993). Using Angular Dispersion of Gradient Direction for Detecting Edge Ribbons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):682–696.

Grewal, M. S. and Andrews, A. P. (2015). *Kalman Filtering - Theory and Practice Using MATLAB*. John Wiley & Sons, Somerset, USA, ed., 4th edition.

Guo, C., Zhang, L., Zhou, X., Zhu, Y., Cao, W., Qiu, X., Cheng, T., and Tian, Y. (2017). Integrating remote sensing information with crop model to monitor wheat growth and yield based on simulation zone partitioning. *Precision Agriculture*, 19(1):1–24.

Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621.

Hart, K., Allen, B., M, L., C, K., P, B., J, E., and A, B. (2013). *Land As an Environmental Resource , Report Prepared for DG Environment, Contract No ENV.B.1/ETU/2011/0029*. Number y. Institute for European Environmental Policy, London, London, United Kingdom.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Science+Business Media, New York, NY, USA, 2nd edition.

Hellinger, E. (1909). Neue Begründung der Theorie quadratischer Formen von unendlich vielen Veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271.

Hoffmann, H., Zhao, G., van Bussel, L. G. J., Enders, A., Specka, X., Sosa, C., Yeluripati, J., Tao, F., Constantin, J., Raynal, H., Teixeira, E., Grosz, B., Doro, L., Zhao, Z., Wang, E., Nendel, C., Kersebaum, K. C., Haas, E., Kiese, R., Klatt, S., Eckersten, H., Vanuytrecht, E., Kuhnert, M., Lewan, E., Rötter, R., Roggero, P. P., Wallach, D., Cammarano, D., Asseng, S., Krauss, G., Siebert, S., Gaiser, T., and Ewert, F. (2015). Variability of effects of spatial climate data aggregation on regional yield simulation by crop models. *Climate Research*, 65:53–69.

Hoogenboom, G., Porter, C. H., Shelia, V., Boote, K. J., Singh, U., White, J. W., Hunt, L. A., Ogoshi, R., Lizaso, J. I., Koo, J., Asseng, S., Singels, A., Moreno, L. P., and Jones, J. W. (2019). Decision Support System for Agrotechnology Transfer (DSSAT) Version 4.7.5; DSSAT Foundation, Gainesville, Florida, USA.

Hsiao, T. C., Heng, L., Steduto, P., Rojas-Lara, B., Raes, D., and Fereres, E. (2009). AquaCrop - The FAO crop model to simulate yield response to water: III. Parameterization and testing for maize. *Agronomy Journal*, 101(3):448–459.

Inan, H. I., Sagris, V., Devos, W., Milenov, P., van Oosterom, P., and Zevenbergen, J. (2010). Data model for the collaboration between land administration systems and agricultural land parcel identification systems. *Journal of Environmental Management*, 91(12):2440–2454.

Ines, A. V. M., Das, N. N., Hansen, J. W., and Njoku, E. G. (2013). Assimilation of remotely sensed soil moisture and vegetation with a crop simulation model for maize yield prediction. *Remote Sensing of Environment*, 138:149–164.

Jégo, G., Pattey, E., and Liu, J. (2012). Using Leaf Area Index, retrieved from optical imagery, in the STICS crop model for predicting yield and biomass of field crops. *Field Crops Research*, 131:63–74.

Jibo, Y., Haikuan, F., and Xiudong, Q. (2016). Monitor key parameters of winter wheat using Crop model. *IOP Conference Series: Earth and Environmental Science*, 46(1).

Jin, M., Liu, X., Wu, L., and Liu, M. (2015). An improved assimilation method with stress factors incorporated in the WOFOST model for the efficient assessment of heavy metal stress levels in rice. *International Journal of Applied Earth Observation and Geoinformation*, 41:118–129.

Jin, X., Kumar, L., Li, Z., Feng, H., Xu, X., Yang, G., and Wang, J. (2018). A review of data assimilation of remote sensing and crop models. *European Journal of Agronomy*, 92(November 2017):141–152.

Jin, X., Kumar, L., Li, Z., Xu, X., Yang, G., and Wang, J. (2016). Estimation of winter wheat biomass and yield by combining the aquacrop model and field hyperspectral data. *Remote Sensing*, 8(12).

Jones, J. W., Antle, J. M., Basso, B., Boote, K. J., Conant, R. T., Foster, I., Godfray, H. C. J., Herrero, M., Howitt, R. E., Janssen, S., Keating, B. A., Munoz-Carpena, R., Porter, C. H., Rosenzweig, C., and Wheeler, T. R. (2017). Toward a new generation of agricultural system data, models, and knowledge products: State of agricultural systems science. *Agricultural Systems*, 155:269–288.

Jones, J. W., Hoogenboom, G., Porter, C. H., Boote, K. J., Batchelor, W. D., Hunt, L. A., Wilkens, P. W., Singh, U., Gijsman, A. J., and Ritchie, J. T. (2003). The DSSAT cropping system model. *European Journal of Agronomy*, 18(3-4):235–265.

Jozdani, S. and Chen, D. (2020). On the versatility of popular and recently proposed supervised evaluation metrics for segmentation quality of remotely sensed images: An experimental case study of building extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 160(January):275–290.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45.

Kalman, R. E. and Bucy, R. S. (1961). New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*, 83:95–108.

Kamilaris, A. and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147(February):70–90.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Keating, B. A., Carberry, P. S., Hammer, G. L., Probert, M. E., Robertson, M. J., Holzworth, D., Huth, N. I., Hargreaves, J. N., Meinke, H., Hochman, Z., McLean, G., Verburg, K., Snow, V., Dimes, J. P., Silburn, M., Wang, E., Brown, S., Bristow, K. L., Asseng, S., Chapman, S., McCown, R. L., Freebairn, D. M., and Smith, C. J. (2003). An overview of APSIM, a model designed for farming systems simulation. *European Journal of Agronomy*, 18(3-4):267–288.

Kemker, R., Salvaggio, C., and Kanan, C. (2018). Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145(June 2017):60–77.

Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks, 27 Nov.-1 Dec. 1995*, pages 1942–1948, Perth, WA, Australia. IEEE.

Kennedy, J., Eberhart, R. C., and Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann, Burlington, MA, 1st edition.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.

Kottek, M., Grieser, J., Beck, C., Rudolf, B., and Rubel, F. (2006). World Map of the Köppen-Geiger climate classification updated. *Meteorologische Zeitschrift*, 15(3):259–263.

Koubarakis, M., Bereta, K., Bilidas, D., Giannousis, K., Ioannidis, T., Pantazi, D. A., Stamoulis, G., Haridi, S., Vlassov, V., Bruzzone, L., Paris, C., Eltoft, T., Krämer, T., Charalabidis, A., Karkaletsis, V., Konstantopoulos, S., Dowling, J., Kakantousis, T., Datcu, M., Dumitru, C. O., Appel, F., Bach, H., Migdall, S., Hughes, N., Arthurs, D., and Fleming, A. (2019). From Copernicus Big Data to Extreme Earth Analytics. *Advances in Database Technology - EDBT*, 2019-March:690–693.

Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Kussul, N., Lemoine, G., Gallego, F. J., Skakun, S. V., Lavreniuk, M., and Shelestov, A. Y. (2016). Parcel-Based Crop Classification in Ukraine Using Landsat-8 Data and Sentinel-1A Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(6):2500–2508.

Längkvist, M., Kiselev, A., Alirezaie, M., and Loutfi, A. (2016). Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sensing*, 8(4).

Li, Y., Zhou, Q., Zhou, J., Zhang, G., Chen, C., and Wang, J. (2014). Assimilating remote sensing information into a coupled hydrology-crop growth model to estimate regional maize yield in arid regions. *Ecological Modelling*, 291:15–27.

Li, Z., Wang, J., Xu, X., Zhao, C., Jin, X., Yang, G., and Feng, H. (2015). Assimilation of Two Variables Derived from Hyperspectral Data into the DSSAT-CERES Model for Grain Yield and Quality Estimation. *Remote Sensing*, 7(9):12400–12418.

Liu, J., Pattey, E., Miller, J. R., McNairn, H., Smith, A., and Hu, B. (2010). Estimating crop stresses, aboveground dry biomass and yield of corn using multi-temporal optical

data combined with a radiation use efficiency model. *Remote Sensing of Environment*, 114(6):1167–1177.

Liu, X. P., Li, X., Peng, X. J., Li, H. B., and He, J. Q. (2008). Swarm intelligence for classification of remote sensing data. *Science in China, Series D: Earth Sciences*, 51(1):79–87.

Liu, Y., Ling, X., Shi, Z., Lv, M., Fang, J., and Zhang, L. (2011). A survey on particle swarm optimization algorithms for multimodal function optimization. *Journal of Software*, 6(12):2449–2455.

Ma, L., Li, M., Ma, X., Cheng, L., Du, P., and Liu, Y. (2017). A review of supervised object-based land-cover image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130:277–293.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., and Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152(November 2018):166–177.

Maas, S. J. (1988). Use of remotely-sensed information in agricultural crop growth models. *Ecological Modelling*, 41(3-4):247–268.

MacDonald, R. B. and Hall, F. G. (1980). Global crop forecasting. *Science*, 208(4445):670–679.

Martin, D. R., Fowlkes, C. C., and Malik, J. (2004). Learning to detect natural image boundaries using brightness and texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549.

Masoud, K. M., Persello, C., and Tolpekin, V. A. (2019). Delineation of Agricultural Field Boundaries from Sentinel-2 Images Using a Novel Super-Resolution Contour Detector Based on Fully Convolutional Networks. *Remote Sensing*, 12(1):59.

McGuinness, K. and O'Connor, N. E. (2010). A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444.

Meijering, E., Jacob, M., Sarria, J.-C., Steiner, P., Hirling, H., and Unser, M. (2004). Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry*, 58A(2):167–176.

Mendes, R., Kennedy, J., and Neves, J. (2003). Watch thy neighbor or how the swarm can learn from its environment. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706)*, pages 88–94, Indianapolis, IN, USA. IEEE.

Ministerium für Energiewende, Landwirtschaft, Umwelt, N. u. D. S.-H. (2020). Nutzung des landwirtschaftlichen Bodens - Schleswig-Holstein.

Monteith, J. L. (1977). Climate and the Efficiency of Crop Production in Britain. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 281(980):277–294.

Mulla, D. J. (2013). Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems Engineering*, 114(4):358–371.

Nair, V. and Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel*, Haifa, Israel.

Niedersachsen, L. f. S. (2018a). Bodennutzung und Ernte 2016. *Statistische Berichte Niedersachsen*, 2016:21.

Niedersachsen, L. f. S. (2018b). Bodennutzung und Ernte 2017. *Statistische Berichte Niedersachsen*, 2017:21.

NIST, N. I. o. S. &. T. (2020). NIST/SEMATECH e-Handbook of Statistical Methods.

Omkar, S. N., Senthilnath, J., Mudigere, D., and Manoj Kumar, M. (2008). Crop Classification using Biologically-inspired Techniques with High Resolution Satellite Image. *Journal of the Indian Society of Remote Sensing*, 36(2):175–182.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 9, pages 62–66.

Papari, G. and Petkov, N. (2011). Edge and line oriented contour detection: State of the art. *Image and Vision Computing*, 29(2-3):79–103.

Peña, J. M., Gutiérrez, P. A., Hervás-Martínez, C., Six, J., Plant, R. E., and López-Granados, F. (2014). Object-based image classification of summer crops with machine learning methods. *Remote Sensing*, 6(6):5019–5041.

Penning de Vries, F. W. T. and Rabbinge, R. (1995). Models in research and education, planning and practice. In Haverkort, A. J. and MacKerron, D. K. L., editors, *Potato Ecology and Modelling of Crops under Conditions Limiting Growth*, pages 1–18. Springer, Dordrecht.

Peram, T., Veeramachaneni, K., and Mohan, C. K. (2003). Fitness-distance-ratio based particle swarm optimization. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706), April 26*, pages 174–181, Indianapolis, IN, USA. IEEE.

Perona, P. and Malik, J. (1990). Scale Space and Edge Detection Using Anisotropic Diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Persello, C., Tolpekin, V. A., Bergado, J. R., and de By, R. A. (2019). Delineation of agricultural fields in smallholder farms from satellite images using fully convolutional networks and combinatorial grouping. *Remote Sensing of Environment*, 231(June).

Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization, An overview. *Swarm Intelligence*, 1:33–57.

Raes, D., Steduto, P., Hsiao, T. C., and Fereres, E. (2009). AquaCrop - The FAO crop model to simulate yield response to water: II. main algorithms and software description. *Agronomy Journal*, 101(3):438–447.

Raes, D., Steduto, P., Hsiao, T. C., and Fereres, E. (2018). *AquaCrop Reference Manual version 6.0-6.1 - Chapter 3: Calculation Procedures*. Rome, Italy.

Rahman, M. S., Di, L., Yu, Z., Yu, E. G., Tang, J., Lin, L., Zhang, C., and Gaigalas, J. (2019). Crop Field Boundary Delineation using Historical Crop Rotation Pattern. *2019 8th International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, (July):1–5.

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256.

Reichle, R. H. (2008). Data assimilation methods in the Earth sciences. *Advances in Water Resources*, 31(11):1411–1418.

Rembold, F., Atzberger, C., Savin, I., and Rojas, O. (2013). Using low resolution satellite imagery for yield prediction and yield anomaly detection. *Remote Sensing*, 5(4):1704–1733.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH '87 Conference Proceedings, July 1987*, volume 21, pages 25–34, Anaheim, California.

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407.

Rodriguez, J. C., Duchemin, B., Hadria, R., Watts, C., Garatuza, J., Chehbouni, A., Khabba, S., Boulet, G., Palacios, E., and Lahrouni, A. (2004). Wheat yield estimation using remote sensing and the STICS model in the semiarid Yaqui valley, Mexico. *Agronomie*, 24:295–304.

Roser, M. and Ortiz-Ospina, E. (2013). Gloabl Extreme Poverty. *Our World in Data*.

Rydberg, A. and Borgefors, G. (2001). Integrated method for boundary delineation of agricultural fields in multispectral satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 39(11):2514–2520.

Sagris, V. and Devos, W. (2008). *LPIS Core Conceptual Model: Methodology for Feature Catalogue and Application Schema*. OPOCE, Luxembourg.

Sands, R. D., Jones, C. A., and Marshall, E. (2014). Global drivers of agricultural demand and supply. Technical Report 174, United States Department of Agriculture, Economic Research Service, Washington, D.C.

Sasaki, Y. (1970). Some Basic Formalisms in Numerical Variational Analysis. *Monthly Weather Review*, 98(12):875–883.

Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., and Kikinis, R. (1998). Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2(2):143–168.

Shen, L., Huang, X., and Fan, C. (2018). Double-group particle swarm optimization and its application in remote sensing image segmentation. *Sensors (Switzerland)*, 18(5):1393–1422.

Silvestro, P. C., Pignatti, S., Pascucci, S., Yang, H., Li, Z., Yang, G., Huang, W., and Casa, R. (2017). Estimating Wheat Yield in China at the Field and District Scale from the Assimilation of Satellite Data into the Aquacrop and Simple Algorithm for Yield (SAFY) Models. *Remote Sensing*, 9(5):1–24.

Son, N. T., Chen, C. F., Chen, C. R., Chang, L. Y., and Chiang, S. H. (2016). Rice yield estimation through assimilating satellite data into a crop simulation model. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 41(July):993–996.

Statistisches Amt für Hamburg und Schleswig-Holstein (2019). *Die Bodennutzung in Schleswig-Holstein*. Statistisches Amt für Hamburg und Schleswig-Holstein, Hamburg, Germany.

Statistisches Landesamt Sachsen-Anhalt (2019). Tabellen Land- und Forstwirtschaft, Fischerei.

Steduto, P., Hsiao, T. C., Fereres, E., and Raes, D. (2012). Crop yield response to water. *FAO Irrigation and Drainage Paper*, 66:500.

Steduto, P., Hsiao, T. C., Raes, D., and Fereres, E. (2009). AquaCrop - The FAO crop model to simulate yield response to water: I. concepts and underlying principles. *Agronomy Journal*, 101(3):426–437.

Tilman, D., Fargione, J., Wolff, B., D'Antonio, C., Dobson, A., Howarth, R., Schindler, D., Schlesinger, W. H., Simberloff, D., and Swackhamer, D. (2001). Forecasting agriculturally driven global environmental change. *Science*, 292(5515):281–284.

Ting, K., Adelzaher, T., Alleyne, A., and Rodriguez, L. (2011). Information Technology and Agriculture - Global Challenges and Opportunities. *The Bridge, National Academy of Engineering*, 41(3):6–13.

Tiwari, P. S., Pande, H., Kumar, M., and Dadhwal, V. K. (2009). Potential of IRS P-6 LISS IV for agriculture field boundary delineation. *Journal of Applied Remote Sensing*, 3(1):033528.

Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Proceedings of the IEEE International Conference on Computer Vision, 7 Jan*, pages 839–846, Bombay, India.

Trémolet, Y. (2006). Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2483–2504.

Tsagkatakis, G., Aidini, A., Fotiadou, K., Giannopoulos, M., Pentari, A., and Tsakalides, P. (2019). Survey of deep-learning approaches for remote sensing observation enhancement. *Sensors*, 19(18):1–39.

Turker, M. and Kok, E. H. (2013). Field-based sub-boundary extraction from remote sensing imagery using perceptual grouping. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79:106–121.

United Nations, Department of Economic and Social Affairs, and Population Division (2015). *World Population Prospects - The 2017 Revision, Key Findings and Advance Tables*. United Nations, Department of Economic and Social Affairs, Population Division, New York.

van Bussel, L. G. J., Ewert, F., Zhao, G., Hoffmann, H., Enders, A., Wallach, D., Asseng, S., Baigorria, G. A., Basso, B., Biernath, C., Cammarano, D., Chryssanthacopoulos, J., Constantin, J., Elliott, J., Glotter, M., Heinlein, F., Kersebaum, K. C., Klein, C., Nendel, C., Priesack, E., Raynal, H., Romero, C. C., Rötter, R. P., Specka, X., and Tao, F. (2016). Spatial sampling of weather data for regional crop yield simulations. *Agricultural and Forest Meteorology*, 220:101–115.

van Diepen, C., Wolf, J., van Keulen, H., and Rappoldt, C. (1989). WOFOST: a simulation model of crop production. *Soil Use and Management*, 5(1):16–24.

van Diepen, C. A., Driessen, P. M., Goudriaan, J., Hijmans, R. J., Hooijer, A. A., van Keulen, H., de Koning, G. H. J., van Kraalingen, D. W. G., Kropff, M. J., van Laar, H. H., Rappoldt, C., Penning de Vries, F. W. T., Spitters, C. J. T., Supit, I., van der Wal, T., and Wolf, J. (1994). System Description of the WOFOST 6.0 Crop Simulation Model Implemented in CGMS, Volume 1: Theory and Algorithms.

Van Ittersum, M., Leffelaar, P., Van Keulen, H., Kropff, M., Bastiaans, L., and Goudriaan, J. (2003a). Developments in modelling crop growth, cropping systems and production systems in the Wageningen school. *NJAS - Wageningen Journal of Life Sciences*, 50(2):239–247.

Van Ittersum, M. K., Leffelaar, P. A., Van Keulen, H., Kropff, M. J., Bastiaans, L., and Goudriaan, J. (2003b). On approaches and applications of the Wageningen crop models. *European Journal of Agronomy*, 18(3-4):201–234.

van Keulen, H. (1975). *Simulation of water use and herbage growth in arid regions*. Phd thesis, Wageningen University, Wageningen, Netherlands.

Velasco, F. A. and Marroquin, J. L. (2003). Growing snakes: Active contours for complex topologies. *Pattern Recognition*, 36(2):475–482.

Verrelst, J., Camps-Valls, G., Muñoz-Marí, J., Rivera, J. P., Veroustraete, F., Clevers, J. G., and Moreno, J. (2015). Optical remote sensing and the retrieval of terrestrial vegetation bio-geophysical properties - A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 108:273–290.

Verrelst, J., Rivera, J. P., Leonenko, G., Alonso, L., and Moreno, J. (2014). Optimizing LUT-based RTM inversion for semiautomatic mapping of crop biophysical parameters from sentinel-2 and -3 data: Role of cost functions. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):257–269.

Wagner, M. P. and Oppelt, N. (2020). Extracting Agricultural Fields from Remote Sensing Imagery Using Graph-Based Growing Contours. *Remote Sensing*, 12(7):1205.

Wagner, M. P., Taravat, A., and Oppelt, N. (2019). Particle swarm optimization for assimilation of remote sensing data in dynamic crop models. In Neale, C. M. U. and Maltese, A., editors, *Proc. SPIE 11149, Remote Sensing for Agriculture, Ecosystems, and Hydrology XXI, SPIE Remote Sensing 2019, Sep 9-12*, Strasbourg, France. SPIE.

Waldner, F. and Diakogiannis, F. I. (2020). Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network. *Remote Sensing of Environment*, 245(May):111741.

Warszawski, L., Frieler, K., Huber, V., Piontek, F., Serdeczny, O., and Schewe, J. (2014). The Inter-Sectoral Impact Model Intercomparison Project (ISI-MIP): project framework. *Proceedings of the National Academy of Sciences of the United States of America*, 111(9):3228–32.

Watkins, B. and van Niekerk, A. (2019). A comparison of object-based image analysis approaches for field boundary delineation using multi-temporal Sentinel-2 imagery. *Computers and Electronics in Agriculture*, 158(February):294–302.

Weigand, M., Staab, J., Wurm, M., and Taubenböck, H. (2020). Spatial and semantic effects of LUCAS samples on fully automated land use/land cover classification in high-resolution Sentinel-2 data. *Int J Appl Earth Obs Geoinformation*, 88(June):102065.

Weiss, M. and Baret, F. (2016). S2ToolBox Level 2 products: LAI, FAPAR, FCOVER - Version 1.1. Technical report, European Space Agency.

Weiss, M., Jacob, F., and Duveiller, G. (2020). Remote sensing for agricultural applications: A meta-review. *Remote Sensing of Environment*, 236(December 2018):111402.

Wulder, M. A., Masek, J. G., Cohen, W. B., Loveland, T. R., and Woodcock, C. E. (2012). Opening the archive: How free data has enabled the science and monitoring promise of Landsat. *Remote Sensing of Environment*, 122:2–10.

Xie, Y., Wang, P., Sun, H., Zhang, S., and Li, L. (2017). Assimilation of Leaf Area Index and Surface Soil Moisture With the CERES-Wheat Model for Winter Wheat Yield Estimation Using a Particle Filter Algorithm. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(4):1303–1316.

Xu, C. and Prince, J. L. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369.

Yan, L. and Roy, D. P. (2014). Automated crop field extraction from multi-temporal Web Enabled Landsat Data. *Remote Sensing of Environment*, 144:42–64.

Yan, L. and Roy, D. P. (2016). Conterminous United States crop field size quantification from multi-temporal Landsat data. *Remote Sensing of Environment*, 172:67–86.

Yang, X.-S. (2014). *Nature-Inspired Optimization Algorithms*. Elsevier, London, United Kingdom, 1st edition.

Yiu, P. (2000). The uses of homogeneous barycentric coordinates in plane Euclidean geometry. *International Journal of Mathematical Education in Science and Technology*, 31(4):569–578.

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., and Fraundorfer, F. (2017). Deep learning in remote sensing: a review. *IEEE Geoscience and Remote Sensing Magazine*, (December):8–36.

Declaration of Authorship

Hiermit erkläre ich, dass die vorliegende Dissertation, abgesehen von der Beratung durch die Betreuer, nach Inhalt und Form selbstständig verfasst wurde und keine weiteren Quellen und Hilfsmittel als die hier angegebenen verwendet wurden. Diese Arbeit wurde weder ganz noch in Teilen an anderer Stelle einer Prüfungskommission vorgelegt. Als kumulative Dissertation sind Kapitel 2 bis 4, wie zu Beginn der jeweiligen Kapitel vermerkt, in den genannten Zeitschriften veröffentlicht. Ich erkläre, dass die vorliegende Arbeit unter Einhaltung der Regeln guter wissenschaftlicher Praxis der Deutschen Forschungsgemeinschaft entstanden ist. Ich versichere, dass mir kein akademischer Grad entzogen wurde.

Kiel, den 25.06.2020