

2023-09-20

Efficient Calculation of Distance Transform on Discrete Global Grid Systems and Its Application in Automatic Soil Sampling Site Selection

Kazemi, Meysam

Kazemi, M. (2023). Efficient calculation of distance transform on Discrete Global Grid Systems and its application in automatic soil sampling site selection (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from <https://prism.ucalgary.ca>.

<https://hdl.handle.net/1880/117163>

Downloaded from PRISM Repository, University of Calgary

UNIVERSITY OF CALGARY

Efficient Calculation of Distance Transform on Discrete Global Grid Systems and Its
Application in Automatic Soil Sampling Site Selection

by

Meysam Kazemi

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA

SEPTEMBER, 2023

© Meysam Kazemi 2023

Abstract

Geospatial data analysis often requires the computing of a distance transform (DT) for a given vector feature. For instance, in wildfire management, it is helpful to find the distance of all points in an area from the wildfire's boundary. Computing a distance transform on traditional Geographic Information Systems (GIS) is usually adopted from image processing methods, albeit prone to distortion resulting from flat maps. Discrete Global Grid Systems (DGGS) are relatively new low-distortion globe-based GIS that discretize the Earth into highly regular cells using multiresolution grids. In this thesis, we introduce an efficient DT algorithm for DGGS. Our novel algorithm heavily exploits the hierarchy of a DGGS and its mathematical properties and applies to many different DGGSs. We evaluate our method by comparing its distortion with the DT methods used in traditional GIS and its speed with the application of general 3D mesh DT algorithms on the DGGS grid. We demonstrate that our method is efficient and has lower distortion. To evaluate our DT algorithm further, we have used a real-world case study of selecting soil test points within agricultural fields. Multiple criteria including the distance of soil test points to different features should be considered to select representative points in a field. We show that DT can help to automate the process of selecting test points, by allowing us to efficiently calculate objectives for a representative test point. DT also allows for efficient calculation of buffers from certain features such as farm headlands and underground pipelines, to avoid certain regions when selecting the test points.

Acknowledgements

I would like to express my heartfelt appreciation to the individuals and organizations who have been instrumental in the completion of my master's degree. Your support, guidance, and encouragement have played a pivotal role in shaping this research.

First and foremost, I want to express my deepest gratitude to my amazing supervisor, Dr. Faramarz Samavati. Your mentorship has been the cornerstone of this journey. You taught me invaluable lessons on how to see the big picture, how to give good presentations, and encouraged me to think like a scientist. I am grateful for all the great conversations we had in our meetings and all the valuable feedback you gave me on my writing and presentations. I also extend my sincere thanks to my co-supervisor, Dr. Emmanuel Stefanakis, for your valuable contributions and insights, which have enriched this thesis. I am deeply grateful to Dr. Adam Runions and Dr. Farhad Maleki for their time in reading and examining my thesis, as well as for the interesting discussions during my defence.

I would like to thank the organizations that generously provided financial support for this research: the University of Calgary, NSERC, Mitacs, Telus Agriculture, and Vivid Theory. Special thanks to Vincent, Garrett, and Jessica for great discussions and insights during my time at the Telus Agriculture project, and Brent, Brett, and Jordan for their support during my time at Vivid Theory.

I had the privilege to be a part of the GIV lab, and the 6th floor of the Math Science building

in general, which is a special place for me. Huge thanks to my past and current labmates Troy Alderson, Majid Amirfakhrian, Kathleen Ang, Arya Banaeizadeh, Erik Biederstadt, Hesam Damghanian, Timothy Davison, Katayoon Etemad, John Hall, Mohammad Hameed, Roghayeh Heidari, Riley James, Armin Kazemi, Reza Khanmohammadi, Hooman Khosravi, Josiah Lansang, Jeffrey Layton, Mia MacTavish, Amir Mehr, Amirhossein Mirtabatabaeipour, Amir Mirzai Golpayegani, Christopher Mossman, Tung Nguyen, Mohammadreza Osouli, Samin Sabokrohiyeh, Xi Wang, Benjamin Ulmer and Lakin Wecker for their friendship, many great conversations, and leaving a lasting impact.

In addition, certain individuals deserve extra recognition. Amirhossein, you have been my friend long before I started my Master's. Thank you for all the late-night conversations that we had and for all your support during the hard times. Hessam and Mahsa, thank you for all your support, especially during the hard COVID times and all the great conversations and stories you told us. Mia, Ben, and Jeremy, thank you for being incredible friends and for all your emotional support. Lakin, Ben, Jeremy, and John, thank you for the countless ways you supported me, for all the insightful discussions, and for helping me in every step of my journey from coding to writing to presentations. I don't know if I would have made it through without your friendship and support.

A special word of thanks goes out to my loving partner, Rayehe, for your unwavering support, understanding, and encouragement throughout this academic endeavour.

In addition, I want to acknowledge the meaningful friendships that have sustained me during this time. Seti, Parmis, Reza, Bahareh, Omar, and Lucky, your friendship and company have been a source of joy and strength.

Finally, I extend my deepest gratitude to my parents for bringing me into this world and raising me, as well as to my younger brother who taught me that life is not boring. Without your unwavering support and love, I wouldn't be where I am today. Thank you so much.

To all those mentioned and to anyone else who has been part of my academic and personal journey, I am profoundly appreciative of your contributions, and I am honoured to have had your support. Thank you all.

Table of Contents

Abstract	ii
Acknowledgements	iii
Table of Contents	vi
List of Tables	ix
List of Figures	xi
1 Introduction	1
1.1 Problem Statement	2
1.2 Goals and Scope	3
1.3 Methodology	4
1.4 Contribution	6
1.5 Thesis Overview	7
2 Background and Related Work	9
2.1 Distance Transform Definition and Applications	10
2.2 Discrete Global Grid Systems	11
2.3 Distance on Globe	14
2.4 Computing Distance Transform	16
2.5 Soil Sampling Techniques in Agriculture	18

2.6	Delineation of Management Zones	21
2.7	Benchmark Soil Sampling	22
2.8	Solving a multi-objective optimization problem	23
3	Calculating Distance Transform on DGGS	25
3.1	Reducing the Search Space for Developing an Efficient Algorithm	25
3.2	An Algorithm to Calculate Distance Transform on DGGS	30
4	Distance Transform Results and Discussion	35
4.1	Correctness Analysis	37
4.2	Performance Analysis	39
4.3	Discussion	43
4.4	Comparison	44
5	Application of Distance transform in Soil Sampling	49
5.1	Benchmark Site Selection Framework	52
5.2	From Data to Management Zone	54
5.3	Optimization Model	56
6	Use Case Results and Evaluation	69
6.1	Modularity and Extendability	78
6.2	Discussion	80
7	Implementation	84
7.1	DT Operation as a Function	84
7.2	DT Visualizer and Tester Program	86
7.3	Case Study	88
8	Conclusion and Future Directions	91
	Bibliography	93

Appendix 104

List of Tables

4.1	Difference in millimeters between the ArcGIS Pro calculated geodesic distances and ours.	39
4.2	Difference in meters between the ArcGIS Pro calculated planar distances and the distances from Algorithm 5.	46
4.3	Execution time of our algorithm and MMP algorithm along with the number of target cells for the border of Ontario.	48
6.1	The list of fields used for evaluation. Total execution time captures the entire process including loading data from the local cache (tested on a computer with an Intel Core i7-6700 CPU).	70
6.2	Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 1.	73
6.3	Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 2.	74
6.4	Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 3.	75
6.5	Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 4.	76

6.6	Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 5.	77
6.7	The mean percent improvement of the selected benchmark sites.	78
6.8	The effect of weights in objectives. This table is based on MZ 2 on field 1.	81
6.9	The mean percent improvement for field 1 when the number of MZs varies between 6 to 30 MZs.	82

List of Figures

1.1	DT applied to different types of grids: (a) a regular image-based grid, (b) a DGGS grid, and (c) an irregular general mesh. In this visualization, darker cells are closer to the feature and brighter cells are farther to the feature.	3
1.2	(a) Imaginary management zones for a field and (b) , the result of applying DT to the management zone boundaries. Highlighted areas in (b) are candidates for test point samples.	6
2.1	(a) The binary image before applying DT. (b) The result of applying DT to the binary image.	10
2.2	(a) Tetrahedron (b) Octahedron (c) Icosahedron (d) Disdyakis dodecahedron (e) Disdyakis triacontahedron	12
2.3	Examples of congruent (a) 1 to 4 quad, (b) 1 to 3 triangle, and (c) non-congruent hexagonal refinement. The black cells are original cells and the green lines show one level of refinement.	13
2.4	Disdyakis Triacontahedron DGGS at resolution (a) 1, (b) 2, and (c) 3.	13
2.5	Disdyakis Triacontahedron DGGS at resolution three in the polyhedral domain (left) and the spherical domain (right).	14
2.6	The red line is the great-circle arc connecting points p and q	15

2.7	The red line is the great circle arc connecting points v and w , and point p is the point in question.	16
2.8	Traditional methods of soil sampling (a) composite sampling and, (b) grid sampling. Blue flags show the sample points used to make the composite sample (only one test), while white flags are sampled and tested individually. (Figures are taken from [41])	19
2.9	Soil sampling management zones practice. (a) homogeneous sub-units of a farm field, (b) composite sampling from each MZ. The soil collected from similar-shaped flags mixes to form a composite sample, which means only three soil tests will be done. (Figures are adapted from [41])	20
2.10	The process of MZ delineation from a performance function across the field. . . .	20
2.11	Benchmark sampling using management zone technique. (Figure is adapted from [41])	21
3.1	Distance of a cell to a feature.	26
3.2	An example of a bounding space (green circle).	27
3.3	Two types of triangle 1 to 4 refinements with appropriate d.	29
3.4	Caption for figure	30
4.1	Distance transform for the border of Ontario at target resolution (a) 7 (Avg. cell size: 1037.3 km^2), (b) 9 (Avg. cell size: 64.8 km^2), (c) 12 (Avg. cell size: 1.0 km^2).	36
4.2	Distance transform for the border of (a) Mainland British Columbia and (b) Nunavut at target resolution 11 (Avg. cell size: 4.0 km^2), and (c) Prince's Edward Island at target resolution 15 (Avg. cell size: 0.016 km^2).	36

4.3	Distance transform for the border of (a) the city of Calgary at target resolution 15 (Avg. cell size: 0.016 km^2) and (b) a farm field in Alberta at target resolution 19 (Avg. cell size: 61.8 m^2).	37
4.4	Distance transform sample points and corresponding boundaries imported into ArcGIS Pro for (a) the province of Ontario, (b) the city of Calgary, and (c) a farm field in Alberta.	38
4.5	The number of distance operations in different target resolutions for the boundary of Ontario.	41
4.6	The number of distance operations in different target resolutions for the boundary of Ontario.	42
4.7	The effect of the base resolution on the performance of the iterative algorithm for the target resolution 12.	43
4.8	Distance transform from Yellowknife using (a) ArcGIS and planar distance calculations, (b, c) DGGs and our method.	45
4.9	Execution time of our algorithm and the MMP algorithm at different resolutions for the border of Ontario.	47
5.1	Criteria for MZ representative area. (a) absolute difference to the median value of the underlying function for MZ 2 (the numbers are in the unit of the performance function), and (b) distance to the boundaries of MZ 2 have been shown using a cool-warm colour map (in meters). (c) blue diamonds are benchmark sites concentrated in the lower right part of the field, while green circles are benchmark sites that are distanced from each other.	51
5.2	The flowchart of selecting benchmark sites for a new field.	53
5.3	An example of selected satellite images for delineation process.	55

5.4	An example of an F-index, F-Map, the histogram of F-Index, and the associated thresholds.	56
5.5	Visualization of absolute difference to median F-index for MZ 3.	59
5.6	Visualization of distance to boundaries of MZ 3. The farthest cell from the boundary is 87 m away from the boundaries.	60
5.7	Visualization of distance to previously sampled point for MZ 3. The farthest cell from this previously sampled point is 1533 m away from it.	61
5.8	Visualization of the steepness for MZ 3. The legend shows the steepness in degrees. The steepest point of this MZ is 7.8 degrees steep.	62
5.9	The extracted headland of the farm field ($D_H = 50m$).	63
5.10	The forced distancing between selected benchmark sites. t_1 and t_2 are already selected benchmark sites. When deciding for point c_i in the next MZ, we only check the distance to t_1 and t_2	64
5.11	Visualization of the error function for MZ 3. All weights are set to 1 ($w_j = 1, j = 1, 2, 3, 4$) in this figure. E_T at the optimal point (denoted by the star) is 0.0279. . . .	67
5.12	Visualization of the final error function for the entire field. All weights are set to 1 in this figure.	68
6.1	Selected benchmark sites for the field (a) 1, (b) 2, (c) 3, (d) 4, and (e) 5.	72
6.2	Using apparent soil electrical conductivity map as performance function.	79
6.3	A visualization of the difference of the F-Index of each cell with (a) first quantile, (b) second quantile (median), and (c) third quantile of the F-index of each MZ.	80
6.4	Delineating a field into (a) 6, (b) 12, (c) 20, and (d) 30 MZs results in the same number of selected benchmark sites.	82

7.1	The scene implemented for visualizing and testing the DT algorithm that shows a panel of tools.	87
7.2	The benchmark site selection scene which shows the UI to control the optimization process.	89

Chapter 1

Introduction

There has not been a wholesale shift to the use of geodetic distance, or of globe-based methods for calculating area.

Michael F. Goodchild, Reimagining
the history of GIS

We are gathering an immense amount of data about the earth using satellites, drones, phones and other devices, which is predicted to be around 2.24 exabytes (2,240,000,000 gigabytes) for just one day [1]. This data has great potential to help us understand and predict geospatial phenomena through analysis and simulation. One example of geospatial phenomena is wildfire; according to Natural Resources Canada (NRCan) an average of 2.5 million hectares of land burns annually because of wildfire [2]. By simulating the wildfires, they can be accurately predicted. Another application for geospatial data is agriculture. *Precision Agriculture* relies on various geospatial data including satellite imagery to help us better understand crop health and soil structure. This helps us to direct the resources to the parts of the agricultural fields that need them the most. Directing the resources not only lowers the cost of agriculture but

also reduces the waste and pollution of the environment.

1.1 Problem Statement

When working with geospatial data, distance is an inseparable concept. Accurately measuring the distance is crucial in geospatial analysis for instance for determining resource allocation for agricultural fields or assessing the extent of damage caused by a wildfire. Moreover, determining the distance of all points of a given region to a geospatial feature (e.g. boundary of a wildfire) is commonly needed. Distance transform (DT) is a mapping that specifies the distance of points in a domain to a specified feature and is a fundamental and frequent operation to perform analyses and simulations of geographic data. It has been used for various geospatial analyses including watershed delineation [3], urban planning [3], pipeline route design [4, 5], and mountain railway alignment [6]. DT can be used to find the farthest distance from a geospatial feature which has applications for optimizing soil test points.

Traditional Geographic Information Systems (GIS) are usually based on flat maps, and they have adopted distance transform methods from image processing techniques [3, 7]. Figure 1.1a shows the distance transform applied to a feature inside of an image (i.e. regular 2D grid). Mapping the curved earth into a flat domain introduces distortion, consequently, any distance transform in image space may contain distortion. When a large area is projected, this projection distortion is greater. Therefore, the traditional GIS approach for distance transform is not directly applicable to large-scale applications such as pipeline route design [4]. Aside from the distortion at large-scales, similar to many other operations for the earth, DT can be better presented and interpreted on the globe rather than 2D maps.

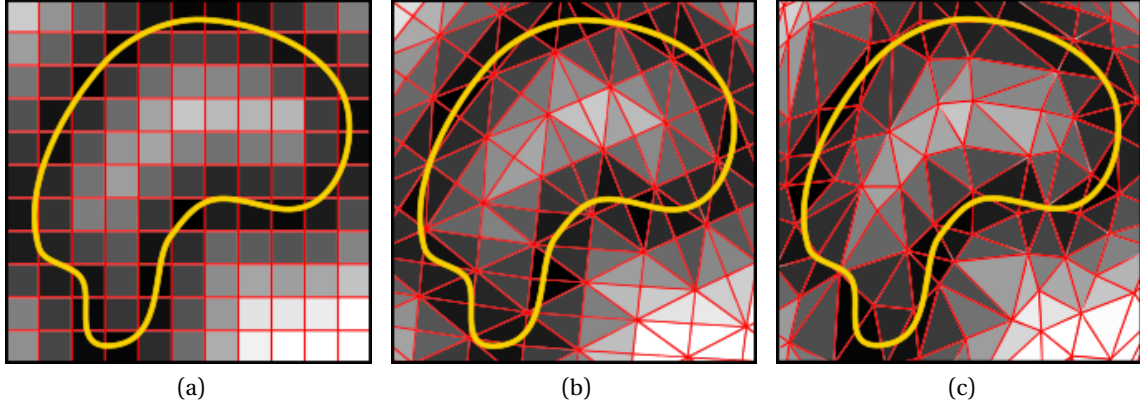


Figure 1.1: DT applied to different types of grids: **(a)** a regular image-based grid, **(b)** a DGGs grid, and **(c)** an irregular general mesh. In this visualization, darker cells are closer to the feature and brighter cells are farther to the feature.

A relatively new GIS approach, Discrete Global Grid Systems (DGGs), is a globe-based representation of the earth that reduces distortion by approximating the earth with a polyhedron [8–10]. A DGGs discretizes the earth into mostly regular cells using multiresolution grid systems. The regularity and multiresolution properties of DGGs are the outcome of the iterative application of a refinement scheme to the initial polyhedron faces [8]. DGGs proved to be useful for complex geospatial data analysis like risk analysis [11] and point cloud processing [12]. However, due to the spherical nature of the globe, it is not possible to find a fully regular discretization for it. Therefore, DT algorithms that are developed for image space will not work on a DGGs. Hence, the problem is how to develop an efficient DT on a DGGs.

1.2 Goals and Scope

Globe-based geospatial systems, including DGGs, promise to address the issues of flat-map GIS. However, developing algorithms for curved shapes is more challenging compared to flat-map GIS. The main goal of this thesis is to develop an efficient DT algorithm on top

of a DGGS. Our DT algorithm does not make any assumptions about the shape of the DGGS cells, the type of refinement, and the underlying projection, which is an advantage and makes this algorithm general and applicable to many DGGSs. Our DT algorithm assumes that the fundamental operation for traversing the DGGS hierarchy is given by the DGGS. After developing the DT algorithm, we evaluate the correctness and performance of our DT algorithm. Then, we compare our DT algorithm with a flat-map-based method in terms of accuracy, and with a general-mesh-based method in terms of performance.

To more practically evaluate our DT algorithm, we use it to solve the real-world problem of optimizing soil test points in precision agriculture. The goal is to use the DT to efficiently calculate different metrics for optimization algorithms. The calculation of DT on DGGS helps us to automate the test point selection process.

1.3 Methodology

It is not possible to find a fully regular multi-resolution discretization for the globe, hence every DGGS' grid is only semi-regular which makes it more challenging than a fully regular grid. However, there is a possibility of exploiting this semi-regularity to develop a more efficient algorithm in comparison with irregular grids (see Figure 1.1b and Figure 1.1c). In the context of general 3D meshes, DT algorithms are developed based on geodesic distance calculations [13, 14]. These general mesh algorithms can be applied to the DGGS grid but are slower, especially with larger grids. For the DGGS grid, we can use specific properties of DGGS to develop a more efficient algorithm. The traditional distance transform algorithms work on either perfectly regular grids (i.e., images) or general meshes. These methods are not fit for the semi-regular mesh of a DGGS and do not exploit the hierarchy of the underlying multiresolution grid. Thus, a novel approach is needed. To address this need, in this thesis, we introduce a new, efficient distance transform algorithm for DGGS.

In DGGS, we define distance transform as the distance of a set of cells to one or a set of features. We use the properties of a DGGS, especially the hierarchy and the geometry of the DGGS, to design an efficient distance transform algorithm. Our algorithm is based on a coarse to fine hierarchical traversal of the DGGS. We start with a coarse resolution and calculate the distance of each coarse cell to the feature. This step is efficient because there is a small number of cells in this resolution. Next, based on calculated distances on the current coarse resolution, we reduce the search space for the cells in higher resolutions and store all the relevant edges of the feature in a data structure. We then iteratively refine the grid to a higher resolution and make use of the pre-calculated search space to find the distance of child cells to the feature. We show that the distance of the child cells is guaranteed to fall within the proximity of parent cells.

Case Study for Evaluation of DT in Practical Application

To evaluate our distance transform algorithm, we chose the real-world problem of optimizing soil test points in agriculture. Precision agriculture relies on soil conditions resulting from soil testing across the field. Soil testing is expensive, and reducing the number of samples is an important task. One viable approach is to divide the farm fields into homogeneous management zones that require only one soil sample. As a result, these sample points must be the best representative of the management zones and satisfy some other geospatial conditions, such as accessibility and being away from headlands. We use DT to calculate the distance of a high-resolution sampling (DGGS cells in high-resolution) of the field to the complex management zone boundaries (see Figure 1.2). The areas of each management zone that have the maximum distance to the management zone boundaries are candidates for representative areas. DT is also used to avoid the proximity of the boundary of the field, as these areas are not representative of the field as well as to find the distance to other features.

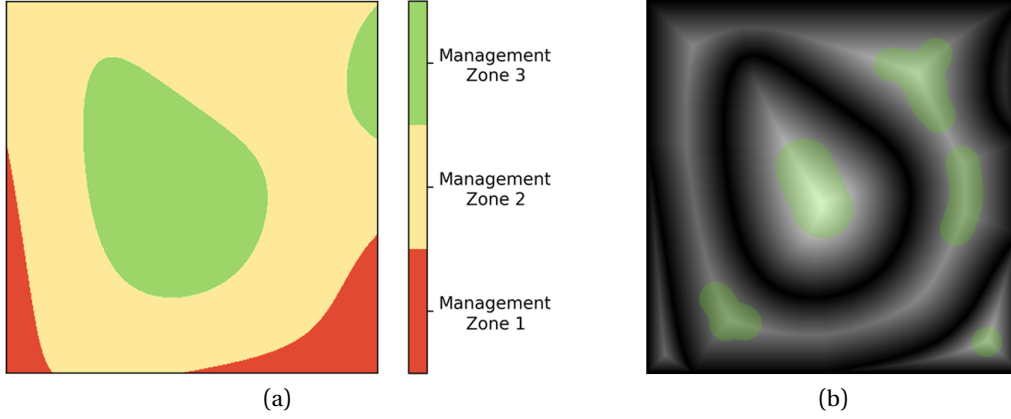


Figure 1.2: **(a)** Imaginary management zones for a field and **(b)**, the result of applying DT to the management zone boundaries. Highlighted areas in (b) are candidates for test point samples.

The conventional GIS process for selecting these points is slow and tedious. In this case study, we introduce a framework for automatically determining locations for test points using a constrained multi-objective optimization model. We define the objectives for a representative area of a management zone. In this model, DT has an essential role in calculating multiple objectives and constraints efficiently. Then, our algorithm optimizes the objectives using a scalarization method while avoiding constraints. We assess our method by testing it on five fields and showing that it generates optimal results. This method is fast, repeatable, and extendable.

1.4 Contribution

The main contributions of this thesis fall into two parts:

1. A new method for efficient calculation of DT on DGGS is introduced. A mathematical theorem is introduced and proved, that allows for designing an efficient algorithm. The performance of this algorithm is also analyzed and reported. Our DT algorithm is

compared to image-based and mesh-based methods. It is shown that our DT algorithm reduces distortion compared to image-based and mesh-based methods and is more efficient than mesh-based methods.

2. DT is applied to a real-world case study to automate the process of selecting soil test points in agricultural fields. Selecting soil test points is a time-consuming task and with the help of DT, this task is automated. It is shown that this automated method is efficient and produces optimal test points.

Chapters 3 and 4 are extended and restructured from the paper “Efficient Calculation of Distance Transform on Discrete Global Grid Systems” by Kazemi, M. and Wecker, L. and Samavati, F. published in *ISPRS International Journal of Geo-Information*, in 2022 [15]. Chapters 5 and 6 are extended and restructured from the paper “Automatic Site Selection in Management Zones Using Multi-Objective Optimization on Remote Sensing Data” by Kazemi, M. and Samavati, F. This paper is currently under review.

1.5 Thesis Overview

The outline for the remaining chapters of this thesis is as follows: Chapter 2 covers the background information and related work on both DT and the case study to give this work context. Chapter 3 introduces a mathematical theorem and builds an efficient DT algorithm based on that. Chapter 4 then covers the correctness and performance analysis of the developed DT algorithm. It also compares the developed DT algorithm with image-based and mesh-based algorithms. Chapter 5 covers how DT is applied to our case study of selecting soil test points. Chapter 6 then presents the result of the case study and covers provides evidence of the optimality of the results. Chapter 7 covers the implementation details of both the DT algorithm

and the case study. Finally, Chapter 8 concludes with a summary of the thesis, along with directions for future work.

Chapter 2

Background and Related Work

In this chapter, we discuss background information for this thesis. We explain related works and papers to state our work among others. Sections 2.1 through 2.4 present necessary background information to understand the DT algorithm, and sections 2.5 through 2.8 present necessary information to understand the case study of DT in agricultural soil sampling.

The basic definition of DT on image space and its applications are presented in section 2.1. Then, section 2.2 presents the prerequisite information about DGGS that gives context to this work. Next, in section 2.3, we briefly discuss how distances are calculated in this work, and in section 2.4, we discuss different algorithms and methods used to compute the DT on various domains as well as the advantages and disadvantages of each method.

Section 2.5 gives a general high-level context to our real-world use case by explaining the importance of soil sampling and the conventional methods used for soil sampling. Then, sections 2.6 and 2.7, give more details about the specific problem in soil sampling that we address. Finally, section 2.8 explains the general method used for solving a multi-objective optimization problem which gives context to the final piece of the case study.

2.1 Distance Transform Definition and Applications

In 1966 Rosenfeld et al. introduced DT as a sequential operation in digital picture processing with applications in shape skeletonization [16]. In the very basic form of Image Space, DT is a transformation from a binary image in which black pixels are object(s) or feature and white pixels are the background, to a gray-scale image. In this gray-scale image, the gray level shows the distance of a background pixel to the feature. Figure 2.1 shows the distance transform applied to a binary image. By looking at Figure 2.1b, it is obvious that the skeleton of the shape can be extracted by following the bright pixels of the image.

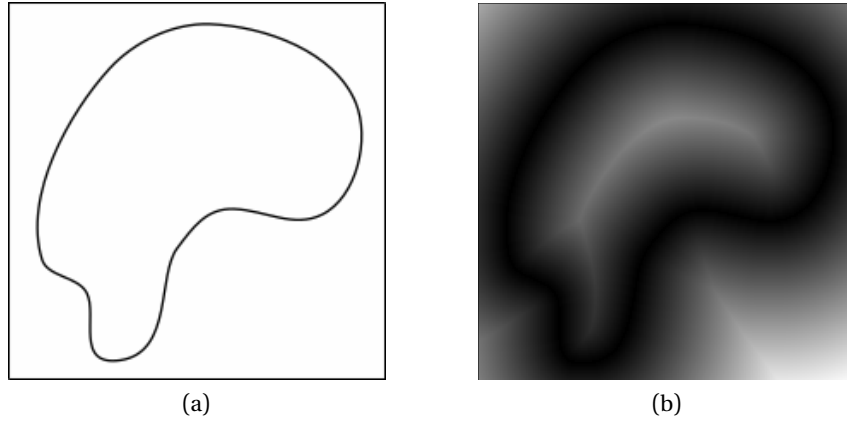


Figure 2.1: **(a)** The binary image before applying DT. **(b)** The result of applying DT to the binary image.

After that, the idea of DT has been applied to many different areas and has applications in medical image processing [17, 18], shape analysis [19–21], computer graphics [22], shortest path computation [23], image segmentation [24], and Convolutional Neural Networks [25], to name a few. Also, different distance metrics such as Manhattan distance [16], Chessboard distance [26, 27], and Chamfer distance [28] have also been used to find distance transform. However, Euclidean distance is still required for many of these applications [29].

Besides the various mentioned applications of DT, de Smith [3] showed that DT is useful for many geospatial applications too. For example, DT may be used for the computation of multilevel buffer zones for watershed delineation and slope lines [3]. DT is also useful to construct voronoi regions which is useful for urban planning such as building hospitals and schools in a city or managing a rescue team in an area [3]. Furthermore, DT has been used for large-scale construction planning such as pipeline route design [4, 5], and mountain railway alignment [6]. Smart agriculture is another example which makes use of DT. When sampling soil from a field, DT can be used to ensure sample points are far enough away from undesirable areas such as the boundary of the field or known areas of contamination [30].

2.2 Discrete Global Grid Systems

DGGS is a novel approach to GIS which approximates the earth with a polyhedron to make a global, universal representation of the earth with less distortion compared to flat maps [8]. DGGS is a discretized, hierarchical, and cell-based representation of the earth that provides efficient neighbourhood access and parent-child traversal [8, 9]. Every DGGS is made of the following main elements.

Initial Polyhedron

Figure 2.2 shows some of the initial polyhedrons that have been used to create a DGGS [8, 9]. The closer this polyhedron is to the surface of the earth, the less the projection distortion. In this work, we use a Disdyakis Triacontahedron DGGS [31].

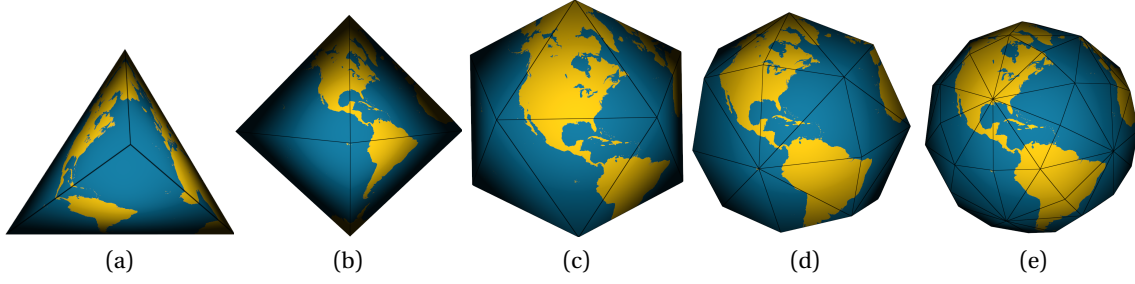


Figure 2.2: **(a)** Tetrahedron **(b)** Octahedron **(c)** Icosahedron **(d)** Disdyakis dodecahedron **(e)** Disdyakis triacontahedron

Refinement Scheme

Cells resulting from the faces of the initial polyhedron are usually considered as the first resolution of the DGGS (i.e. zero level of refinement). A refinement scheme is applied to the faces of the polyhedron to make higher resolutions and a set of finer cells. This refinement can be congruent or non-congruent [8, 9]. Figure 2.3 shows examples of refinements. The resolution at which the data is being presented is the target resolution of the DGGS. Figure 2.4 shows the Disdyakis Triacontahedron DGGS at different refinement levels [31]. At resolution 1 the average area of the Disdyakis Triacontahedron DGGS cells is around 4250546.6 km^2 . As this DGGS uses a 1 to 4 refinement each consecutive resolution reduces the area by a factor of $1/4$. This results in an average area of 15.4 m^2 at resolution 20.

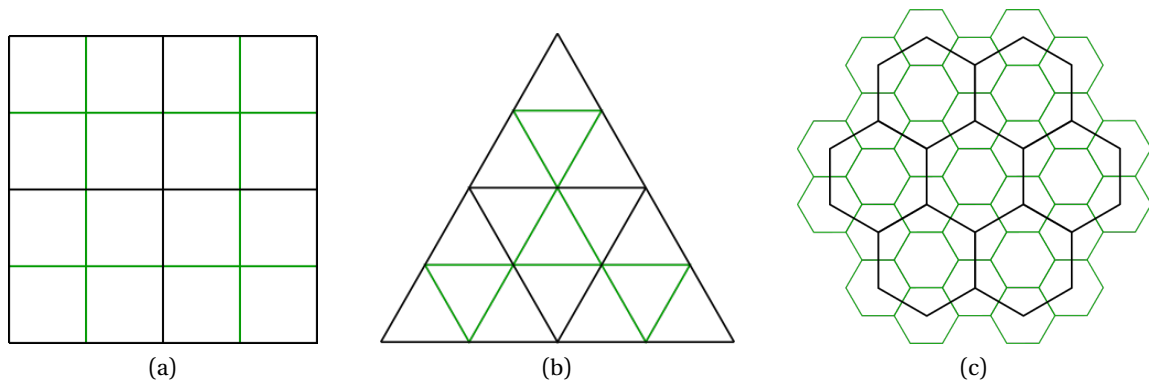


Figure 2.3: Examples of congruent **(a)** 1 to 4 quad, **(b)** 1 to 3 triangle, and **(c)** non-congruent hexagonal refinement. The black cells are original cells and the green lines show one level of refinement.

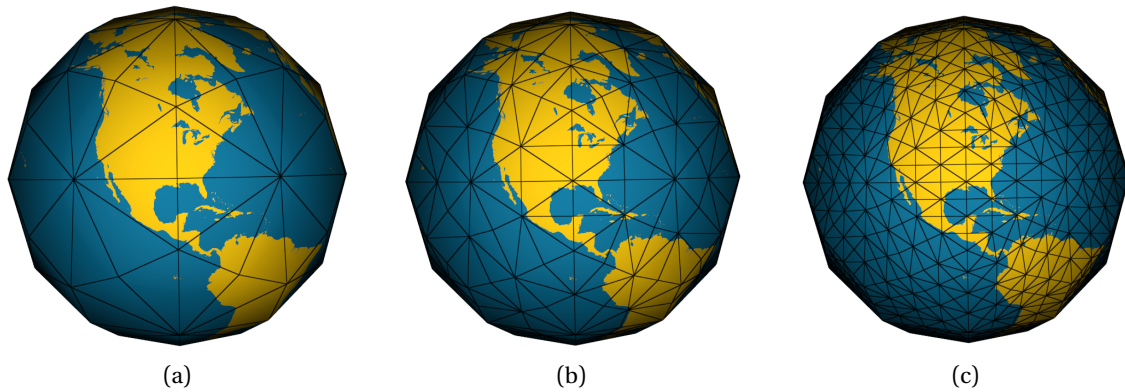


Figure 2.4: Disdyakis Triacontahedron DGGS at resolution **(a)** 1, **(b)** 2, and **(c)** 3.

Cell Shape

The cell shape of a DGGS naturally comes from the choice of the initial polyhedron and the refinement scheme. The most common cell shapes are quads, hexagons, and triangles [8, 9].

Projection

Within a DGGS, projection is the method of transferring information between the polyhedral domain and the spherical domain of the earth. Figure 2.5 shows this projection for Disdyakis Triacontahedron DGGS as an example [31].

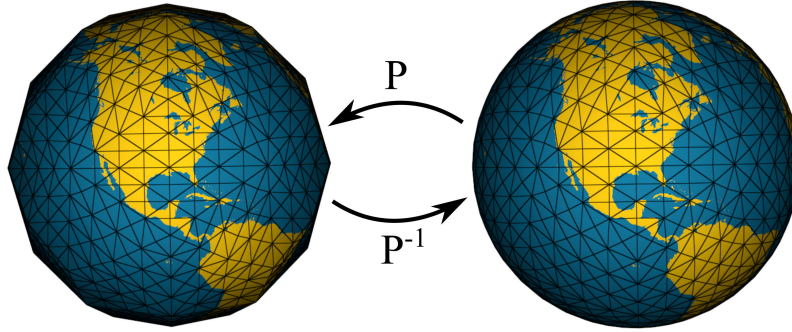


Figure 2.5: Disdyakis Triacontahedron DGGS at resolution three in the polyhedral domain (left) and the spherical domain (right).

Cell Indexing

To assign and retrieve data to and from the cells, we need to assign some indices to the cells. Each index uniquely identifies one cell of the DGGS and a database can rely on these indices instead of coordinates to store the data.

2.3 Distance on Globe

The real distance between two points on the earth depends on the topology of the earth between those two points. However, this distance is difficult to compute, which is why an approximation of the earth is often used to measure distances between points. In this work, the distance between two points is calculated on a spherical domain via great-circle arc (i.e.,

geodesic) calculations. We use the following formula to calculate the distance between the two points of p and q on the sphere (see Figure 2.6) with r representing the radius of the earth.

$$Distance(p, q) = acos(\frac{\vec{p}}{|\vec{p}|} \cdot \frac{\vec{q}}{|\vec{q}|}) * r$$

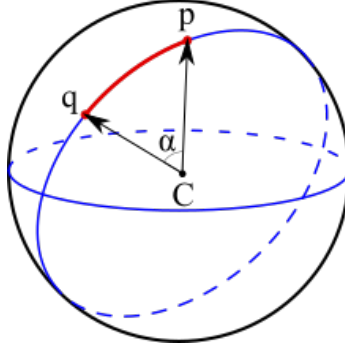


Figure 2.6: The red line is the great-circle arc connecting points p and q .

The distance of a point on earth to a great circle arc is also needed in this work which is presented in Algorithm 1. In this algorithm, the arc is represented by the two endpoints which are connected to each other with a great-circle arc. It is assumed that the point and the two endpoints are on the spherical domain.

Figure 2.7 gives some geometric context. We assume a unit sphere. Vectors \vec{v} and \vec{w} form a plane. The vector $\vec{N} = \vec{v} \times \vec{w}$ is the normal to that plane. We project the point p to that plane. Then, by normalizing that projected vector, we push it to the great circle that intersects v and w (this is p' and line 3 in Algorithm 1). So, p' lies on the great circle, but we do not know if p' is between v and w (on the arc) or outside of it. We test it by calculating the cross vectors $\vec{p}' \times \vec{v}$ and $\vec{p}' \times \vec{w}$. The results of these crosses are also normal to the aforementioned plane. If the cross vectors point in different directions (180° angle between them), then p' is between v and w , otherwise (0° angle between them), p' is outside of them.

Algorithm 1 Point To Great Circle Arc Distance

Input: point, arc

Output: distance

```
1: normal  $\leftarrow$  normalize( $v$ .cross( $w$ ))
2: pDotNormal  $\leftarrow$   $p$ .dot(normal)
3: projection  $\leftarrow$  normalize(point - (pDotNormal * normal))
4: cross1  $\leftarrow$  projection.cross( $v$ )
5: cross2  $\leftarrow$  projection.cross( $w$ )
6: dotOfCrosses  $\leftarrow$  cross1.dot(cross2)
7: if dotOfCrosses < 0 then
8:   distance  $\leftarrow$  Distance(point, projection)
9: else
10:  d1  $\leftarrow$  Distance(point,  $v$ )
11:  d2  $\leftarrow$  Distance(point,  $w$ )
12:  distance  $\leftarrow$  min(d1, d2)
13: end if
14: return distance
```

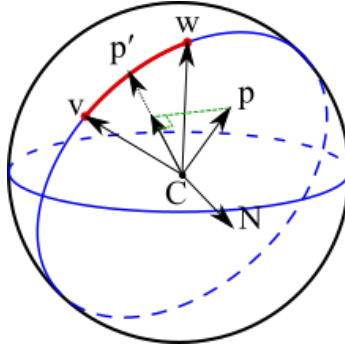


Figure 2.7: The red line is the great circle arc connecting points v and w , and point p is the point in question.

2.4 Computing Distance Transform

While a DGGS has benefits over a traditional GIS, the problem of distance transform on DGGS is not investigated in the literature. In this section, we see how distance transform is computed in image space and mesh space. Image space is the perfectly regular end of the

spectrum where distance transform algorithms are efficient. At the other end of the spectrum are irregular general meshes in which distance transform is inefficient. DGGS is in the middle of this spectrum where there is some level of regularity but they are not perfectly regular.

Computing DT on Image Space

Distance transform has been studied extensively in image processing for 2D images since [16] (See also [29]). It can be computed efficiently in perfectly regular 2D domains, for example, see [32–34]. However, these algorithms exploit full regularity of the image domain and applying them to a DGGS grid poses some challenges. One challenge is that the concept of neighbour and distance are not aligned in DGGS grid (i.e., The neighbours from different directions usually do not have the same distance). The second challenge is that the DGGS grid is not perfectly regular but semi-regular.

Another flaw of using image-based algorithms for GIS applications is accuracy. De smith's work [3] uses image-based algorithms to do a distance transform, thus it is required to project the surface of the earth onto a 2D plane which produces distortion. Using recent methods of projection in GIS, the distance distortion in a small-scale, such as a city, might be negligible. However, in large-scale applications, this method has measurable distortion which impacts accuracy. Our algorithm, in contrast, calculates the distance transform on the globe, which means the algorithm is applicable to larger scale applications such as pipeline route design [4, 5], mountain railway alignment [6], and those which operate on a global scale.

Computing DT on Mesh Space

Computing DT on a general mesh is more difficult than on 2D images due to the impact of curvature on distance, as well as the irregular connectivity of general meshes. In 1987, Mitchell et al. introduced an algorithm that computes the exact geodesic distance on a triangular mesh

[35]. This algorithm is called the MMP algorithm after the initials of its authors. However, there was no implementation of an MMP algorithm for 17 years, and it was not until 2005 when Surazhsky et al. introduced one [36]. The challenge with this work is that it works only for point features. Bommers et al. generalized the MMP algorithm to work with any arbitrary vector feature [13]. Approximations of geodesic distance have also been investigated in computer graphics for example by solving the heat equation on the surface of a mesh [14]. However, in GIS exact distance calculations are important. While these algorithms can be applied to a DGGS, they do not exploit any regularity or hierarchy within it.

2.5 Soil Sampling Techniques in Agriculture

To understand the context of our case study, in this section, we present the importance of soil sampling and different techniques of soil sampling. The productivity of agriculture depends on the soil nutrients which are maintained by fertilizers. To understand the health of the soil and apply the right amount and type of fertilizer, a soil test is needed. This test helps to determine the deficiency of soil nutrients. However, soil testing is expensive [37], and as a result, based on a report from the United States Department of Agriculture (USDA), only around 30% of farms in the U.S. adopted soil testing methods [38]. Therefore, it is essential to minimize the cost of soil testing by minimizing the number of soil tests while trying to capture the overall soil condition across the field [39]. The traditional methods of soil sampling include composite sampling (Figure 2.8a) and grid sampling (Figure 2.8b) [40, 41]. Composite sampling is the practice of collecting soil from various random locations in the field and mixing those samples to make a composite sample. This single composite sample then is sent to a laboratory for soil testing which gives an average understanding of the soil of the entire field. However, it is common to expect a large variability of nutrients across fields. Therefore, composite sampling may not be precise, particularly for larger fields. On the contrary, grid sampling gives a more

accurate understanding of the field (depending on the grid size) by subdividing the entire field into many smaller regions (subfields) and sampling and testing each subfield. Depending on the number of subfields, the main drawback of grid sampling is the higher cost due to excessive testing.

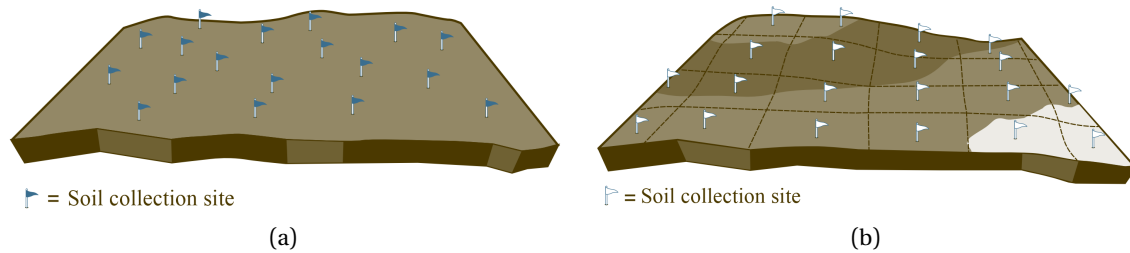


Figure 2.8: Traditional methods of soil sampling (a) composite sampling and, (b) grid sampling. Blue flags show the sample points used to make the composite sample (only one test), while white flags are sampled and tested individually. (Figures are taken from [41])

Precision Agriculture relies on new technologies and various data to help us better understand the soil structure and facilitate soil testing by carefully selecting a small number of sites to test the soil [42]. This method increases crop yield and profitability while lowering the levels of traditional inputs needed to grow crops, such as fertilizers [43]. One of the most popular approaches in precision agriculture is *Soil Sampling Management Zones* or *Site-specific Crop Management* [42]. In this method, depicted in Figure 2.9a, a field is divided into several Management Zones (MZ) which are relatively homogeneous sub-units of a farm field that can each be treated uniformly. The MZs are usually delineated by a multilevel thresholding method (from now on, simply referred to as thresholding) on a performance function across the field (e.g., historical yield). Figure 2.10 shows an example of a performance function across the field and how thresholding this function makes the MZs. By establishing MZs, each MZ is considered homogeneous in the composite soil sampling method. As shown in Figure 2.9b, a composite sample is done per MZ, which lowers the cost of testing compared to grid sampling by reducing

the number of tests.

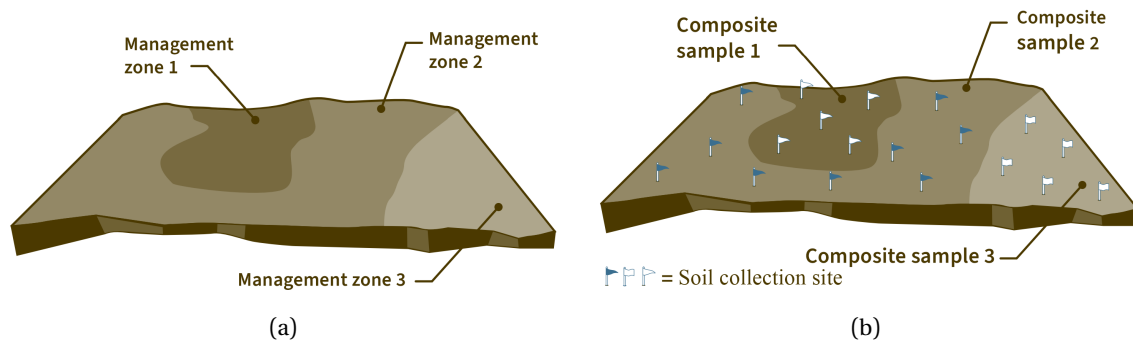


Figure 2.9: Soil sampling management zones practice. (a) homogeneous sub-units of a farm field, (b) composite sampling from each MZ. The soil collected from similar-shaped flags mixes to form a composite sample, which means only three soil tests will be done. (Figures are adapted from [41])

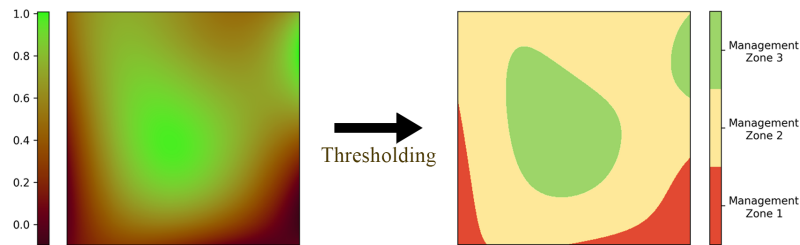


Figure 2.10: The process of MZ delineation from a performance function across the field.

Another practice of soil sampling is benchmark sampling. In this practice, the soil is sampled from a benchmark site — which is a very small area within the entire field or within each MZ — rather than making a soil composite [41]. Keyes and Gillund showed that benchmark sampling can replace composite sampling without losing the test precision [44]. The benefit of benchmark sampling is that by sampling from the same benchmark sites over the years, the trend of soil nutrient changes can be tracked [40]. For this method of sampling, it is important to select a small area of the field as a benchmark site that is representative of the soil of the field. Choosing a single benchmark site for the entire field suffers again from the issue of not accounting for

soil variability. Instead, using the soil sampling management zone technique, benchmark sampling can be done per MZ as shown in Figure 2.11, and the result of the test is applicable to all parts of that MZ. This method of testing not only provides precise information about the soil composition but also reveals the trend of the nutrient changes for each MZ while remaining a cheap method of soil testing.

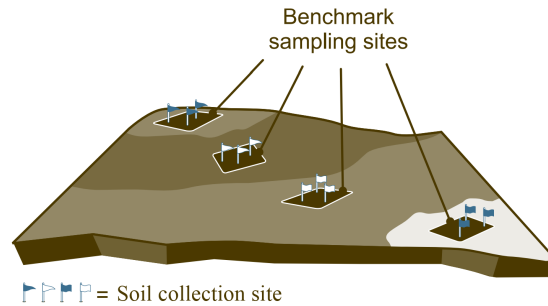


Figure 2.11: Benchmark sampling using management zone technique. (Figure is adapted from [41])

2.6 Delineation of Management Zones

Management zones play an important role in our case study. Therefore, we need to know how they are constructed before discussing how to select the benchmark sites. Different delineation methods for site-specific management zones exist based on different information, usually yield maps, soil properties, topographic properties, electrical conductivity data, or remote sensing and vegetation indices. Hornung et al. [45] have compared the soil-colour-based and yield-based methods. Both methods require the availability of agricultural maps such as soil colour and texture maps, yield maps, and topography maps. Fraisse et al. [46] proposed a method which relies on soil electrical conductivity data. However seasonal effects like weather have an impact on the electrical conductivity, so the electrical conductivity map of a field cannot be compared to one from another field.

Satellite imagery presents a great added value because of its availability and relatively low cost. Georgi et al. [47] propose an automatic delineation algorithm based on only satellite remote sensing data. In this method, they select satellite images of the field that show the spatial pattern of plant growth. Then, they extract the near-infrared (NIR) band of the selected images and average them over years to form the performance function. In this thesis, we use a similar method that constructs the performance function by averaging the selected Normalized Difference Vegetation Index (NDVI) images from satellite remote sensing data [30]. The details of this method are discussed in 5.2.

2.7 Benchmark Soil Sampling

Benchmark soil sampling involves taking soil samples from a designated benchmark site, which is a small area within either the entire field or each MZ [40]. Research has shown that benchmark sampling is comparable in accuracy to composite sampling in terms of test precision [44]. The advantage of benchmark sampling is that by consistently sampling from the same benchmark sites over time, changes in soil nutrient trends can be monitored [40]. Due to soil variability, different sampling designs try to find representative points of a field with the help of different sensory information (e.g. electrical conductivity of the soil) [48, 49]. Another sampling scheme acts in multiple stages where first, a large number of points are generated and then filtered based on some criteria [50]. One study introduces a method for identifying the representative sampled points from a set of already sampled and tested points using clustering algorithms and calculating fuzzy membership values for each point [51]. Overall, finding representative areas of the field remains a challenge for benchmark sampling. Several studies have suggested that management zone delineation provides a basis for benchmark soil sampling [52–56]. Although MZs are assumed to be homogeneous for management purposes, the soil within each MZ still varies by location. Therefore, it is important to carefully select representative

areas rather than making arbitrary choices. Different studies have used approximate measures to manually select representative locations for benchmark sampling [57–59]. For example, the benchmark sites in Alberta were selected to be representative of soil-landscape and agricultural land-use found in the agricultural area, under the Alberta Environment Sustainable Agriculture (AESAs) Soil Quality Monitoring Program [57].

Benchmark sampling is a practical and cost-effective method of soil sampling [60, 61]. Typically, the benchmark sampling sites are selected by experts after field surveys, or by farmers using some guidelines like in [41]. However, the literature lacks algorithmic methods for selecting benchmark sites from MZs. This thesis addresses this gap by introducing an algorithm for the automatic selection of benchmark sites that relies solely on satellite images and does not need any auxiliary data (e.g. electrical conductivity sensors).

2.8 Solving a multi-objective optimization problem

A multi-objective optimization problem is not trivial to solve. The preferences of objectives should be given for each specific optimization problem, otherwise, there is no optimal answer to a multi-objective optimization problem. In this case, a set of answers can only be *Pareto optimal* [62]. A Pareto optimal answer means you can not make any objective of that better without making other objectives worse. In general, methods for solving a multi-objective optimization problem can be categorized into three categories: methods with (1) a priori articulation of preferences, (2) posteriori articulation of preferences, and (3) no articulation of preferences [62]. The most common and general method of solving is to combine all objectives into a function to make it a single objective function which is called *scalarization*. However, different scalarization methods, such as Global Criterion, Achievement Function, Compromise Function, and Objective Product use different combinations [62]. If these methods implement a weighting system for combination, they are a method with a priori articulation of preferences.

A weighted scalarization method converts a multi-objective optimization problem into a single-objective optimization problem. The typical method of solving a single-objective optimization problem is solving for critical points directly if the mathematical expression of the objective function is well-known and is differentiable (e.g., linear least squares) [63, 64]. When the mathematical expression for the objective function is unknown, gradient-based methods such as gradient descent, Newton's method, or the Quasi-Newton method can be used [63, 64]. These methods typically start from a random point and rely on the gradient to find the direction of the steepest descent. The gradient is usually approximated by numerical methods for the objective function. This process is iteratively repeated until the algorithm converges. However, these methods may converge to a local minimum depending on the starting point. If the domain of the objective function is discrete and bounded (e.g., raster data such as satellite imagery), a complete search method can be applied by evaluating the objective function for each value in the domain (e.g., each pixel in the raster). This approach guarantees to find the global minimum but can be expensive if the domain is large. DGGS, as a data integration platform, is a discretization of the earth; hence, it can be naturally used to evaluate and solve an optimization problem efficiently. Moreover, we can control the size of the search space by changing the resolution of the DGGS to tune the trade-off between performance and accuracy.

Chapter 3

Calculating Distance Transform on DGGS

The main goal of this thesis is to develop an efficient algorithm for computing a distance transform on a DGGS. In contrast to image space and mesh space algorithms, our method exploits the hierarchy and the geometric properties of a DGGS grid. This allows our method to be more efficient than the general mesh-based versions. In Section 3.1, we explain how DGGS hierarchy is exploited to reduce the search space of the DT. Section 3.2, is introducing two algorithms that rely on the DGGS hierarchy to calculate DT.

3.1 Reducing the Search Space for Developing an Efficient Algorithm

The DT algorithm relies on a massive reduction of search space using the hierarchy of DGGS cells. We first discuss and prove a theorem that enables us to exploit the hierarchy to reduce the search space.

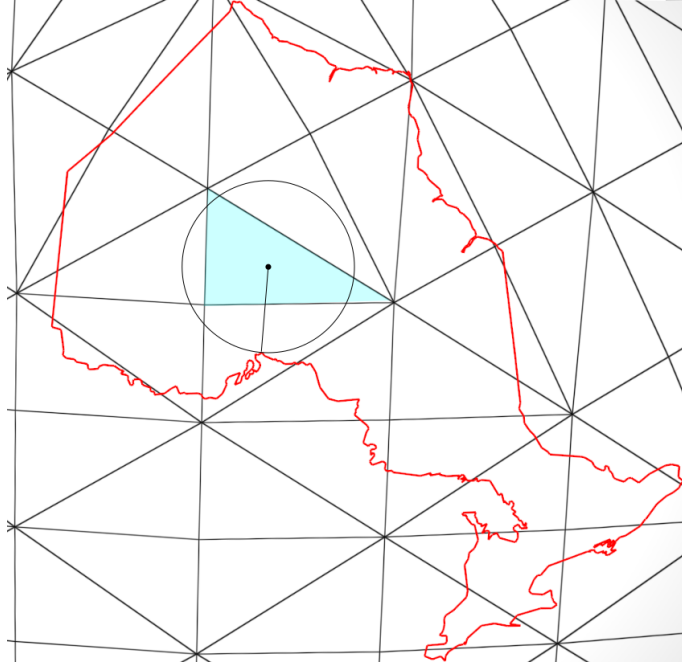


Figure 3.1: Distance of a cell to a feature.

Our algorithm needs three main inputs. First is a vector feature, e.g. the border of a country. This input consists of a list of edges, where each edge is defined by two points on the earth that are connected by a great-circle arc. The second input is the DGGs that we want to operate on. The DGGs enables us to utilize the hierarchical grid to optimize the algorithm. And the last input is the resolution of the DGGs on which DT needs to be calculated. We call this resolution the target resolution. The goal is to compute the minimum distance from each cell of a region at the target resolution to the feature. The minimum distance from a cell to the feature is defined as the minimum distance from a representative point of the cell to the feature. Like Figure 3.1, this representative point can naturally be the centroid of the cell, but any other interior point would work with our algorithm too. The objective is to compute such a distance for all of the cells of the region at the target resolution to form a distance field.

To exploit the hierarchical nature of a DGGs, our algorithm starts at a coarse resolution

which has few cells and does not require many distance calculations. When refining the grid, we make use of the calculations on the coarse resolution to reduce the number of distance calculations at the target resolution. We accomplish this by introducing and proving Theorem 1, which allows us to iteratively reduce the search space when traversing to higher resolutions.

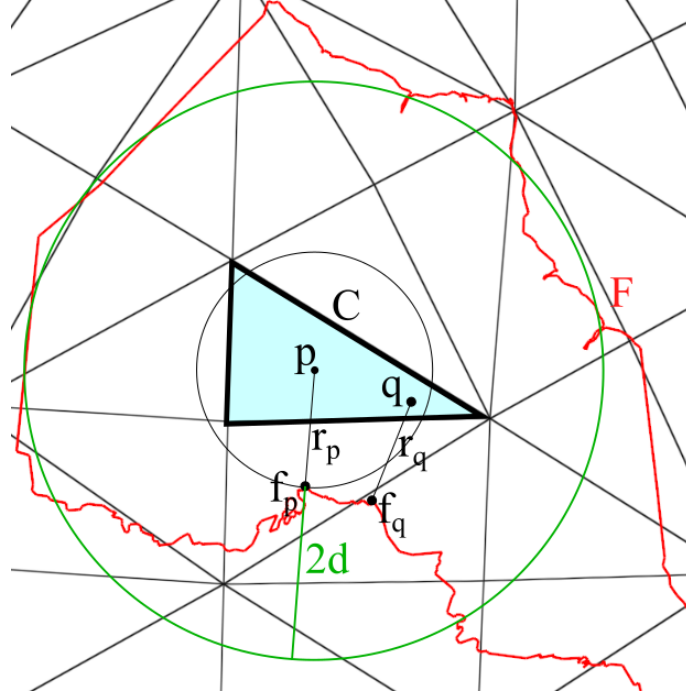


Figure 3.2: An example of a bounding space (green circle).

Before describing the theorem, let us first establish some notations and a simple illustration provided in Figure 3.2. Let p be the representative point of cell C , and r_p be the distance of p to the feature F . As demonstrated in Figure 3.2, $r_p = |p - f_p|$ where f_p is the closest point of feature F to representative point p . During the hierarchical traversal of cells, we must evaluate r_q where q is the representative point of a descendant cell of C . Obviously, the closest point f_q can be different from f_p . However, when p and q are close, the search space of f_q becomes a small subset of F . Theorem 1 reduces the search space of the f_q by providing a bound for this

space.

Theorem 1. *Given an arbitrarily shaped cell C , its representative point p , its distance to feature r_p , and q , the representative point of a descendant cell of C , the search space of f_q is all the edges of the F inside of a circle centred at p with radius $r_p + 2d$, where d is the distance of the farthest possible q to p .*

Based on the Theorem 1, it is guaranteed that the closest point f_q to any point in the highlighted triangle in Figure 3.2, is inside the bounding circle (the green circle).

Proof of Theorem 1. Based on the definition of the r_q and f_q we have (see Figure 3.2):

$$\min_{f \in F} |f - q| = r_q \leq |f_p - q| \quad (3.1)$$

Otherwise, f_p is closer to q than f_q , and it contradicts the assumption that f_q is the closest point of the feature to q . Using triangle inequality we have:

$$|f_p - q| \leq r_p + |p - q| \quad (3.2)$$

Therefore using (3.1) and (3.2) and the definition of d , we obtain:

$$r_q \leq r_p + |p - q| \leq r_p + d \quad (3.3)$$

To find the search space of f_q in respect to p , using triangle inequality, we have:

$$|f_q - p| \leq r_q + |p - q| \quad (3.4)$$

and then using (3.3) we obtain:

$$|f_q - p| \leq r_p + d + |p - q| \leq r_p + 2d \quad (3.5)$$

□



Figure 3.3: Two types of triangle 1 to 4 refinements with appropriate d .

Figure 3.3 shows examples of d in triangular grids with congruent refinements with the assumption that p is the centroid of the shapes. When d is a small value, the search space is smaller and the resulting algorithm becomes faster. The value of d depends on the shape of the cell and the refinement used in DGGS. In general, $d = \max_{q \in \text{children}(C)} |p - q| = |p - q_m|$. For congruent refinements, d can simply be determined using the parent cell's geometry (i.e. q_m is a point on the boundary of the cell C). For non-congruent refinements, d can be determined with a similar method using the footprint of the cell C . For example, Kevin Sahr [65] provides such a footprint for a non-congruent aperture 3 hexagonal tree system. Based on [65], the r shown in Figure 3.4 covers the entire footprint of the ancestor cell. Therefore, this radius can be used as d .

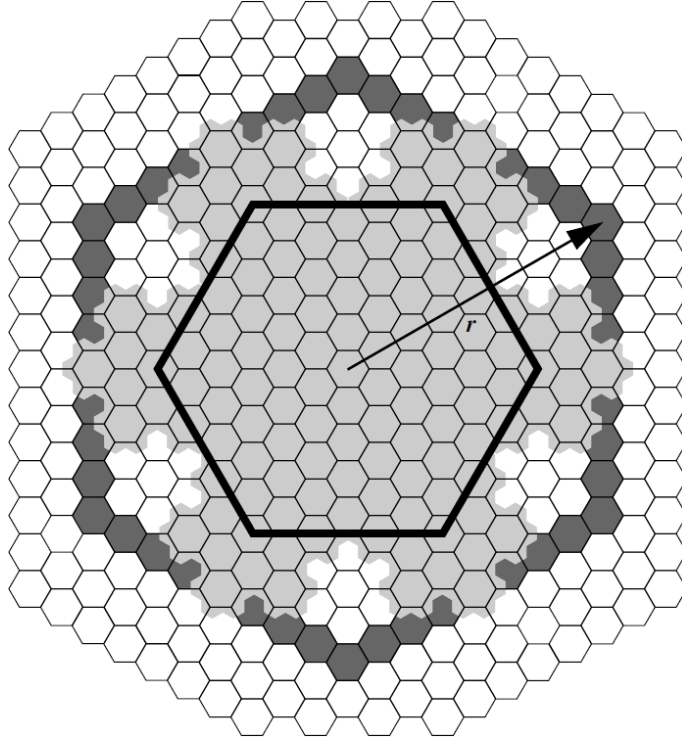


Figure 3.4: The gray cells are the footprint of the large black hexagon at 4 resolutions higher. Image taken from [65]¹.

3.2 An Algorithm to Calculate Distance Transform on DGGS

This section introduces two algorithms that calculate a distance transform on a DGGS grid. The basic idea is presented in the first algorithm and the second algorithm is a modification of the first algorithm which repeats the first algorithm to gain more efficiency. We make use of Theorem 1 to describe the first algorithm as follows.

¹Reprinted from Computers, Environment and Urban Systems, 32, Sahr, K., Location coding on icosahedral aperture 3 hexagon discrete global grids, 186, Copyright 2023, with permission from Elsevier (Licence No. 5612350087487). The copyright is attached in the appendix.

Single Stepping from Base to Target Resolution

To calculate DT on a DGGS, we start from a coarse resolution (i.e., base resolution). Once in this resolution, computing distance using the exhaustive search algorithm becomes very quick. Therefore, for each cell, we check the distance to all edges of the feature and find the minimum. After finding DT in the base resolution, we compute the bounding circle for each cell in it. We then find and store all the edges of the feature that are within this circle; we call this list of edges the “candidate list” of the cell. Based on Theorem 1, we refine the grid to the target resolution and use the stored candidate lists for DT computation of child (or descendant) cells. So in the target resolution, we simply need to check the distance from each cell to the edges of the feature that are stored in their parents’ (or ancestors’) candidate list. This process is presented in Algorithm 2. The pseudocode shows how this algorithm can be implemented in three steps: **1)** calculating the bounding circles and the candidate lists in the base resolution (lines 1-5), **2)** refining the base cells to get the target cells with a jump from the base resolution to the target resolution (lines 6-9), and **3)** calculating the distances in the target resolution based on the candidate lists (lines 10-13).

Algorithm 2 Compute Distance Transform

Input: grid, feature, baseRes, targetRes

Output: distanceField

```
1: baseCells ← getCoveringCellsAtResolution(feature, baseRes)
2: for each cell in baseCells do
3:   distance ← computeDistance(cell, feature)
4:   cell.candidateList ← computeCandidateList(cell, distance, feature)
5: end for
6: targetCells ← new List()
7: for each cell in baseCells do
8:   targetCells.add(refineToRes(cell, targetRes) )
9: end for
10: for each cell in targetCells do
11:   candidateList ← cell.ancestor.candidateList
12:   distanceField.add(cell, computeDistance(cell, candidateList) )
13: end for
14: return distanceField
```

Based on Theorem 1, the candidate list calculated in the base resolution is valid for all descendants of the base resolution. This enables us to jump from the base resolution to the target resolution. However, in the next section, we show how it is preferable to refine the mesh one resolution at a time in order to fully exploit the hierarchy. Algorithm 3 and Algorithm 4 show the details of the *computeCandidateList* and *computeDistance* subroutines, respectively. The implementation of the *pointToGreatCircleArcDistance* subroutine is given in 2.3.

Algorithm 3 Compute Candidate List

Input: cell, distanceToFeature, edgeList**Output:** candidateList

```
1: representative  $\leftarrow$  getRepresentativePoint(cell)
2: d  $\leftarrow$  getD(cell)
3: candidateList  $\leftarrow$  new List()
4: for each edge in edgeList do
5:   distance  $\leftarrow$  pointToGreatCircleArcDistance(representative, edge)
6:   if distance < distanceToFeature + 2d then
7:     candidateList.add(edge)
8:   end if
9: end for
10: return candidateList
```

Algorithm 4 Compute Distance

Input: cell, edgeList**Output:** distance

```
1: representative  $\leftarrow$  getRepresentativePoint(cell)
2: distance  $\leftarrow$   $+\infty$ 
3: for each edge in edgeList do
4:   arcDistance  $\leftarrow$  pointToGreatCircleArcDistance(representative, edge)
5:   distance  $\leftarrow$  min(distance, arcDistance)
6: end for
7: return distance
```

Iterative Refinement

In section 3.2, we discussed how this algorithm is done in a single step from the base resolution to the target resolution. However, this process can be repeated iteratively between the base and the target resolutions to make the candidate lists smaller (i.e., reduce the search space) step-by-step. The algorithm for this modification is presented in Algorithm 5. The basic idea is to go through the grid from the base resolution and refine the grid one resolution at a time to reach the target resolution. This way, we can refine the candidate lists iteratively in each step. Line 2 of Algorithm 5 is the main loop that controls iterations of the algorithm. Lines

3-20 of this algorithm are similar to Algorithm 2 with a small difference in Algorithm 2's first step. To calculate the candidate lists, if it's the first time we are calculating, there is no previous candidate list and no previous distance field (Lines 4-6). The next times, we use the previous candidate list and filter this list to make smaller lists for the next resolution (Lines 7-10).

Algorithm 5 Compute Distance Transform with Iterative Refinement

Input: grid, feature, baseRes, targetRes

Output: distanceField

```

1: coarseCells ← getCoveringCellsAtResolution(feature, baseRes)
2: for currentRes ← coarseRes + 1 to targetRes step 1 do
3:   for each cell in coarseCells do
4:     if isFirstTime then
5:       distance ← computeDistance(cell, feature)
6:       cell.candidateList ← computeCandidateList(cell, distance, feature)
7:     else
8:       distance ← distanceField.getDistance(cell)
9:       cell.candidateList ← computeCandidateList(cell, distance,
        cell.parent.candidateList)
10:    end if
11:  end for
12:  fineCells ← new List()
13:  for each cell in coarseCells do
14:    fineCells.add(refineToRes(cell, currentRes) )
15:  end for
16:  distanceField.clear()
17:  for each cell in fineCells do
18:    candidateList ← cell.parent.candidateList
19:    distanceField.add(cell, computeDistance(cell, candidateList) )
20:  end for
21:  coarseCells ← fineCells
22: end for
23: return distanceField

```

Chapter 4

Distance Transform Results and Discussion

The proposed DT algorithm works with any DGGS grid regardless of the shape of the cells. To test and benchmark the algorithm, we have implemented it on a Disdyakis Triacontahedron DGGS [31]. The cells of this DGGS are triangular cells with one-to-four congruent refinement as shown in Figure 2.4. First we show some visualizations of the output of DT. Then we describe some tests to evaluate the correctness and performance of our algorithm.

Figure 4.1 shows a visualization of DT for Ontario at different target resolutions and Figure 4.2 shows a visualization for three other provinces or territories of Canada. Figure 4.3 shows the result of DT for two smaller scale features, the border of the city of Calgary and a farm field in Alberta.

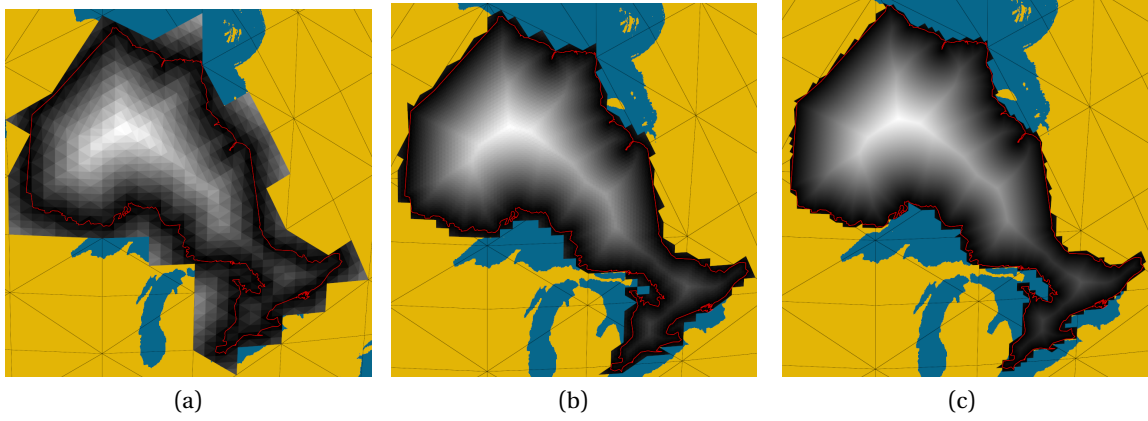


Figure 4.1: Distance transform for the border of Ontario at target resolution **(a)** 7 (Avg. cell size: 1037.3 km^2), **(b)** 9 (Avg. cell size: 64.8 km^2), **(c)** 12 (Avg. cell size: 1.0 km^2).

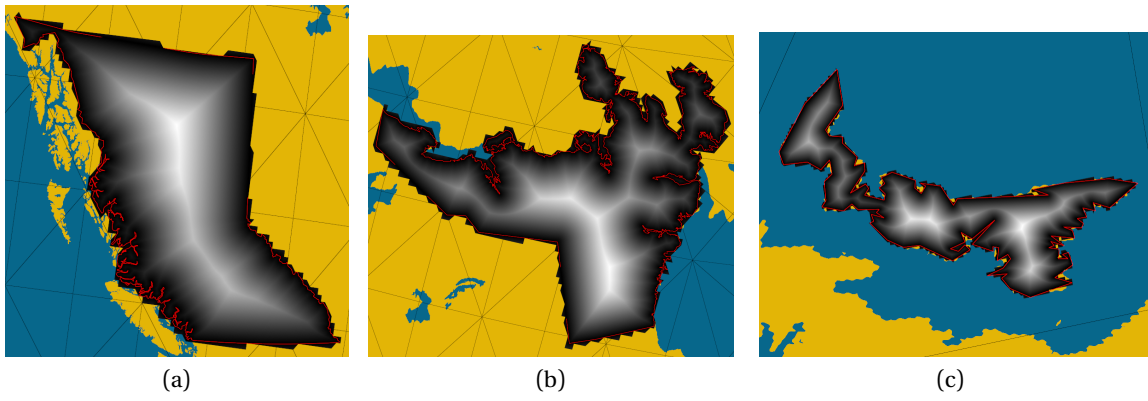


Figure 4.2: Distance transform for the border of **(a)** Mainland British Columbia and **(b)** Nunavut at target resolution 11 (Avg. cell size: 4.0 km^2), and **(c)** Prince's Edward Island at target resolution 15 (Avg. cell size: 0.016 km^2).

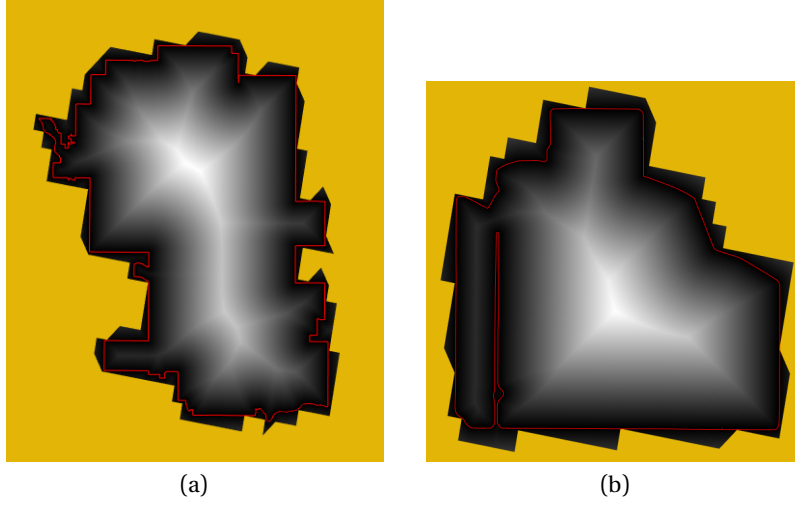


Figure 4.3: Distance transform for the border of (a) the city of Calgary at target resolution 15 (Avg. cell size: 0.016 km^2) and (b) a farm field in Alberta at target resolution 19 (Avg. cell size: 61.8 m^2).

4.1 Correctness Analysis

In sections 3.2 and 3.2, we discussed the correctness of Algorithm 2 and Algorithm 5, which both rely on Theorem 1 to reduce the search space by using the hierarchical properties of a DGGS. In this section, we introduce an empirical test which provides further evidence for the correctness of Algorithm 2 and 5. As the ground truth we use the result of the “Generate Near Table” tool from the proximity tools of ArcGIS Pro with the **geodesic** distance option. To use this tool, we output two pieces of information from our software, (1) the midpoints of the cells used for our DT calculations, and (2) the feature (or boundary) from which the distance is measured. Our system considers line segments of the feature as great-circle arcs, while this is not the case for ArcGIS Pro. To address this issue, we construct a high resolution sampling from the features’ line segments (i.e., 10 meter distance between sample points). At this scale, there is practically no difference between a great-circle arc and a straight line. We then import these two pieces of information into ArcGIS Pro with the spatial reference system EPSG:4047 to match our DGGS

The “Generate Near Table” tool outputs the **geodesic** distance from each point to the boundary. Using this data, we calculate the difference from the distances we calculated for these points using Algorithm 2 and Algorithm 5. Table 4.1 presents the extents of the calculated difference.

Difference	Min (mm)	Max (mm)	Mean (mm)	Std. Deviation (mm)
Ontario (Algorithm 2)	0.0	0.0	0.0	0.0
Calgary (Algorithm 2)	-1.3	1.8	0.0	0.1
Farm field (Algorithm 2)	-6.4	11.4	0.1	0.6
Ontario (Algorithm 5)	0.0	0.0	0.0	0.0
Calgary (Algorithm 5)	-1.3	1.8	0.0	0.1
Farm field (Algorithm 5)	-6.4	11.4	0.1	0.6

Table 4.1: Difference in millimeters between the ArcGIS Pro calculated **geodesic** distances and ours.

Table 4.1 shows the accuracy of our algorithms and demonstrates that our main algorithms, implementation of all subroutines, and results are correct. It also shows that when identical points are given, both Algorithm 2 and Algorithm 5 produce the exact same distance. The small difference between ArcGIS Pro and our algorithm might be caused by boundary resampling or floating point errors.

4.2 Performance Analysis

To analyze performance, we use the boundary of the province Ontario of Canada as the input feature which has 449 edges. In the process of this algorithm, DGGS operations are used to obtain the vertex positions of a cell (used to find representative point of a cell and also to calculate d), and children of a cell. Also, operations to project a point from the polyhedral domain of a DGGS to the spherical domain (used in lines 1-3 of Algorithm 1). The aim of the proposed algorithm is to be efficient with the assumption that DGGS operations are efficient.

In other words, this algorithm tries to be efficient by minimizing the number of distance calculation operations needed to compute the distance field. Therefore, to evaluate the algorithm independent of the efficiency of DGGS operations, we calculate and report the number of distance calculation operations performed for each run of the algorithm.

The distance calculation operation or Algorithm 1, calculates the minimum spherical distance from a point to a great circle arc. This is the unit of work and our algorithm tries to minimize the number of occurrences of this operation. The baseline is that we do not exploit the hierarchy of the DGGS and directly use the target resolution to calculate the distance. To achieve this, for each cell in target resolution, we check the distance from the cell to all of the feature edges. So the total number of operations is the number of cells in target resolution times the number of edges of the feature. In theory, the ideal scenario is that for each cell in the target resolution, we know exactly which edge is the closest and compute the distance of the cell only to that edge. In this case, the total number of operations is only the number of cells in target resolution. Figure 4.5 showcases the dramatic improvement in the performance of our algorithms explained in 3.2 and 3.2 compared to the baseline which makes it possible to calculate DT for higher resolutions.

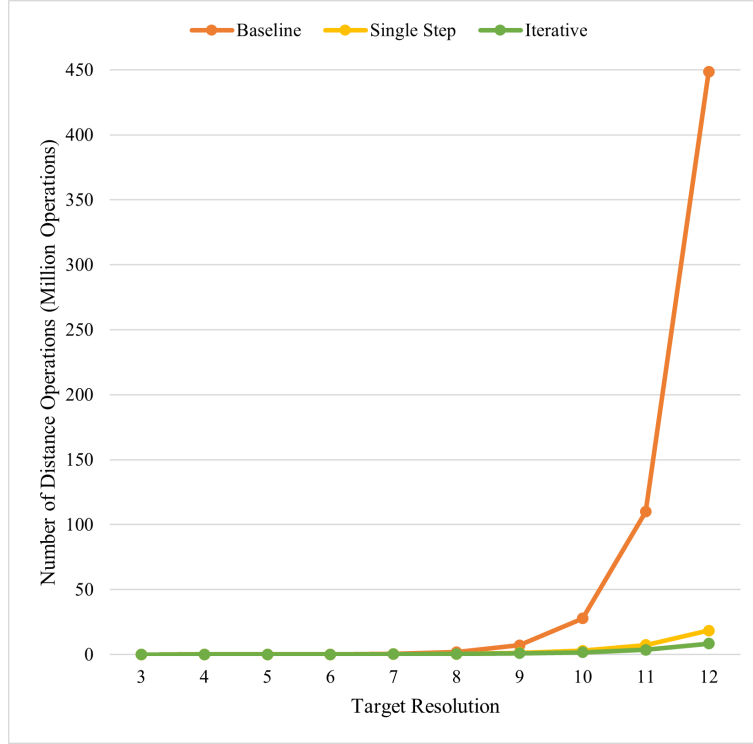


Figure 4.5: The number of distance operations in different target resolutions for the boundary of Ontario.

The graphs in Figure 4.5 are exponential because the number of target cells grows exponentially with the target resolution. To better understand the efficiency of our algorithm, we can slightly change our metric. Figure 4.6 shows the performance of the algorithms in another metric, reporting the total number of operations per target cell. The domain of this metric is between one (being our ideal scenario of knowing the closest edge to each cell exactly) and the number of edges of the feature (449 for the border of Ontario). The closer this number is to one, the better the performance.

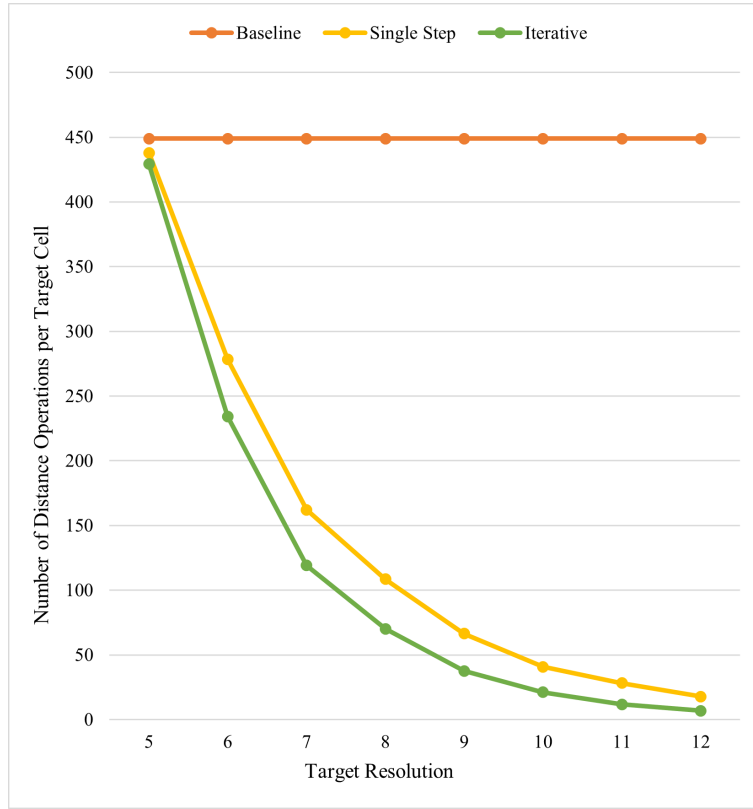


Figure 4.6: The number of distance operations in different target resolutions for the boundary of Ontario.

Interestingly, this metric is a constant for the baseline algorithm due to us checking every edge of the feature for each cell at the target resolution. What's more is that as we observe higher resolutions we can better exploit the hierarchy and get closer to the ideal of 1 operation per target cell.

Based on the analysis presented in this section, the iterative algorithm outperforms the first algorithm. For the boundary of the province of Ontario and the target resolution of 12, the best number of operations per target cell we could achieve using the first algorithm is 17.7. However, using the iterative algorithm with the same inputs, we can achieve 6.9 operations per target cell, which is considerably lower.

4.3 Discussion

For a specific target resolution, the base resolution is an input to our algorithm that doesn't affect the result but gives us greater freedom of choice. We previously discussed that this should be a *coarse* resolution, but a concrete number has not been stated. Figure 4.7 shows the number of operations used to calculate DT in resolution 12 using the iterative algorithm from different base resolutions. We observe that the number of operations almost strictly increases with increasing the base resolution. However, there is no considerable difference between the base resolutions 3 to 7 (minimum of 6.9 operations and maximum of 7.5 operations). It is clear that if the resolution of the base is very close to the target resolution, we do not gain a large improvement. Also, our algorithm is not sensitive to this input as long as a coarse resolution is chosen.

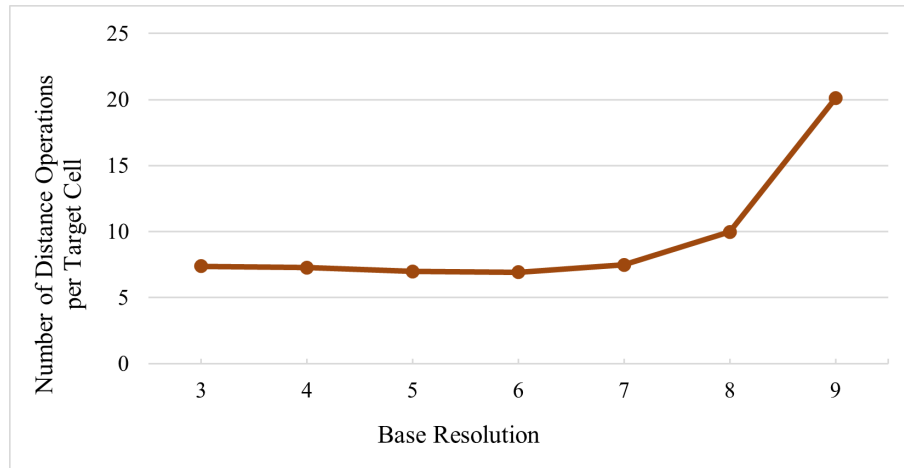


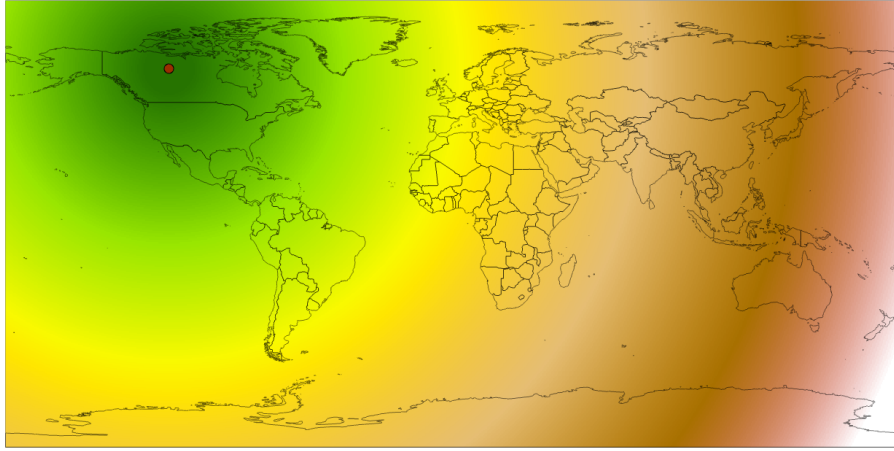
Figure 4.7: The effect of the base resolution on the performance of the iterative algorithm for the target resolution 12.

The single-step algorithm calculates the candidate lists only in the base resolution first and then jumps to the target resolution. The iterative algorithm on the other hand, after calculating the candidate lists in the base resolutions, refines the candidate lists in every resolution in

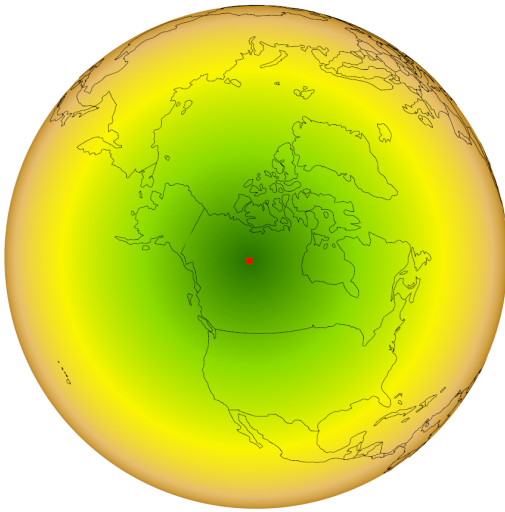
between. These two algorithms act as extremes, meaning other options are also possible. We have investigated all the combinations possible to reach from the base resolution to the target resolution. We found that always visiting every resolution in between, as the iterative algorithm does, is always the most efficient.

4.4 Comparison

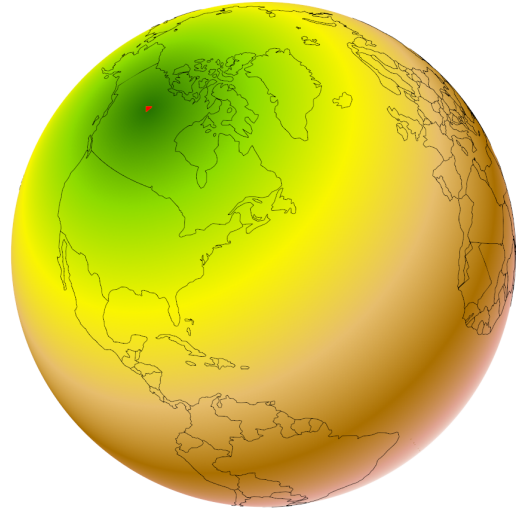
Distance transform methods used in traditional GIS that are based on image space introduce distortions. Figure 4.8a shows the distance transform computed in ArcGIS Pro software using image-based methods, and the distortions can be clearly seen. Figure 4.8b and Figure 4.8c shows a visualization calculated on the DGGS with our method.



(a)



(b)



(c)

Figure 4.8: Distance transform from Yellowknife using **(a)** ArcGIS and planar distance calculations, **(b, c)** DGGS and our method.

To quantify the amount of distortion caused by traditional GIS (i.e., flat map projections), we performed a test which is similar to Section 4.1. We use the same features as Figure 4.4,

but we switch to the **planar** distance option within the “Generate Near Table” tool. To use the planar distance calculations, we first project the boundary and the sample points to the planar domain using the Universal Transverse Mercator (UTM) coordinate system. Each zone in the UTM coordinate system is a cartesian coordinate system with meter units. Table 4.2 shows the comparison between the ArcGIS Pro **planar** distance calculation and the distances produced by Algorithm 5. As we expect, significant distance distortion is present in the planar setting and it reduces to smaller amounts as the scale decreases. In this test, if the feature was not contained in a single UTM zone, we projected all points to the zone that contains the largest portion of the feature. Specifically, we used UTM zone 17N for Ontario, 11N for Calgary, and 12N for the farm field.

Difference	Min (m)	Max (m)	Mean (m)	Std. Deviation (m)
Ontario	-85.9256	3268.89	431.938	581.674
Calgary	0.0068	32.80	6.301	6.725
Farm field	-0.0058	1.80	0.307	0.359

Table 4.2: Difference in meters between the ArcGIS Pro calculated **planar** distances and the distances from Algorithm 5.

To compare the MMP algorithm [35] for general meshes with our method, there are two aspects to consider. First, the MMP algorithm gives the exact geodesic distance *on the mesh*. In the case of a DGGS, the mesh is an approximation of a sphere, and points must be projected from the sphere onto the face of the mesh. The MMP algorithm does not consider the effects of the projection and thus its geodesic calculations will not be the same as those calculated on the sphere. However, for higher resolutions of the DGGS, where the mesh is closer to the surface of the sphere, the MMP algorithm becomes a closer approximation to the spherical geodesic distances. Second, the MMP algorithm is less efficient than our algorithm for higher resolutions. For this comparison, we have used the border of Ontario to benchmark the algorithms. The process of which is done on a computer with an Intel Core i7-6700 CPU with both algorithms

being tested under the same conditions using the Google Benchmark tool. Figure 4.9 then compares the execution time of the two algorithms, and Table 4.3 lists the number of target cells at different resolutions next to their respective execution times. In resolutions lower than 9, the MMP algorithm is faster (though it is less accurate), but the difference is negligible. As we go to higher resolutions and greater numbers of target cells, the MMP algorithm takes more time than our method by a wide margin.

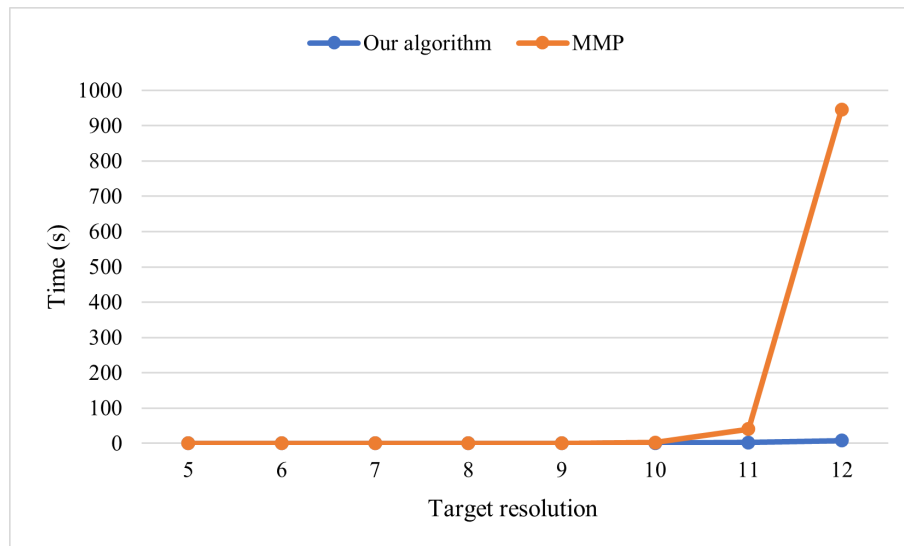


Figure 4.9: Execution time of our algorithm and the MMP algorithm at different resolutions for the border of Ontario.

Target resolution	# of Target cells	Our algorithm (ms)	MMP (ms)
5	208	74.0	0.8
6	464	66.6	1.6
7	1488	113.3	5.7
8	5952	159.6	33.9
9	19392	273.9	240.7
10	69696	719.7	2474.8
11	260480	2200.3	40178.0
12	1041920	7541.7	946192.2

Table 4.3: Execution time of our algorithm and MMP algorithm along with the number of target cells for the border of Ontario.

Chapter 5

Application of Distance transform in Soil Sampling¹

To evaluate our proposed DT algorithm in a real-world case study, we chose the problem of selecting benchmark sites in agricultural fields. As discussed in section 2.7, benchmark sampling is a superior soil sampling technique, and MZ delineation provides a basis for it. The main challenge of benchmark sampling is selecting a benchmark site within each MZ that is a proper representative of that MZ. The criteria for the representative area of the MZ depend on the method used for delineating the MZs. In general, the following criteria are important:

1. *Being close to the median of the performance function used for MZ delineation within each MZ:* Median is statistically considered a good representative of a data set because it is a robust measure describing the central tendency of the data. Therefore, it is desired to select the benchmark site in a place where its performance function value is close to the median of its MZ. Figure 5.1a shows a visualization for this criterion for only one MZ in

¹This chapter is addressing a real-world problem derived from a Mitacs project from Decisive Farming Company (part of Telus Agriculture).

which blue regions have a smaller absolute difference to the median.

2. *Being far from the boundary of the MZ*: Areas close to the MZ boundaries are more sensitive to input changes or year-to-year variation. To obtain more robust benchmark sites, it is better to find areas away from the boundaries of each MZ. Figure 5.1b shows this criterion for one MZ; locations with darker blue are farther from the boundary.
3. *Being close to the previously sampled points*: To be able to perform a benchmark sampling, if any previously sampled points are available, it is better to select the new benchmark site close to the previously sampled points.
4. *Steepness*: The nutrient levels at steep areas may vary a lot from year to year. Moreover, if an area is steep, it may not be accessible for the sampling truck. Therefore, it is desired to select benchmark sites away from steep regions.
5. *Avoiding headlands*: Headlands are areas of the farm field where heavy agricultural machines, such as combine harvesters, take turns. These areas might be affected by denser soil and overlapping application of fertilizer. Hence, these areas are not representative of the entire field and should be avoided.
6. *Avoiding inaccessible areas*: Inaccessible areas should be avoided so that sampling trucks can collect samples.
7. *Avoiding the proximity of benchmark sites in other MZs*: The selected benchmark sites in different MZs should be far away from each other to ensure the benchmark sites do not all come from a small region of the field. Figure 5.1c shows an example of benchmark sites that are concentrated in a small area versus benchmark sites that are distanced from each other.

In various scenarios, all or a subset of these criteria must be satisfied.

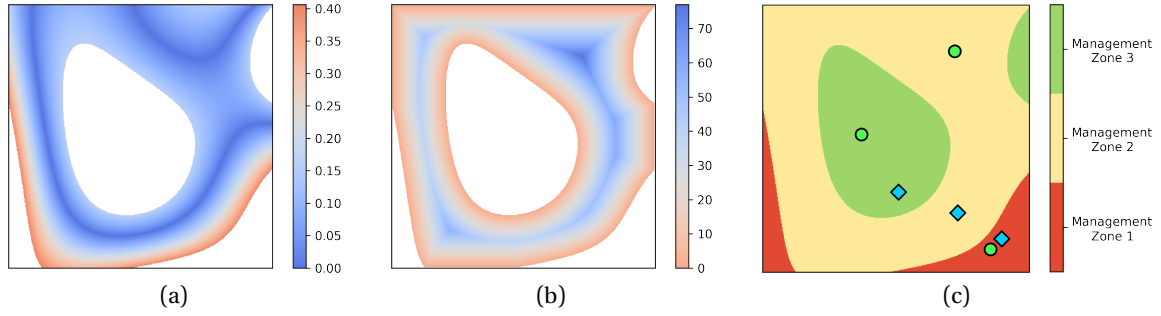


Figure 5.1: Criteria for MZ representative area. (a) absolute difference to the median value of the underlying function for MZ 2 (the numbers are in the unit of the performance function), and (b) distance to the boundaries of MZ 2 have been shown using a cool-warm colour map (in meters). (c) blue diamonds are benchmark sites concentrated in the lower right part of the field, while green circles are benchmark sites that are distanced from each other.

The process of selecting benchmark sites based on the criteria is cumbersome requiring integrating various datasets and manually comparing different values. Therefore, the challenge is how to automatically satisfy all criteria and constraints using the available datasets. We propose a framework that offers an automated and algorithmic approach that integrates various data inputs such as MZs, Digital Elevation Model, and field boundary into a multi-objective and multi-constraint optimization model. By solving this optimization model, the framework generates benchmark sites that meet specific needs. In this framework, for each MZ, several distance metrics are calculated and combined into a single weighted error function. By minimizing this error function, the optimal area for placing benchmark sites is determined. In cases where MZs are not available, any given performance function, typically time-varying remote sensing measurements, can be used instead. MZs can then be determined using a flexible thresholding technique.

In this case study, DGGS helps us to support integrating different inputs, calculating various distances (e.g., to the MZ boundaries) efficiently, solving the optimization problem, and automatically repeating this process for various regions. Recently, DGGS has been selected as

an effective GIS platform used for data integration and analysis of remote sensing data [12] and congruent geography applications [10], which provides more confidence in selecting DGGS for this task.

This case study proposes an algorithmic framework that effectively automates the process of selecting benchmark sites for a field, based on its georeferenced shape file and remote sensing measurements in time. The framework is highly adaptable and can easily incorporate different measurements and site selection criteria. Compared to the conventional GIS process, which typically takes several days with the involvement of GIS technicians, our automatic process of generating benchmark sites using the proposed algorithmic framework takes only a few seconds. We also provide a comprehensive evaluation and analysis of the resulting benchmark sites to confirm that our optimization model works as expected. Furthermore, we demonstrate how our framework can be leveraged to incorporate different criteria that are not considered in this work, making it a highly adaptable tool for selecting benchmark sites for fields.

5.1 Benchmark Site Selection Framework

The criteria for a representative area of a field is formulated into a framework as a constrained multi-objective optimization problem. Criteria that specify areas to avoid are our constraints and criteria that we want to optimize for are objectives. The main input data, which are performance function (usually comes from satellite imagery or sensor measurement) and DEM, come in discrete space; hence, objective functions are naturally calculated in discrete space too. Consequently, we need a common discretization of space to store input data, calculate the objective functions, and solve the optimization problem. We exploit the multi-resolution property of a DGGS to choose a resolution as a discrete space.

Our automated process of selecting benchmark sites for a field is presented in Figure 5.2 in six steps. The process starts by loading the boundary of the field in step 1. In step 2, the

performance function and DEM are loaded either by downloading them or loading them from a local cache. Step 3, constructs the MZs from the performance function for step 4. The main contribution of this case study is in steps 4 and 5. In step 4, first, MZs, performance function, and DEM data are resampled into DGGS cells at the target resolution, which means for each DGGS cell we know what the MZ, performance function value, and elevation are. This data integration using a DGGS allows additional datasets such as different satellite imagery (e.g., thermal imagery or radar data), aerial imagery, and wetland maps, to be integrated with the algorithm in the future as needed. Next, we compute all of the objectives and constraints and store them back in the DGGS cells. All of the objectives are computed on the centroid of DGGS cells. In step 5, the objectives are normalized and combined in a single error function to solve the optimization problem. A complete search in DGGS cells finds the optimum benchmark site. Lastly, in step 6, the selected benchmark sites are saved in a GeoJSON² format.

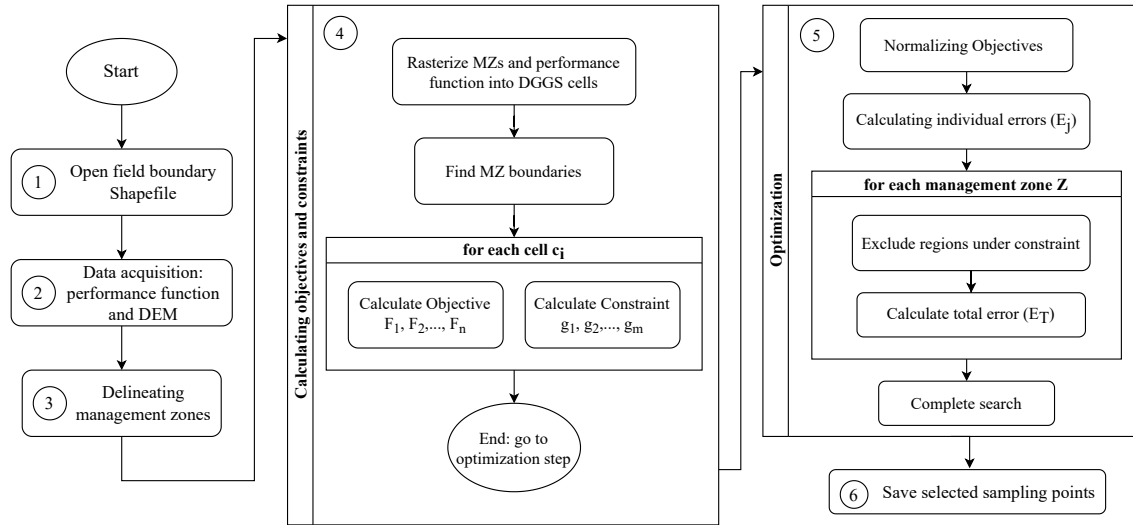


Figure 5.2: The flowchart of selecting benchmark sites for a new field.

In section 5.2, we explain the specific MZ delineation method used in our framework (step

²GeoJSON is a format designed for representing geographical features based on the JSON (JavaScript Object Notation) format.

3 of flowchart) and in section 5.3 we define a set of objectives and constraints and explain how our framework is built to satisfy the objectives and constraints (step 4 and 5 of flowchart).

5.2 From Data to Management Zone

The performance function is the key to delineating the field to MZs. There are several choices for performance function. For example, Georgi et al. used the average historical satellite imagery as a performance function [47]. We use a similar method to construct the performance function from the Normalized Difference Vegetation Index (NDVI) as an indicator for plant health [30]. Moreover, a recent study shows that both soil electrical conductivity and NDVI are correlated to soil nutrient [66]. This performance function is called *Fertility Index*. The fertility index serves as a proxy measure to approximate the fertility of the soil by monitoring the historical growth of plants. Higher fertility index values indicate better growing conditions (i.e., soil fertility), as plants have historically grown well in those areas. Similarly, in the final delineated MZs, lower MZs are regions of the field that perform poorly while higher MZs correspond to better-performing regions. This delineation process uses satellite imagery data in three following steps:

1. *Selecting images*: One image as a good representative for each growing season is selected from the recent years (e.g. 3 to 6 years). The criteria for selecting this image are that it should be cloud and haze free and be near the time of the harvest to better show the potential of the soil. This image represents the variation in soil fertility through the visual growth of plants. Figure 5.3 presents an example of the selected images for a given field. Note that, although these images are displayed in RGB colours, red and near-infrared (NIR) bands are used in calculating performance function.
2. *Calculating Performance function - Fertility Index (F-Index)*: Normalized Difference

Vegetation Index (NDVI) is calculated for each of the selected images using red and NIR bands which is an indicator for the health of plants [67]. While the range of the NDVI is between $[-1, 1]$, for the selected images the range of values is usually around $[0.2, 0.95]$ with the mean around 0.8. Then, the NDVI values of each image are normalized and scaled in a way that the mean value M is a fixed number in the range $[0, 255]$ (e.g., $M = 100$). Finally, the normalized NDVI values are averaged and the resulting averaged normalized values are referred to as *F-Index* (see Figure 5.4).

3. *Thresholding*: To finally delineate the F-Index into MZs, the next step is to divide the entire range of F-Index into a certain number of bins B (e.g. $B=6$). The F-Index thresholds are selected in a way that the area of the MZs forms a normal distribution. The resulting map after thresholding is called *F-Map* (see Figure 5.4).

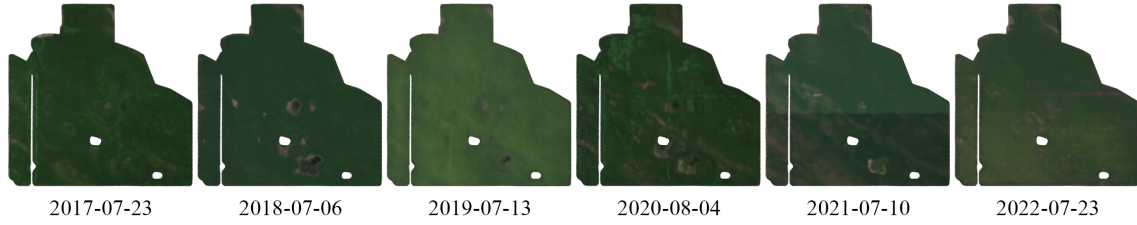


Figure 5.3: An example of selected satellite images for delineation process.

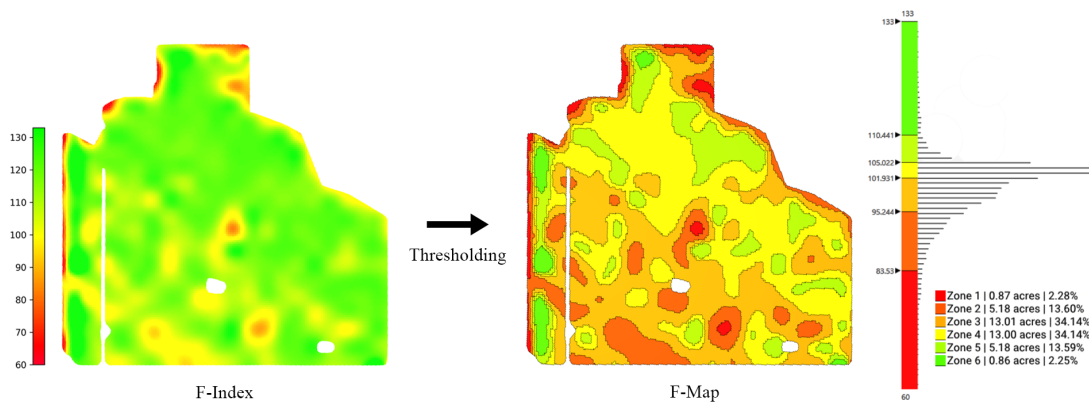


Figure 5.4: An example of an F-index, F-Map, the histogram of F-Index, and the associated thresholds.

Figure 5.4 presents an illustrative example of an F-Index and F-Map along with a histogram of the F-Index and the thresholds used in the process. The histogram of F-Index (see Figure 5.4 right) depicts the distribution of the performance function across the field. Upon examining the histogram, we observe that roughly 70% of this field has F-Index values between 95 and 105. As a result, dividing the range of the F-Index values into equal-length bins would yield MZs with a very small area. Likewise, If the field is divided into equal-area MZs, some MZs will have a substantial variation in F-Index while some other MZs will have a very narrow range of values. To address this issue, the threshold values are selected in a way that the areas of MZs follow a normal distribution.

5.3 Optimization Model

Once we have evaluated MZs (i.e. F-Map), the next question is how to identify the optimal benchmark sites within each MZ. As discussed at the beginning of this chapter, there are several objectives and some constraints that a site must meet to be considered a good representative point of an MZ. We chose the following criteria and constraints in building our framework:

Objective F_1 , close to the value of median F-Index: The F-Index of the benchmark site should be close to the median F-index of the MZ.

Objective F_2 , away from MZ boundaries: The benchmark site should be located away from MZ boundaries.

Objective F_3 , close to previously sampled points: The benchmark site should be situated close to previously sampled points (if available).

Objective F_4 , belong to flatter regions: The benchmark site should be on a flat area of the field.

Constraint g_1 , avoid certain regions: In practice, it is desired to avoid locating the benchmark sites in certain regions (e.g. inaccessible and unrepresentative areas). Normally, the benchmark site should be at least D_H meters (e.g., 30 meters) away from any headlands (headland condition).

Constraint g_2 , avoid the proximity of benchmark sites in other MZs: The benchmark site should not be in close proximity to other selected benchmark sites from the other MZ to ensure getting samples from a larger region.

We formulate these requirements as a constrained multi-objective optimization problem. The objective functions F_1 to F_4 must be optimized together in consideration with constraints g_1 and g_2 . The objectives are distance-based and we aim to minimize all of them as best as possible. The two constraints are binary concepts that must be adhered to within the feasible space of site locations. However, some of the objectives can also be considered as constraints. For example, we can maximize the distance to the MZ boundaries as an objective (F_2), or we can set a constraint that the benchmark site must be at least n meters away from any MZ boundary. Although “being away from the boundary with a certain distance” is sufficient, to

keep our optimization model flexible and avoid adding another parameter, we consider this property as a distance-based objective. The same thing can be said about the steepness and distance to previously sampled points. If we set thresholds, we transform our objectives into constraints that exclude any space that does not meet the threshold. However, it's important to note that some of these spaces may still be valuable in terms of other objectives and finding good thresholds to not end up with empty search spaces is difficult.

Computing the Objective Functions

The first step towards solving the optimization problem is calculating the values of objective functions. In this section, we discuss how each of the objective functions F_1 to F_4 are calculated. The resolution of satellite data (and F-Map) is 10 m (10 m x 10 m square pixels), while the DEM data comes in 12 m resolution. Thus, we use resolution 19 of disdyakis triacontahedron DGGS [31], as the target resolution in which DGGS cells (Avg. cell size = 61.8 m^2) are a bit smaller than the input data sources. Then, the F-Index, F-Map, and DEM data are resampled into DGGS cells at the target resolution. Next, we compute all of the objectives and store them back in the DGGS cells.

Close to the Value of Median F-Index (F_1)

The first objective is to minimize the distance between the F-Index values of the MZ cells and M_Z , the median of the management zone Z . Let's denote the DGGS cells of Z by c_1, c_2, \dots, c_n and their respective F-Index values by f_1, f_2, \dots, f_n . This objective is defined as:

$$F_1(c_i) = |f_i - M_Z|.$$

Figure 5.5 shows a visualization of this objective function. The blue colours in this visualization show the less absolute difference between each cell's F-Index and the median F-Index.

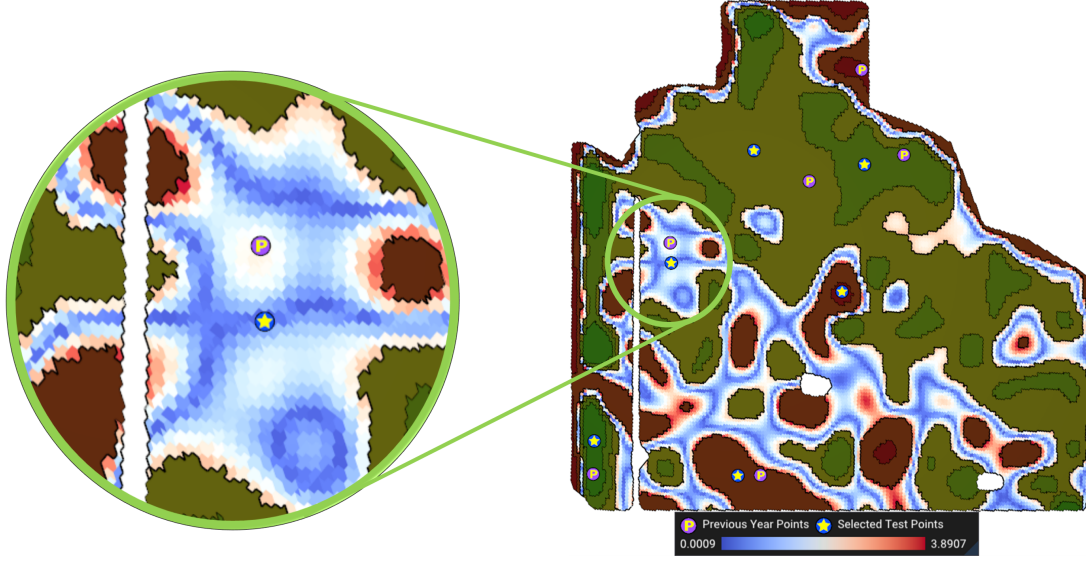


Figure 5.5: Visualization of absolute difference to median F-index for MZ 3.

Away from MZ Boundaries (F_2)

We use the distance transform operator on top of the DGGS to compute the geodesic distance of each cell c_i to S , the spatial boundary of management zone Z . The distance transform efficiently calculates the geodesic distance of all cells in a region to a given vector feature (i.e., the MZ boundaries). The second objective F_2 is to maximize this distance

$$F_2(c_i) = D_S(c_i)$$

where D_S denotes the geodesic distance of c_i to S . Figure 5.6 shows a visualization of this objective function. The cooler the colours (darker blue) the farther the cell is from the boundaries.

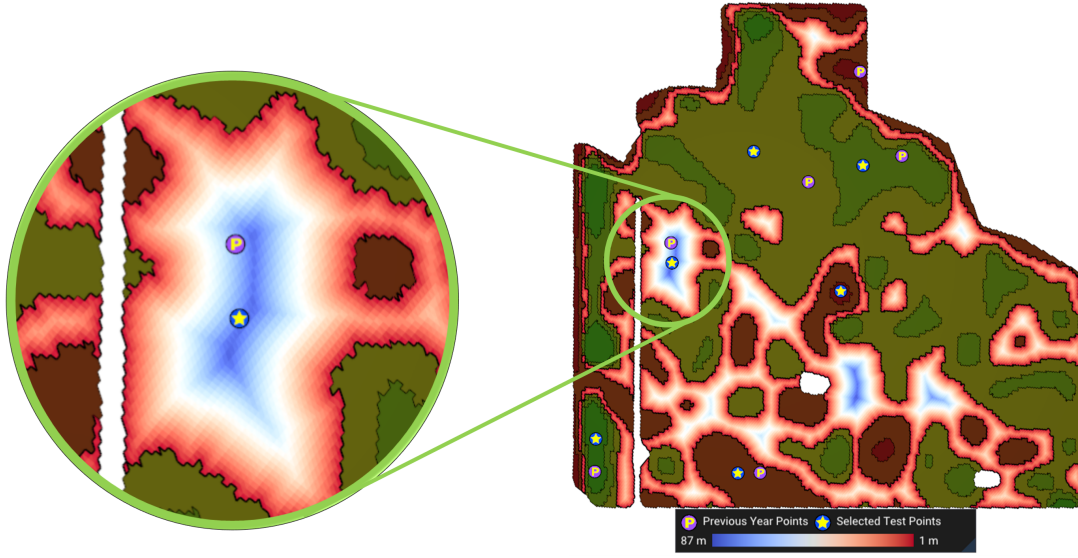


Figure 5.6: Visualization of distance to boundaries of MZ 3. The farthest cell from the boundary is 87 m away from the boundaries.

Close to Previously Sampled Points (F_3)

Let P_z represent the sampled point of management zone Z from previous years. Again using distance transform, we determine the geodesic distance from the previously sampled point P_z to the centroid of all cells within Z . The third objective F_3 is to minimize this distance

$$F_3(c_i) = D_{P_z}(c_i)$$

where D_{P_z} denotes the geodesic distance of c_i to previously sampled point P_z . Figure 5.7 shows a visualization of this objective function. The cooler the colours (darker blue) the closer the cell is to the previously sampled point.

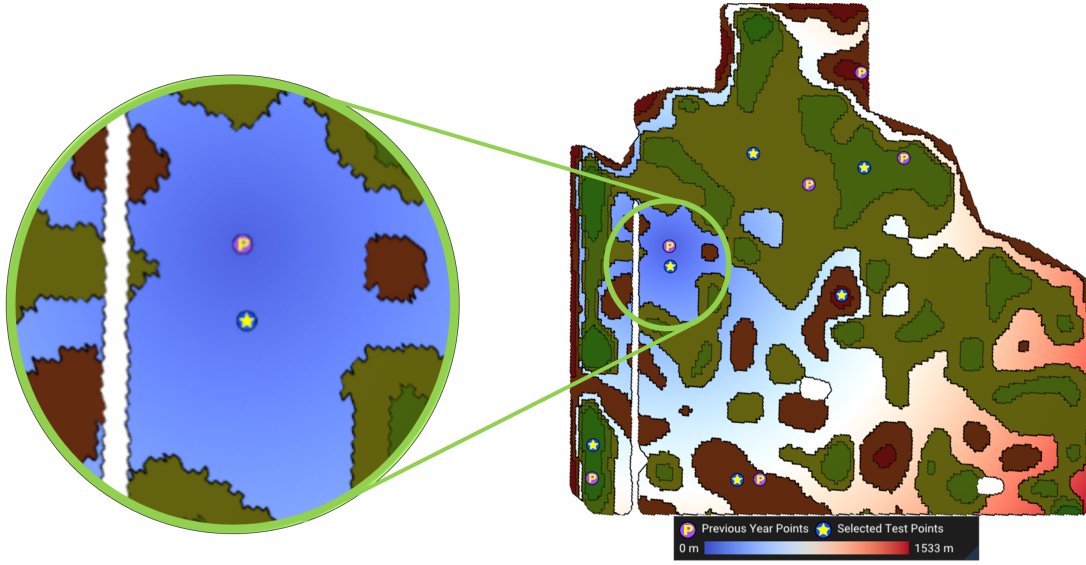


Figure 5.7: Visualization of distance to previously sampled point for MZ 3. The farthest cell from this previously sampled point is 1533 m away from it.

Belong to Flatter Regions (F_4)

The steepness of the cell c_i (denoted by $S(c_i)$) is calculated from the DEM of the field by calculating the gradient vector. The gradient vector $\vec{G}(c_i)$ shows the direction of change of elevation, which is approximated using the difference in elevation between neighbouring cells. Steepness is then determined by:

$$F_4(c_i) = S(c_i) = \frac{\pi}{2} - \arccos(\vec{G}(c_i) \cdot \vec{N}(c_i))$$

where $\vec{N}(c_i)$ is the normal vector of the cell c_i . Figure 5.8 shows a visualization of this objective function. The cooler the colours (darker blue) the cell is in a flatter region.

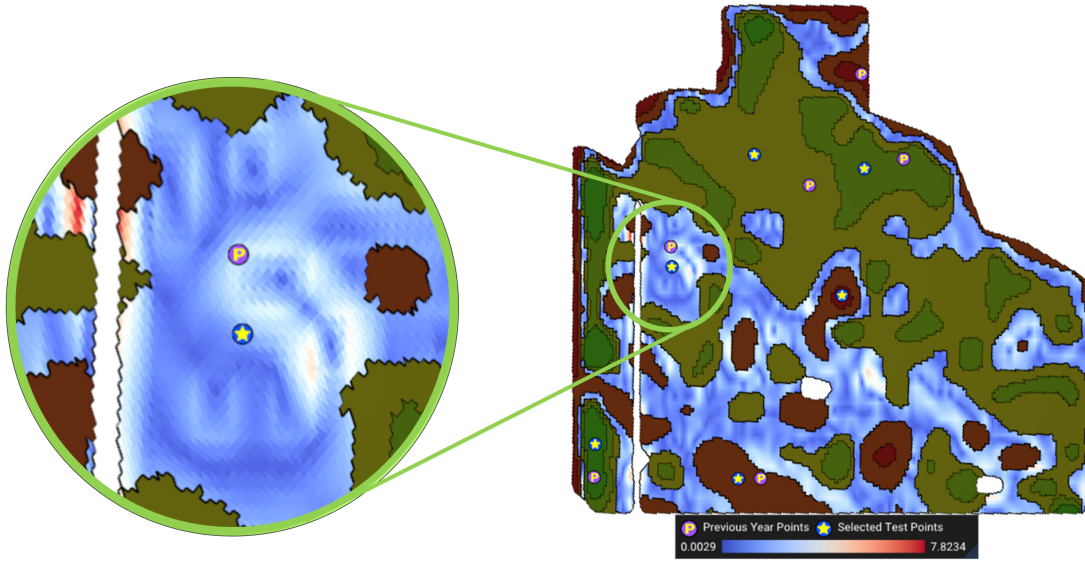


Figure 5.8: Visualization of the steepness for MZ 3. The legend shows the steepness in degrees. The steepest point of this MZ is 7.8 degrees steep.

Computing the Constraints

In this section, we discuss how to determine the feasible space of the optimization by calculating constraints for each cell inside the field.

Avoid Certain Regions (g_1)

We trivially exclude any cells from the inaccessible and unrepresentative regions by subtracting these regions from the entire field. For the headland condition, we use the distance transform operation of DGGS to calculate a buffer of D_H meters from the boundary of the field to avoid the areas under the headland. Figure 5.9 shows the areas of the farm avoided due to headland.

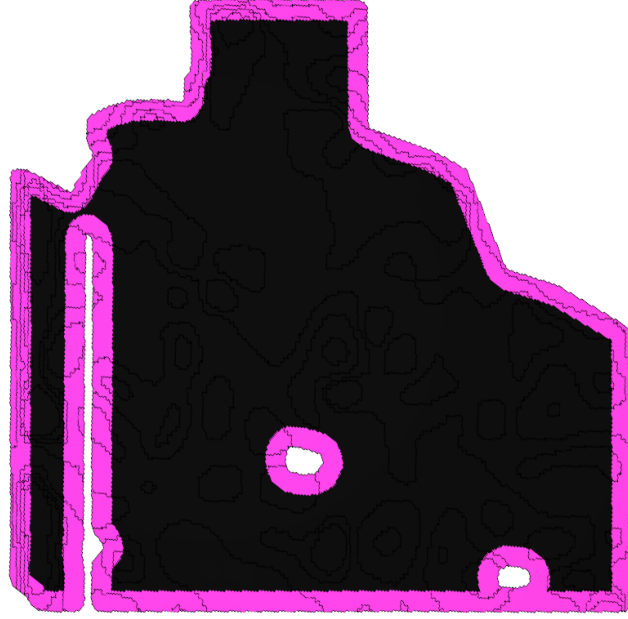


Figure 5.9: The extracted headland of the farm field ($D_H = 50m$).

Avoid the Proximity of Benchmark Sites in Other MZs (g_2)

Our goal is to select benchmark sites that are from a larger region of the field to better represent the entire field. To achieve this, we set a minimum radius D_R between the sites (see Figure 5.10). This global constraint is unique compared to other criteria that are local to their respective MZs. We begin by selecting a benchmark site in one MZ and then limiting the areas in other MZs that are within the specified radius of this site. We continue this process iteratively until we have chosen benchmark sites for all MZs. To do this, we remove c_i from the search space if $D_t(c_i) \leq D_R$ for all already selected sites t (see Figure 5.10). We easily use the distance transform of DGGS to calculate the distance of all points in the farm to already selected benchmark sites in each step.

By using this method, the benchmark sites selected earlier have an advantage over the ones chosen later, as the latter are subject to more constraints. If D_R is small, the change of order

has a minimal impact. However, for larger D_R , it makes sense to prioritize MZs according to their level of importance. Therefore, we start with the most important MZ in order to find a more optimal benchmark site for it, and then we continue to select benchmark sites for less important MZs. The most important MZ is the one that best represents the entire field. Hence, the most important MZ is the one with its median F-Index closest to that of the entire field. We sort the MZs based on the distance of their median F-Index to the median F-Index of the entire field. The radius mentioned above can be dynamically changed, but the default is set to be 15% of the field's diameter. Figure 5.10 shows how we use this radius to force the selected benchmark sites to be at a reasonable distance from each other.

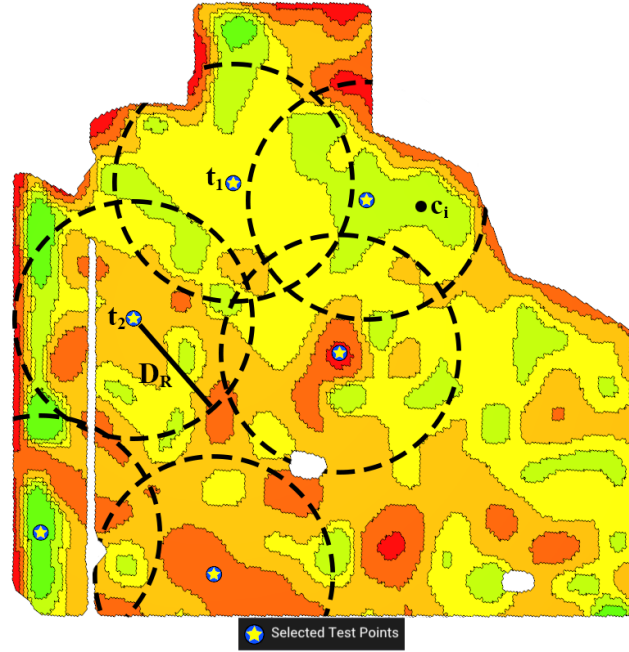


Figure 5.10: The forced distancing between selected benchmark sites. t_1 and t_2 are already selected benchmark sites. When deciding for point c_i in the next MZ, we only check the distance to t_1 and t_2 .

Solving the Optimization Problem

With all objective functions and constraints ready, we need to optimize them together. To satisfy constraint g_2 , we solve the optimization for each MZ separately in the order of importance discussed in Section 5.3. For each MZ, to accommodate the constraints, we remove DGGs cells under constraint from our feasible search space. This not only ensures that no point under constraint will ever be selected as a benchmark site but also makes the optimization more efficient by reducing the search space. Next, to solve the multi-objective optimization problem we use a scalarization method which is a common method that transforms a multi-objective optimization problem into a single-objective optimization problem [62]. Because we have calculated all of our objectives and constraints in the discrete space of a DGGs (resolution 19 of disdyakis triacontahedron DGGs. Avg. cell size = 61.8 m^2 [31]), for a 200-hectare field, the total unconstrained search space will have roughly 32,400 cells. This is a very small search space and modern personal computers can evaluate the objective functions for the entire space within a fraction of a second. Hence, to efficiently find the global minimum of the objective function, we perform an efficient complete search on the feasible search space (i.e. comparing objective functions on c_1, c_2, \dots, c_n).

To transform our multi-objective optimization problem into a single-objective optimization with a scalarization method, we use a *weighted squared sum* method. A weighted squared sum enables us to control the effect of each objective function relative to each other and also to penalize large errors. With this, we combine all of the objectives into a global objective function or an *Error Function*. Before combining, we first normalize all the individual objective functions. Without normalization, objectives with large values may overpower and dominate the optimization process. To map objective function F_j into the range $[0, 1]$, we use a simple linear mapping:

$$\tilde{F}_j = \frac{F_j - \min(F_j)}{\max(F_j) - \min(F_j)}.$$

Next, we define an error function E_j for each of the objectives as follows:

$$E_j = \begin{cases} \tilde{F}_j & \text{if objective needs to be minimized (e.g. distance to median F-Index)} \\ 1 - \tilde{F}_j & \text{if objective needs to be maximized (e.g. distance to MZ boundaries).} \end{cases}$$

Now we have a vector of error functions. In order to scalarize these errors into a single error function, we combine them using weighted squares sum:

$$E_T(c_i) = \sum_{j=1}^m w_j E_j^2(c_i)$$

where w_j is the weight of the objective F_j , and m is the number of objectives ($m = 4$ in our case). Figure 5.11 shows the visualization of the error function and the yellow star is the location of the minimum of this function. Note that for illustrative purposes, the constraints are not removed from the search space in this figure.

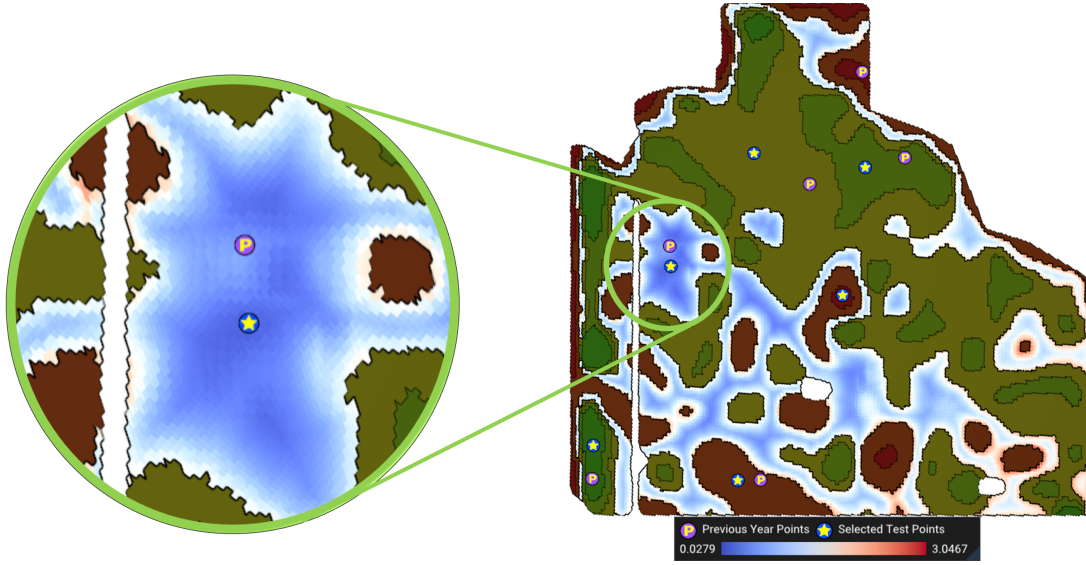


Figure 5.11: Visualization of the error function for MZ 3. All weights are set to 1 ($w_j = 1$, $j = 1, 2, 3, 4$) in this figure. E_T at the optimal point (denoted by the star) is 0.0279.

Figure 5.12 shows the final error function only for feasible search space considering constraints for all MZs. The order of MZ optimization in this figure is MZ 4, MZ 3, MZ 5, MZ 2, MZ 6, and lastly MZ 1. The areas under the headland constraint (g_1) and distribution constraint (g_2) are marked with pink.

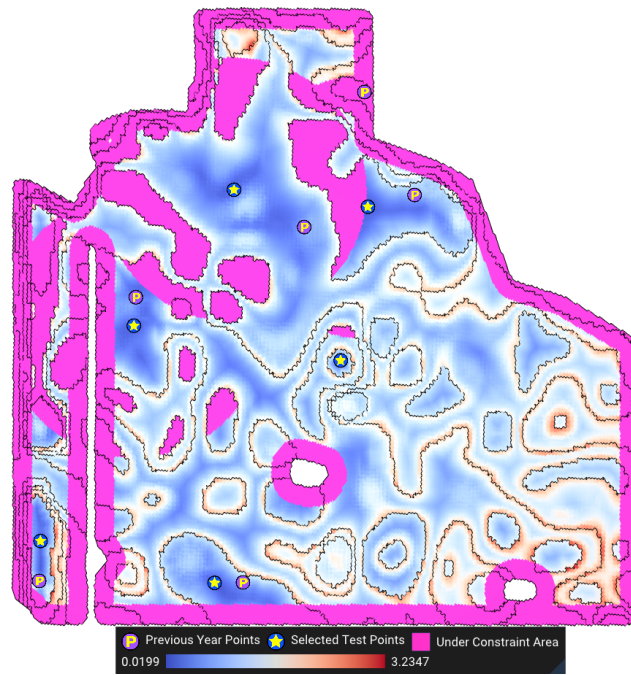


Figure 5.12: Visualization of the final error function for the entire field. All weights are set to 1 in this figure.

Chapter 6

Use Case Results and Evaluation

The process of selecting benchmark sites within a field takes multiple days for a GIS technician to complete. The primary objective of this work is to provide a framework to automate this process by avoiding typical manual GIS processes like integrating different layers of information and manually comparing the values of objectives for selected points. The method is implemented in C++ programming language and uses OpenGL for rendering the graphical interface. The application provides two interfaces: a Graphical User Interface (GUI) for illustration and debugging purposes and a Command Line Interface (CLI) to fully automate the process. The CLI version only takes a few seconds for a relatively large field (see Table 6.1) and only needs the boundary of the field as input with a few additional settings. To demonstrate the repeatability of this automatic approach, and to show that our algorithm works for different fields, we chose 5 different fields of varying sizes across the Canadian provinces of Alberta, Saskatchewan, and Manitoba. The list of the fields and the execution time of our algorithm is given in Table 6.1.

Table 6.1: The list of fields used for evaluation. Total execution time captures the entire process including loading data from the local cache (tested on a computer with an Intel Core i7-6700 CPU).






Field number	Field shape	Location	Area (hectare)	Number of MZs	D_R (meter)	Total execution time (sec-ond)	Steps 4 and 5 time (sec-ond)
1		Southern Alberta	198.2	6	350	22.91	13.82
2		Southern Saskatchewan	102.3	6	269	13.76	7.37
3		Southern Saskatchewan	59.1	5	218	8.22	3.20
4		Southern Manitoba	43.9	5	169	7.88	3.16
5		Western Alberta	36.5	4	141	8.87	3.13

Figure 6.1 shows the selected benchmark sites for each field. To generate these results, six years of satellite imagery is used for MZ delineation, the headland (D_H) is set to 30 m, the D_R for each field is given in Table 6.1, and the objective weights w_j are all set to 1. The effect of different weights on optimization is discussed later in the next subsection. To demonstrate the correctness of the optimization, we calculated the statistics of the objectives for each MZ for all fields. The statistics for field 1 are reported in Table 6.2 and the same statistics for other fields are presented in the appendix. Each row of these tables includes the mean and the range of each objective across the MZ along with the best value for that objective, the objective value at the selected benchmark site, and the percent improvement relative to the

$$\text{mean} \left(\frac{\text{benchmark site} - \text{mean}}{\text{mean}} \times 100 \right).$$

The alternative to benchmark sampling in the MZs is MZ composite sampling. If composite sampling is done properly, the objectives of the composite sample should be close to the mean of the objectives of the MZ. Hence, by reporting these statistics in tables 6.2 to 6.6 we show that the objectives of the selected benchmark sites are improved in comparison to the composite sampling. This enables the farm manager to track the nutrient changes. In all the fields we evaluated, the objective for the selected benchmark site is always better than the mean of all the cells of the MZ except for a couple of cases in fields 1 and 3 that are slightly worse. In both these cases, the problematic benchmark sites are in the first and last MZs which represent the smallest area of the field (see Figure 5.4). Selecting good benchmark sites for the middle MZs—which represent the majority of the area of the field—at the cost of slightly worse than mean benchmark sites for the first and last zones is reasonable. The presented statistics show that in the middle MZs (i.e. MZ 3 and 4 out of the six total), the objectives at the benchmark sites are significantly better than the mean.

Overall, our algorithm produces better benchmark sites than a composite sample according to our criteria. Note that the Objective F_3 (being close to previously sampled points) is excluded from our objectives because we did not have previously sampled points for any of the fields. Moreover, we excluded cells under constraint from our mean and range calculations to get a fair comparison. Lastly, our optimization constraints g_1 and g_2 are satisfied; no benchmark sites are selected in headland (D_H) or in proximity (D_R) to other benchmark sites. Constraint g_2 , in combination with objective F_2 , will result in a relatively good covering of the field by benchmark sites which can be visually verified by looking at Figure 6.1.

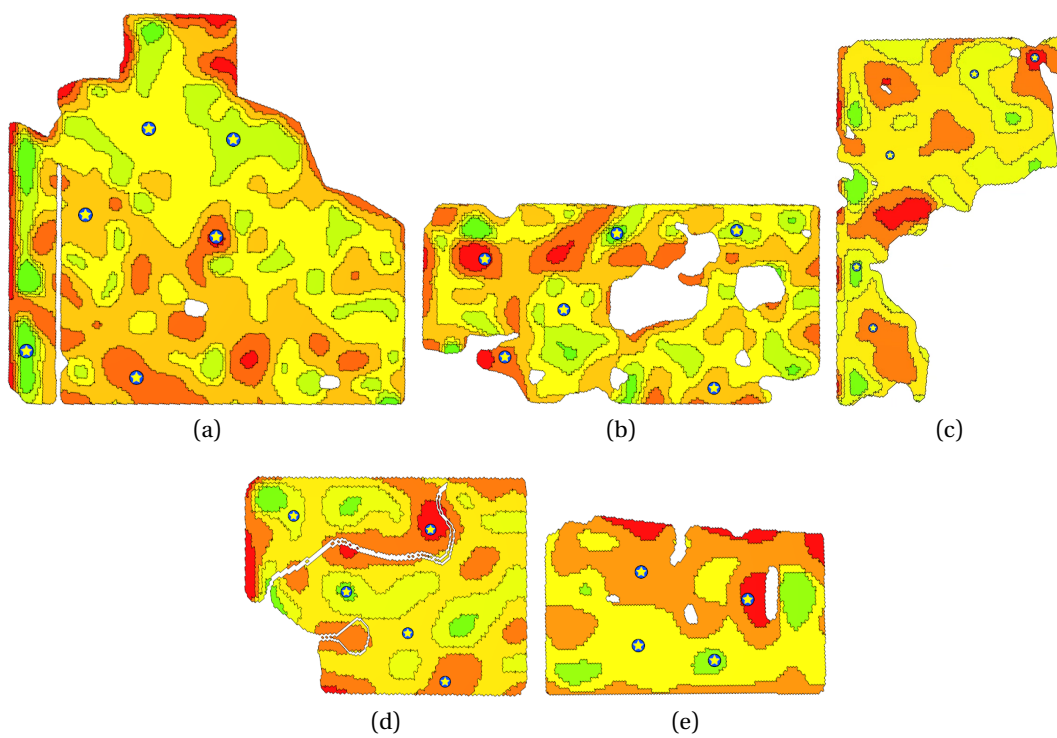


Figure 6.1: Selected benchmark sites for the field (a) 1, (b) 2, (c) 3, (d) 4, and (e) 5.

Table 6.2: Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 1.

MZ	Objective	Benchmark site	Mean of all cells	Percent improvement	Best	Range
1	Median F-Index	1.359	1.966	30.9%	0.126	[0.126, 3.563]
	Distance To Boundary Steepness	23.910	7.916	202.1%	24.560	[1.238, 24.560]
		0.085	0.725	88.3%	0.007	[0.007, 1.826]
2	Median F-Index	1.125	2.342	52.0%	0.001	[0.001, 8.562]
	Distance To Boundary Steepness	63.797	17.337	268.0%	77.104	[1.238, 77.104]
		0.358	1.234	71.0%	0.015	[0.015, 5.319]
3	Median F-Index	0.037	1.586	97.7%	0.001	[0.001, 3.891]
	Distance To Boundary Steepness	80.167	19.285	315.7%	87.775	[1.238, 87.775]
		1.126	1.230	8.5%	0.010	[0.010, 5.179]
4	Median F-Index	0.045	0.707	93.6%	0.000	[0.000, 1.715]
	Distance To Boundary Steepness	125.790	24.736	408.5%	141.578	[1.238, 141.578]
		0.943	0.953	1.0%	0.002	[0.002, 4.336]
5	Median F-Index	0.038	0.899	95.8%	0.001	[0.001, 4.604]
	Distance To Boundary Steepness	81.755	14.912	448.3%	83.902	[1.238, 83.902]
		0.934	0.994	6.1%	0.004	[0.004, 4.322]
6	Median F-Index	4.046	3.351	-20.7%	0.000	[0.000, 18.670]
	Distance To Boundary Steepness	29.766	12.167	144.6%	43.304	[1.238, 43.304]
		0.797	1.123	29.1%	0.015	[0.015, 3.804]

Table 6.3: Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 2.

MZ	Objective	Benchmark site	Mean of all cells	Percent improvement	Best	Range
1	Median F-Index	13.393	18.202	26.4%	0.000	[0.000, 47.910]
	Distance To Boundary Steepness	27.535	16.483	67.0%	48.759	[1.263, 48.759]
		2.207	2.843	22.4%	1.395	[1.395, 4.932]
2	Median F-Index	2.920	4.453	34.4%	0.017	[0.017, 18.161]
	Distance To Boundary Steepness	40.850	9.775	317.9%	43.235	[1.263, 43.235]
		1.010	3.405	70.3%	0.021	[0.021, 10.844]
3	Median F-Index	0.013	1.550	99.2%	0.000	[0.000, 4.431]
	Distance To Boundary Steepness	56.517	12.870	339.1%	56.517	[1.259, 56.517]
		1.046	2.700	61.3%	0.006	[0.006, 11.680]
4	Median F-Index	0.140	1.075	87.0%	0.000	[0.000, 2.610]
	Distance To Boundary Steepness	55.644	12.011	363.3%	59.418	[1.259, 59.418]
		1.173	2.040	42.5%	0.003	[0.003, 9.167]
5	Median F-Index	0.720	1.000	28.0%	0.002	[0.002, 2.917]
	Distance To Boundary Steepness	29.709	10.057	195.4%	41.810	[1.259, 41.810]
		1.863	2.226	16.3%	0.013	[0.013, 10.052]
6	Median F-Index	3.272	3.351	2.4%	0.000	[0.000, 13.995]
	Distance To Boundary Steepness	23.627	9.387	151.7%	33.618	[1.259, 33.618]
		2.146	3.095	30.7%	0.108	[0.108, 8.438]

Table 6.4: Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 3.

MZ	Objective	Benchmark site	Mean of all cells	Percent improvement	Best	Range
1	Median F-Index	4.590	2.883	-59.2%	0.126	[0.126, 8.487]
	Distance To Boundary	23.519	11.948	96.8%	27.191	[1.298, 27.191]
	Steepness	2.163	3.028	28.6%	0.260	[0.260, 6.357]
2	Median F-Index	0.681	2.145	68.3%	0.000	[0.000, 6.899]
	Distance To Boundary	44.390	18.374	141.6%	60.355	[1.291, 60.355]
	Steepness	0.868	2.521	65.6%	0.031	[0.031, 8.747]
3	Median F-Index	0.268	1.939	86.2%	0.000	[0.000, 4.174]
	Distance To Boundary	86.394	18.537	366.1%	91.327	[1.291, 91.327]
	Steepness	1.759	2.137	17.7%	0.026	[0.026, 9.368]
4	Median F-Index	0.344	1.834	81.2%	0.000	[0.000, 5.624]
	Distance To Boundary	48.860	13.752	255.3%	53.131	[1.291, 53.131]
	Steepness	1.863	2.167	14.0%	0.002	[0.002, 9.731]
5	Median F-Index	0.144	1.805	92.0%	0.022	[0.022, 3.604]
	Distance To Boundary	17.555	6.031	191.1%	18.800	[1.295, 18.800]
	Steepness	1.158	2.205	47.5%	0.849	[0.849, 4.675]

Table 6.5: Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 4.

MZ	Objective	Benchmark site	Mean of all cells	Percent improvement	Best	Range
1	Median F-Index	1.450	1.861	22.1%	0.000	[0.000, 4.030]
	Distance To Boundary Steepness	24.923	11.987	107.9%	34.016	[1.305, 34.016]
		0.917	1.132	19.0%	0.115	[0.115, 2.264]
2	Median F-Index	1.341	1.829	26.7%	0.019	[0.019, 6.067]
	Distance To Boundary Steepness	34.657	11.745	195.1%	53.336	[1.302, 53.336]
		0.163	1.116	85.4%	0.023	[0.023, 3.109]
3	Median F-Index	0.149	1.326	88.8%	0.000	[0.000, 3.028]
	Distance To Boundary Steepness	44.455	12.907	244.4%	49.212	[1.298, 49.212]
		0.280	1.125	75.1%	0.018	[0.018, 3.517]
4	Median F-Index	0.003	1.398	99.8%	0.003	[0.003, 4.184]
	Distance To Boundary Steepness	30.103	9.557	215.0%	34.657	[1.302, 34.657]
		0.737	1.093	32.6%	0.014	[0.014, 3.374]
5	Median F-Index	0.034	0.626	94.6%	0.034	[0.034, 1.158]
	Distance To Boundary Steepness	15.860	5.758	175.5%	18.325	[1.302, 18.325]
		0.318	0.899	64.7%	0.043	[0.043, 2.574]

Table 6.6: Range and the average of the objectives compared to the objectives of selected benchmark sites for Field 5.

MZ	Objective	Benchmark site	Mean of all cells	Percent improvement	Best	Range
1	Median F-Index	2.340	5.078	53.9%	0.294	[0.294, 14.785]
	Distance To Boundary	18.382	10.907	68.5%	32.534	[1.216, 32.534]
	Steepness	0.451	0.665	32.3%	0.029	[0.029, 1.712]
2	Median F-Index	2.361	2.469	4.4%	0.000	[0.000, 10.209]
	Distance To Boundary	85.619	22.106	287.3%	87.502	[1.216, 87.502]
	Steepness	0.510	0.934	45.4%	0.010	[0.010, 4.003]
3	Median F-Index	0.168	1.441	88.4%	0.000	[0.000, 3.680]
	Distance To Boundary	105.288	26.053	304.1%	109.590	[1.212, 109.590]
	Steepness	0.347	0.857	59.5%	0.004	[0.004, 3.446]
4	Median F-Index	0.101	3.439	97.1%	0.000	[0.000, 16.610]
	Distance To Boundary	27.363	13.264	106.3%	44.053	[1.216, 44.053]
	Steepness	0.861	1.045	17.7%	0.050	[0.050, 3.369]

To comprehensively summarize these tables, for each field we are reporting the mean percent improvement over all MZs of the field. The mean percent improvement is reported in Table 6.7 for each objective. This aggregated result emphasizes how the representative site objectives are improved in comparison to composite sampling.

Table 6.7: The mean percent improvement of the selected benchmark sites.

Field number	Median F-index (F_1)	Distance to boundary (F_2)	Steepness (F_4)
1	58.2%	297.9%	34.0%
2	46.2%	239.1%	40.6%
3	53.7%	210.2%	34.7%
4	66.4%	187.6%	55.4%
5	60.9%	191.6%	38.7%

6.1 Modularity and Extendability

One important flexibility of our framework is modularity and extendability. Although we chose a specific performance function, a specific MZ delineation method, and a specific set of criteria to demonstrate our optimization model, our algorithmic framework is flexible and defines a mathematical structure to be used in other scenarios as well.

Our framework gets an MZ map as input which can be constructed from any performance function. Apparent soil electrical conductivity (EC) is a particular performance function that is proven to highly correlate with soil nutrients and is useful for MZ delineation [66, 68]. To demonstrate that our framework works for other performance functions as well, we used a real EC dataset for a field in Selbitz (Elbe), Germany [69]. For simplicity, we used the same thresholding method to delineate MZs from EC data. The left column in Figure 6.2 shows the EC data and the F-Index for this field in Germany and the right column is the selected benchmark

sites along with the delineated MZs. This figure demonstrates that our framework is general and works for an arbitrary performance function.

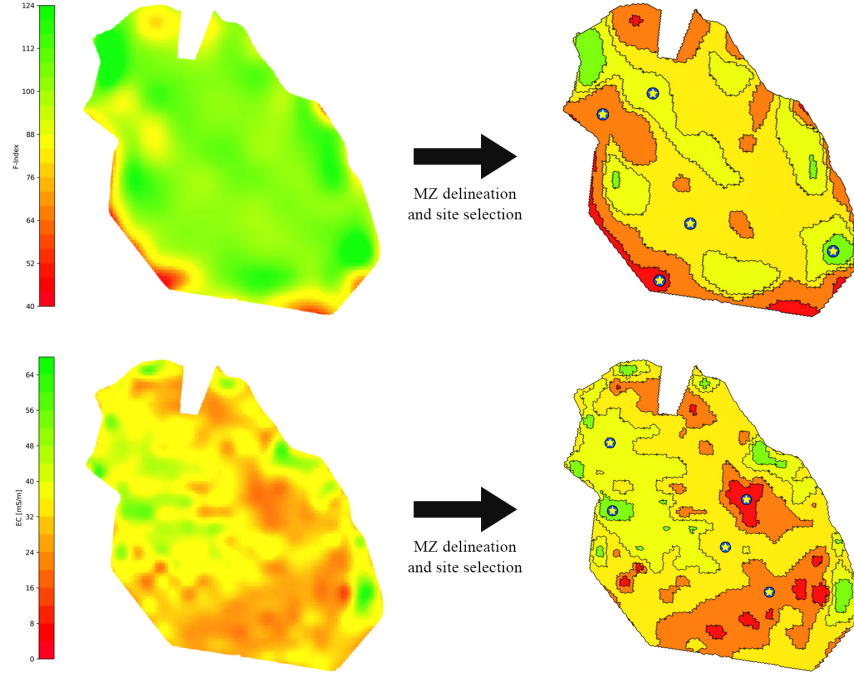


Figure 6.2: Using apparent soil electrical conductivity map as performance function.

Similar to the performance function and MZs, the representative site criteria can be extended in both forms of objectives and constraints. For example, new constraints to avoid power lines, pipelines, and other barriers are just another region to avoid and can be treated the same as headland in g_1 . Additionally, more objectives can be added or the existing objectives can be swapped with another one. For example, instead of F_1 , one may want to use the first quantile and third quantile along with the median. In this case, one can easily swap the F_1 to another objective which calculates the difference of F-Index to another value (e.g., first or third quantile or mean). Figure 6.3 demonstrates the application of first and third quantiles along with the median (second quantile) as a new objective which shows that our framework can be used to satisfy new criteria.

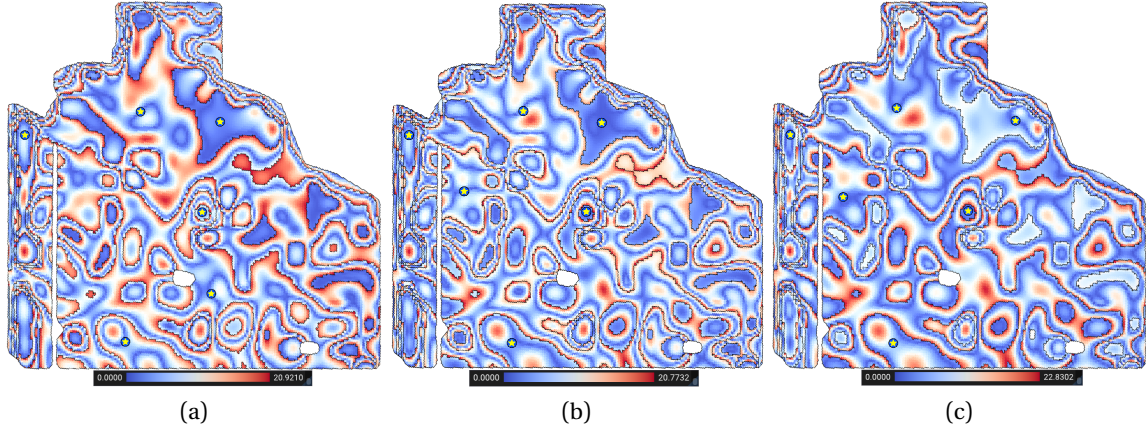


Figure 6.3: A visualization of the difference of the F-Index of each cell with (a) first quantile, (b) second quantile (median), and (c) third quantile of the F-index of each MZ.

6.2 Discussion

Our automatic benchmark site selection method is flexible in various aspects. Specifically, the relative importance of the objectives and the number of MZs are tunable. The following subsections discuss and demonstrate each of these flexibilities of our method.

Optimization Weights

One flexibility of our method is defining the importance of objectives relative to each other by means of weights. The weights are available as a simple interface for the end user to adjust if necessary. For example, if a field is very flat, assigning a lower weight for steepness can help the algorithm to optimize better for other criteria. Looking at Table 6.8, we observe that a lower weight for steepness helped the model lower the difference to the median F-Index by 1.124 units. Recalculating the error E_T after changing the weights is instant, which means that the user can quickly and efficiently test different configurations to get the desired results.

Table 6.8: The effect of weights in objectives. This table is based on MZ 2 on field 1.

Objective	Median F-index (F_1)	Distance to boundary (F_2)	Steepness (F_4)
Range	[0.001, 8.562]	[1.238, 77.104]	[0.015, 5.319]
Best	0.001	77.104	0.015
Mean of all cells	2.342	17.337	1.234
$w_1 = 1, w_2 = 1,$ $w_4 = 1$	1.125	63.797	0.358
$w_1 = 1, w_2 = 1,$ $w_4 = 0.5$	1.143	64.655	0.766
$w_1 = 1, w_2 = 0.5,$ $w_4 = 0.5$	0.933	61.139	0.535
$w_1 = 1, w_2 = 0.25,$ $w_4 = 0.01$	0.446	48.915	0.867
$w_1 = 1, w_2 = 0.01,$ $w_4 = 0.01$	0.001	38.603	0.289
$w_1 = 0.5, w_2 = 1,$ $w_4 = 0.1$	1.460	66.111	1.353

Number of MZs

The number of MZs is usually set based on the area of the field. However, our automated method is adjustable and can delineate a field to any number of MZs. For example, if one wants to take 20 samples of a smaller field, one can set the number of MZs to 20, and the algorithm finds a test point for each MZ. This is practical when a deep understanding of the variability of the soil of a field is needed. Figure 6.4 shows an example of a varying number of MZs on a field. Table 6.9 demonstrates that our method remains fast and performant when the number of MZs increases. We intentionally decreased the D_R for the higher number of MZs, because more benchmark sites have to be fit in the same area. Moreover, for the results presented in Table 6.9, the D_H is set to 0, because in the higher number of MZs, the lower MZ ends up completely in

the headland. In practice, the MZs with very tiny areas are not worth sampling.

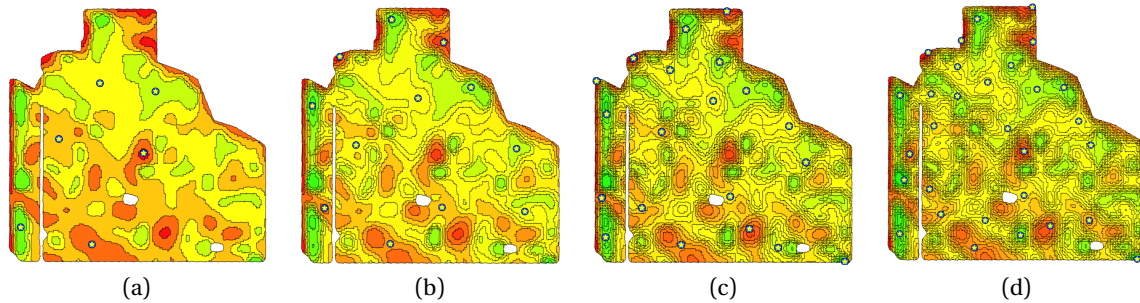


Figure 6.4: Delineating a field into (a) 6, (b) 12, (c) 20, and (d) 30 MZs results in the same number of selected benchmark sites.

Table 6.9: The mean percent improvement for field 1 when the number of MZs varies between 6 to 30 MZs.

Number of MZs	D_R (meter)	Execution time (second)	Mean percent improvement	
6	350	22.54	Median F-index	62.9%
			Distance To Boundary	337.2%
			Steepness	36.2%
12	320	28.74	Median F-index	74.8%
			Distance To Boundary	326.0%
			Steepness	38.0%
20	215	32.75	Median F-index	77.2%
			Distance To Boundary	355.8%
			Steepness	47.6%
30	165	37.61	Median F-index	67.9%
			Distance To Boundary	354.1%
			Steepness	32.4%

Concluding Remarks

Our proposed site selection framework, with the help of the DGGS and DT algorithm, is capable of incorporating arbitrary criteria for choosing a representative area of a field. The

framework is shown to be invariant to The MZ delineation method, the objectives and constraints, and the number of MZs. The importance of the objectives can also be controlled with the means of weights. This framework is shown to produce better results than an MZ composite sampling, and it runs within a few dozen seconds.

Chapter 7

Implementation

Our DT algorithm is implemented as a templated pure function using C++ Programming language. In order to visualize, test, and benchmark the DT algorithm, a visual driver program is implemented using C++ and OpenGL for rendering. The case study of the DT algorithm is implemented using C++ programming language as well which intersects with a project for Telus Agriculture company. In this chapter, we explain the details of our implementation in three sections.

7.1 DT Operation as a Function

For the most flexibility, the DT algorithm is implemented as a function that is templated over the DGGS and index type of the DGGS cells. This allows our implementation to be independent of the DGGS and work with any potential DGGS. The inputs to the DT function are an instance of a DGGS, a polygon to calculate the DT for, a base and a target resolution, and finally a step size. The step size is by default -1 , indicating the single jump from base to target resolution. The step size of 1 can be passed in which case this function will be equivalent to the iterative algorithm. The output of the function is a standard map from the DGGS cell index to a double

number indicating the distance of that cell to the input polygon in meters. The signature of the DT function is as follows:

```
template <typename DGGST, typename IndexT, typename IndexListT>
std::map<IndexT, double, dggs::IndexComparator> calculateDistanceField(
    DGGST const &dggs,
    Polygon const &polygon,
    Resolution const baseRes, Resolution const targetRes
    int stepSize = -1,
);
```

The algorithm that extracts the borders between a field's MZs, doesn't output a single contiguous polygon for each MZ, it rather outputs a list of noncontiguous edges which together form the complex boundaries for MZs. To make the DT function flexible in accepting this kind of input, we have overloaded the DT function. The signature for the overloaded function is as follows:

```
template <typename DGGST, typename IndexT, typename IndexListT>
std::map<IndexT, double, dggs::IndexComparator> calculateDistanceField(
    DGGST const &dggs,
    Polygon const &polygon,
    EdgeList const &edgeList,
    Resolution const baseRes, Resolution const targetRes
    int stepSize = -1,
);
```

Note that this function received an additional input edge list, as well as the polygon. In this case, the polygon is only used to determine the initial set of DGGs cells at the base resolution

to be refined.

There are four important data structures used to implement the DT function. One is `IndexListT parentCells` which holds a list of ancestor cells in each step of refinement. Second, `std::map<IndexT, double> distances`, which holds the distance of all cells to the polygon (or `edgeList`) in the current resolution. After the last iteration of the algorithm, it will contain the output of the algorithm in the target resolution. Third, `std::vector<EdgeList> buffer` (equivalent to `candidateList` in Algorithm 5), which contains a list of the edges of the polygon for each parent cell, that are within the $2d$ of the parent cell. In other words, the buffer holds the reduced search space for each parent cell. Finally, `std::vector<IndexListT> refinedCells` contains the descendent of each parent cell, in the next level of refinement.

The first step of the algorithm is to get an initial set of cells that are covering the input polygon in base resolution. This forms the initial `parentCells` which refines to the target resolution step by step. To get the set of initial cells we use a rasterization algorithm for DGGS [70]. Because the rasterization is done on a coarse resolution, it is efficient. The implementation of the body of the DT algorithm is a C++ implementation of Algorithm 5.

7.2 DT Visualizer and Tester Program

To test the DT function, an implementation of a DGGS is needed. We have used the GIV Digital Earth software package which contains the implementation for Disdyakis Triacontahe-dron DGGS [31]. The GIV Digital Earth software package is the result of a collaboration of a number of different individuals in the University of Calgary GIV lab. We have implemented a set of tools as a scene using GIV Digital Earth software which is depicted in Figure 7.1.

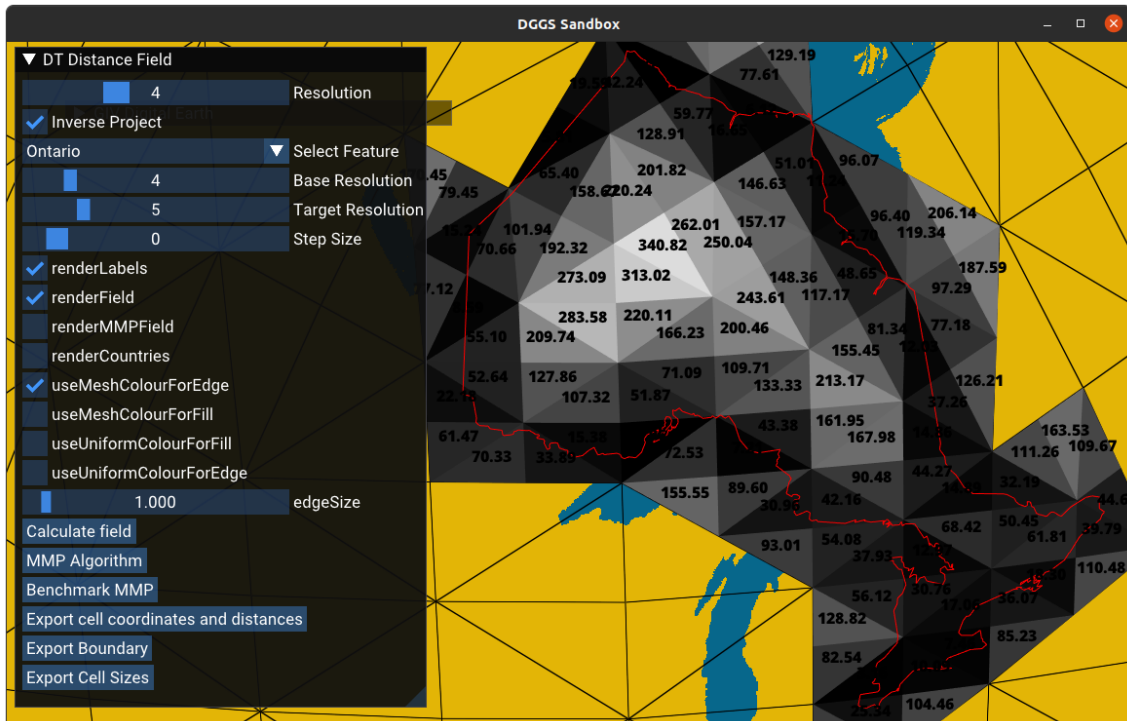


Figure 7.1: The scene implemented for visualizing and testing the DT algorithm that shows a panel of tools.

A user interface (UI) is integrated into this test scene that controls all the inputs to the DT algorithm, including the feature, base and target resolution, and step size. For the input feature, the boundary of all Canadian provinces and territories are available as well as the boundary of the city of Calgary and a farm field in Alberta to be able to test the algorithm with features with different scales. The output of the algorithm is visualized by both a gray-scale colour visualization and by writing the distances in km on DGGS cells as labels (see Figure 7.1). There also are options to show/hide each visualization.

To compare our DT algorithm with the MMP algorithm, we have integrated an implementation of the MMP algorithm from the libigl library into this scene[71]. Google Benchmark tool is used to benchmark both our algorithm and the MMP algorithm. A water-tight indexed

mesh is constructed from the set of covering cells at target resolution to be passed to the MMP algorithm. A list of face indices is passed to the MMP algorithm as the source and the resulting distances are written in another vector passed in with reference.

Last but not least, to be able to use the ArcGIS Pro software to analyze the correctness and the distortion of our algorithm, we have implemented a couple of utility functions to export the distance field and feature as GeoJson files. The distance field is exported as a collection of `Point` data with distance as properties. The feature, however, is exported as a GeoJson with a `LineString` type. To get an exact representation of the feature in ArcGIS Pro, we have sub-sampled each edge of the feature in 10-meter distances.

7.3 Case Study

The case study of this thesis is also implemented as a scene on the GIV Digital Earth software package. Figure 7.2 shows this scene along with all the options for controlling the inputs and weights, visualizations, and exporting the final points. All the visualizations shown in Chapter 6 are exported from this scene. The main input for the benchmark site selection algorithm is stored in a variable called *farmData* which has the following form:

```
struct CellData {  
    Point centroid;  
    unsigned short zone;  
    double Findex;  
};  
using FarmData = std::map<dggs::Index, CellData, dggs::IndexComparator>;
```

which is getting initialized from the F-Map data. The optimization objectives and constraints are implemented using C++ struct as follows:

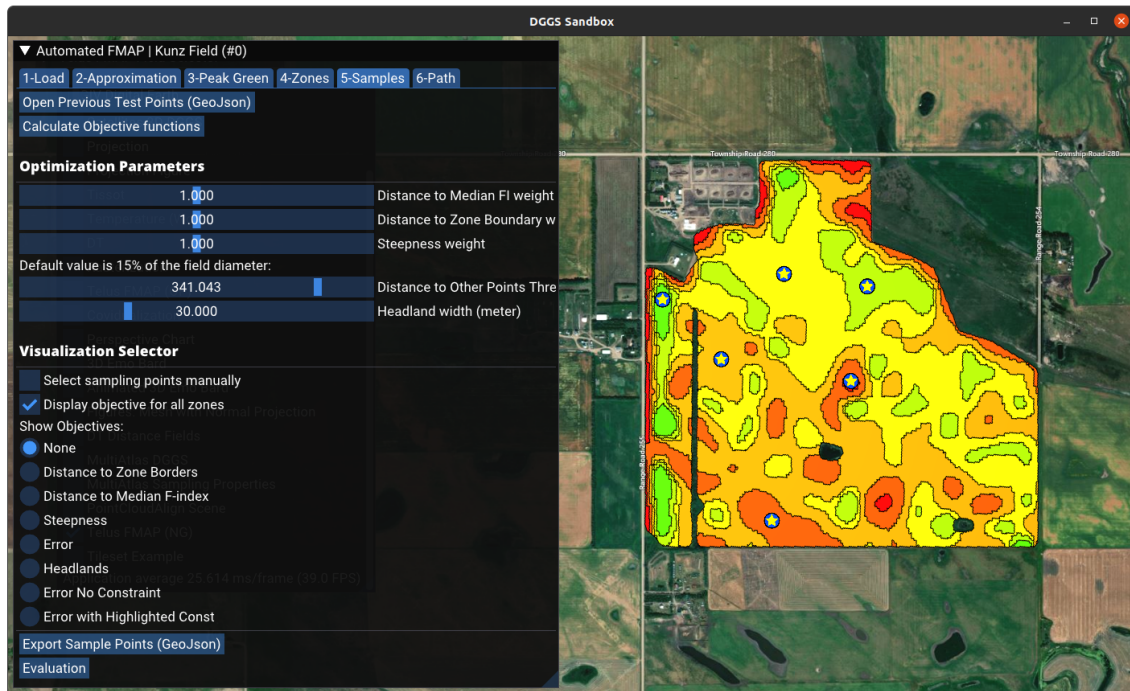


Figure 7.2: The benchmark site selection scene which shows the UI to control the optimization process.

```
using OptimizationObjective = std::map<dggs::Index, double, dggs::IndexComparator>;
using OptimizationConstraint = std::map<dggs::Index, bool, dggs::IndexComparator>;
struct OptimizationParameter {
    OptimizationObjective distanceToBoundary;
    OptimizationObjective distanceToMedianFindex;
    OptimizationObjective distanceToPrevPoint;
    OptimizationObjective steepness;
    OptimizationConstraint isHeadland;
    OptimizationObjective error;
};
```

The optimization objective stores a standard map from each DGGs cell in farmData to the

value of the objective, while the constraint is a standard map to a boolean indicating whether the cell is under a constraint. The calculation of each of these objectives is detailed in Chapter 5.

To be able to use the DT algorithm, we need to construct a vector feature for the boundaries of the MZs. To do this, we iterate over all cells in `farmData`, then we get the neighbours of each cell and test whether the neighbour is in the same MZ. If not, the edge between the cell and the neighbour is a part of the boundary, so we store it in a list. Later, we pass the list to the DT function. Once we have all the objectives and constraints ready in the `OptimizationParameter` struct, calculating the error is straightforward. We combine all of the objectives using a squared sum and we find the minimum of this new objective (i.e., error).

Chapter 8

Conclusion and Future Directions

With the immense amount of data becoming available, distance transform as a tool to analyze such data is important. The problem of distance transform is a solved problem in image processing, but for the geospatial data, image-based methods are unfit. The fast-growing new approach of GIS, DGGS, as a tool to better integrate and analyze such data, needs distance transform. To this end, we have proposed a complete and comprehensive method for efficiently calculating the distance transform on top of a DGGS grid. Our approach properly accommodates any potential DGGS regardless of the shape of the grid cells and the congruency of the refinement scheme. We have discussed how to fine-tune the parameters of our algorithm to get the best results for the border of Ontario as an input feature. We have also compared our method with the image-based methods and general mesh methods. The comparison shows that our method is superior in terms of accuracy and efficiency for large datasets.

Additionally, we introduced an automated framework to solve the problem of selecting benchmark sites as a real-world case study for our efficient DT algorithm. We showed how DT can be used to calculate the distance-based objectives of "away from MZ boundaries" and "close to previously sampled points", and distance-based constraints of "avoiding certain

regions“ like headlands and ”avoiding the proximity of benchmark sites in other MZs“. We showed that this time-consuming process can be efficiently solved and automated on a DGGs using DT and the produced benchmark sites are better than a MZ composite sample. The repeatability of the method is shown by testing the automated method on five different fields. This method takes only a few seconds for a new field in comparison to the manual process by the GIS technician which takes days. We discussed that our method is general and flexible. We showed that different performance functions including EC and different delineation methods as well as new criteria can be used within our framework. To the best of your knowledge, this work is the first automated method for selecting benchmark sites within an agricultural field.

There is still some important future work to be done. First of all, one easy improvement for our DT algorithm is parallelization. As the calculation of DT for each DGGs cell is completely independent of the other cells, the DT algorithm can be parallelized over the coarse cells easily. However, further analysis is required to understand the performance gains of such parallelization.

Second, Our DT algorithm calculates the distance based on a spherical great-circle arc calculation. Despite most of the DGGs that use a sphere for their reference model of the earth, some other DGGs use an oblate spheroid to provide a more accurate representation. We suspect that our method is applicable to such DGGs if one can define an efficient distance calculation method for the ellipsoid. However, further research is required to prove this claim.

Lastly, another interesting active branch of DGGs research is 3D DGGs. There exist some works that aim to build a 3D DGGs or extend current 2D DGGs to 3D DGGs [72, 73]. For this work, we have assumed a 2D DGGs, but the same idea might be able to be reimplemented for a 3D DGGs. The initial idea is to prove Theorem 1 for 3D shapes using a bonding sphere instead of a bounding circle and use this new theorem to build the algorithm. However, more research is required to evaluate this idea and make any potential necessary changes to the algorithm.

Bibliography

- [1] Trang VoPham, Jaime E. Hart, Francine Laden, and Yao-Yi Chiang. Emerging trends in geospatial artificial intelligence (geoai): potential applications for environmental epidemiology. *Environmental Health*, 17(1):40, Apr 2018. ISSN 1476-069X. doi: 10.1186/s12940-018-0386-x. URL <https://doi.org/10.1186/s12940-018-0386-x>.
- [2] Natural Resources Canada. Forest fires, 2023. URL <https://natural-resources.canada.ca/our-natural-resources/forests/wildland-fires-insects-disturbances/forest-fires/13143>.
- [3] Michael J de Smith. Distance transforms as a new tool in spatial analysis, urban planning, and gis. *Environment and planning B: Planning and design*, 31(1):85–104, 2004.
- [4] Sigurjon N Kjaernested, Magnus Th Jonsson, and Halldor Palsson. Methodology for pipeline route selection using the nsga ii and distance transform algorithms. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 54822, pages 543–552, 2011.
- [5] Hlynur Kristinsson, Magnus Thor Jonsson, and Fjóla Jónsdóttir. Pipe route design using variable topography distance transforms. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 4739, pages 331–336, 2005.

- [6] Hao Pu, Hong Zhang, Wei Li, Jiaying Xiong, Jianping Hu, and Jie Wang. Concurrent optimization of mountain railway alignment and station locations using a distance transform algorithm. *Computers & Industrial Engineering*, 127:1297–1314, 2019. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2018.01.004>. URL <https://www.sciencedirect.com/science/article/pii/S0360835218300044>.
- [7] ArcGIS. Euclidean distance. <https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/euclidean-distance.htm>, 2021. Accessed: 2021-06-17.
- [8] Ali Mahdavi-Amiri, Troy Alderson, and Faramarz Samavati. A survey of digital earth. *Computers & Graphics*, 53:95–117, 2015.
- [9] Troy Alderson, Matthew Purss, Xiaoping Du, Ali Mahdavi-Amiri, and Faramarz Samavati. Digital earth platforms. In *Manual of digital earth*, pages 25–54. Springer, Singapore, 2020.
- [10] Michael F. Goodchild. Reimagining the history of gis. *Annals of GIS*, 24(1):1–8, 2018. doi: 10.1080/19475683.2018.1424737. URL <https://doi.org/10.1080/19475683.2018.1424737>.
- [11] Andrew Rawson, Zoheir Sabeur, and Mario Brito. Intelligent geospatial maritime risk analytics using the discrete global grid system. *Big Earth Data*, 6(3):294–322, 2022. doi: 10.1080/20964471.2021.1965370.
- [12] Eleanor A. Bash, Lakin Wecker, Mir Mustafizur Rahman, Christine F. Dow, Greg McDermid, Faramarz F. Samavati, Ken Whitehead, Brian J. Moorman, Dorota Medrzycka, and Luke Copland. A multi-resolution approach to point cloud registration without control points. *Remote Sensing*, 15(4), 2023. ISSN 2072-4292. doi: 10.3390/rs15041161. URL <https://www.mdpi.com/2072-4292/15/4/1161>.

- [13] David Bommes and Leif Kobbelt. Accurate computation of geodesic distance fields for polygonal curves on triangle meshes. In *VMV*, volume 7, pages 151–160, 2007.
- [14] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Commun. ACM*, 60(11):90–99, oct 2017. ISSN 0001-0782. doi: 10.1145/3131280. URL <http://doi.acm.org/10.1145/3131280>.
- [15] Meysam Kazemi, Lakin Wecker, and Faramarz Samavati. Efficient calculation of distance transform on discrete global grid systems. *ISPRS International Journal of Geo-Information*, 11(6), 2022. ISSN 2220-9964. doi: 10.3390/ijgi11060322. URL <https://www.mdpi.com/2220-9964/11/6/322>.
- [16] Azriel Rosenfeld and John L Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM (JACM)*, 13(4):471–494, 1966.
- [17] Olivier Cuisenaire. Distance transformations: fast algorithms and applications to medical image processing. Technical report, Louvain-la-Neuve, Belgium, 1999.
- [18] Yan Wang, Xu Wei, Fengze Liu, Jieneng Chen, Yuyin Zhou, Wei Shen, Elliot K Fishman, and Alan L Yuille. Deep distance transform for tubular structure segmentation in ct scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3833–3842, 2020.
- [19] H-C Liu and Mandyam D. Srinath. Partial shape classification using contour matching in distance transformation. *IEEE Transactions on pattern analysis and machine intelligence*, 12(11):1072–1079, 1990.
- [20] Stéphane Lavallée and Richard Szeliski. Recovering the position and orientation of free-form objects from image contours using 3d distance maps. *IEEE Transactions on pattern analysis and machine intelligence*, 17(4):378–390, 1995.

- [21] Ralph Costa Teixeira. *Curvature motions, medial axes and distance transforms*. Harvard University, 1998.
- [22] Konstantinos Moustakas, Dimitrios Tzovaras, and Michael Gerassimos Strintzis. Sq-map: Efficient layered collision detection and haptic rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):80–93, 2006.
- [23] Young-Ho Choi, Tae-Kyeong Lee, Sang-Hoon Baek, and Se-Young Oh. Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5788–5793. IEEE, 2009.
- [24] PP Acharjya, A Sinha, S Sarkar, S Dey, and S Ghosh. A new approach of watershed algorithm using distance transform applied to image segmentation. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2):185–189, 2013.
- [25] Jun Ma, Zhan Wei, Yiwen Zhang, Yixin Wang, Rongfei Lv, Cheng Zhu, Chen Gaoxiang, Jianan Liu, Chao Peng, Lei Wang, et al. How distance transform maps boost segmentation cnns: an empirical study. In *Medical Imaging with Deep Learning*, pages 479–492. PMLR, 2020.
- [26] Yu-Hua Lee and Shi-Jinn Horng. Fast parallel chessboard distance transform algorithms. In *Proceedings of 1996 International Conference on Parallel and Distributed Systems*, pages 488–493. IEEE, 1996.
- [27] Yu-Hua Lee and Shi-Jinn Horng. Optimal computing the chessboard distance transform on parallel processing systems. *Computer Vision and Image Understanding*, 73(3):374–390, 1999.

- [28] M Akmal Butt and Petros Maragos. Optimum design of chamfer distance transforms. *IEEE Transactions on Image Processing*, 7(10):1477–1484, 1998.
- [29] Ricardo Fabbri, Luciano Da F Costa, Julio C Torelli, and Odemir M Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys (CSUR)*, 40(1):1–44, 2008.
- [30] Tasha Schmaltz and Alex Melnitchouk. Variable zone crop-specific inputs prescription method and systems therefor, Dec 2014.
- [31] John Hall, Lakin Wecker, Benjamin Ulmer, and Faramarz Samavati. Disdyakis triacontahe-dron dggs. *ISPRS International Journal of Geo-Information*, 9(5):315, 2020.
- [32] Krzysztof Chris Ciesielski, Xinjian Chen, Jayaram K Udupa, and George J Grevera. Linear time algorithms for exact distance transform. *Journal of Mathematical Imaging and Vision*, 39(3):193–209, 2011.
- [33] Frank Y Shih and Yi-Ta Wu. Fast euclidean distance transformation in two scans using a 3×3 neighborhood. *Computer Vision and Image Understanding*, 93(2):195–205, 2004.
- [34] Yves Lucet. New sequential exact euclidean distance transform algorithms based on convex analysis. *Image and Vision Computing*, 27(1-2):37–44, 2009.
- [35] Joseph SB Mitchell, David M Mount, and Christos H Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.
- [36] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. *ACM transactions on graphics (TOG)*, 24(3):553–560, 2005.

- [37] G Lavanya, C Rani, and P Ganeshkumar. An automated low cost iot based fertilizer intimation system for smart agriculture. *Sustainable Computing: Informatics and Systems*, 28:100300, 2020. ISSN 2210-5379. doi: <https://doi.org/10.1016/j.suscom.2019.01.002>. URL <https://www.sciencedirect.com/science/article/pii/S2210537918302567>.
- [38] David Schimmelpfennig. Farm profits and adoption of precision agriculture. Technical Report 1477-2016-121190, United States Department of Agriculture (USDA), 2016-10, 2016. URL <http://ageconsearch.umn.edu/record/249773>.
- [39] Aurélie P. Harou, Malgosia Madajewicz, Hope Michelson, Cheryl A. Palm, Nyambilila Amuri, Christopher Magomba, Johnson M. Semoka, Kevin Tschirhart, and Ray Weil. The joint effects of information and financing constraints on technology adoption: Evidence from a field experiment in rural tanzania. *Journal of Development Economics*, 155:102707, 2022. ISSN 0304-3878. doi: <https://doi.org/10.1016/j.jdeveco.2021.102707>. URL <https://www.sciencedirect.com/science/article/pii/S030438782100081X>.
- [40] Martin R Carter and Edward Gerard Gregorich, editors. *Soil sampling and methods of analysis*. CRC press, Boca Raton, FL, USA, 2007. ISBN 978-0-8593-3586-0.
- [41] Alberta Agriculture and Food. *Nutrient Management Planning Guide*. Alberta Agriculture and Food, 2015.
- [42] David J. Mulla. Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems Engineering*, 114(4):358–371, 2013. ISSN 1537-5110. doi: <https://doi.org/10.1016/j.biosystemseng.2012.08.009>. URL <https://www.sciencedirect.com/science/article/pii/S1537511012001419>. Special Issue: Sensing Technologies for Sustainable Agriculture.

- [43] Francis J. Pierce and Peter Nowak. Aspects of precision agriculture. In Donald L. Sparks, editor, *Advances in Agronomy*, volume 67 of *Advances in Agronomy*, pages 1–85. Academic Press, 1999. doi: [https://doi.org/10.1016/S0065-2113\(08\)60513-1](https://doi.org/10.1016/S0065-2113(08)60513-1). URL <https://www.sciencedirect.com/science/article/pii/S0065211308605131>.
- [44] D Keyes and G Gillund. Benchmark sampling of agricultural fields. In *Soils and Crops Workshop*, 1995.
- [45] A Hornung, R Khosla, R Reich, D Inman, and DG Westfall. Comparison of site-specific management zones: Soil-color-based and yield-based. *Agronomy Journal*, 98(2):407–415, 2006.
- [46] CW Fraisse, KA Sudduth, and NR Kitchen. Delineation of site-specific management zones by unsupervised classification of topographic attributes and soil electrical conductivity. *Transactions of the ASAE*, 44(1):155, 2001.
- [47] Claudia Georgi, Daniel Spengler, Sibylle Itzerott, and Birgit Kleinschmit. Automatic delineation algorithm for site-specific management zones based on satellite remote sensing data. *Precision Agriculture*, 19(4):684–707, 2018.
- [48] Alison B Tarr, Kenneth J Moore, Philip M Dixon, C Lee Burras, and Mary H Wiedenhoeft. Use of soil electroconductivity in a multistage soil-sampling scheme. *Crop Management*, 2(1):1–9, 2003.
- [49] Rongjiang Yao, Jingsong Yang, Xiufang Zhao, Xiaobing Chen, Jianjun Han, Xiaoming Li, Meixian Liu, and Hongbo Shao. A new soil sampling design in coastal saline region using em38 and vqt method. *CLEAN–Soil, Air, Water*, 40(9):972–979, 2012.
- [50] Felix Stumpf, Karsten Schmidt, Thorsten Behrens, Sarah Schönbrodt-Stitt, Giovanni Buzzo, Christian Dumperth, Alexandre Wadoux, Wei Xiang, and Thomas Scholten. Incorporating

limited field operability and legacy soil samples in a hypercube sampling design for digital soil mapping. *Journal of Plant Nutrition and Soil Science*, 179(4):499–509, 2016.

- [51] Yiming An, Lin Yang, A-Xing Zhu, Chengzhi Qin, and JingJing Shi. Identification of representative samples from existing samples for digital soil mapping. *Geoderma*, 311:109–119, 2018.
- [52] Lydia Mumbi Chabala, Augustine Mulolwa, and Obed Lungu. Landform classification for digital soil mapping in the chongwe-rufunsa area, zambia. *Agric. For. Fish*, 2:156–160, 2013.
- [53] Jon J Fridgen, Newell R Kitchen, Kenneth A Sudduth, Scott T Drummond, William J Wiebold, and Clyde W Fraisse. Management zone analyst (mza) software for subfield management zone delineation. *Agronomy Journal*, 96(1):100–108, 2004.
- [54] Brian John Kozar. *Predicting soil water distribution using topographic models within four Montana farm fields*. PhD thesis, Montana State University-Bozeman, College of Agriculture, 2002.
- [55] RA MacMillan. A protocol for preparing digital elevation (dem) data for input and analysis using the landform segmentation model (lsm) programs. *Soil Variability Analysis to Enhance Crop Production (SVAECP) Project*, 2000.
- [56] Gabor Milics, Jakab Kauser, and Attila Kovacs. Profit maximization in soybean (glycine max (l.) merr.) using variable rate technology (vrt) in the sárrét region, hungary. Agri-Tech Economics Papers 296767, Harper Adams University, Land, Farm & Agribusiness Management Department, Oct 2019. URL <https://ideas.repec.org/p/ags/haaepa/296767.html>.

- [57] Jason Cathcart, Karen Cannon, and Jody Heinz. Selection and establishment of alberta agricultural soil quality benchmark sites. *Canadian Journal of Soil Science*, 88(3):399–408, 2008.
- [58] Suzanne Card. *Evaluation of two field methods to estimate soil organic matter in Alberta soils*. AESA Soil Quality Monitoring Program, 2004.
- [59] Ricson Lorenzo Ines, Jose Paulo Banzon Tuazon, and Morrimer Nemesio Daag. Utilization of small farm reservoir (sfr) for upland agriculture of bataan, philippines. *International Journal of Applied Agricultural Sciences*, 4(1):1–6, 2018. doi: 10.11648/j.ijaas.20180401.11.
- [60] Nutrien Ag Solutions. Soil sampling tips, 2019. URL <https://www.nutrienagsolutions.ca/about/news/soil-sampling-tips>.
- [61] Canola Council of Canada. How to take a good soil sample, 2013. URL <https://www.canolacouncil.org/canola-watch/2013/10/02/how-to-take-a-good-soil-sample/><https://www.canolacouncil.org/canola-watch/2013/10/02/how-to-take-a-good-soil-sample/>.
- [62] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.
- [63] Brian Everitt. *Introduction to Optimization Methods and their Application in Statistics*. Springer Netherlands, 2012. ISBN 9789401079174. URL <https://books.google.ca/books?id=k0qJngEACAAJ>.
- [64] Jorge Nocedal Stephen J Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. ISBN 978-0-387-30303-1. doi: 10.1007/978-0-387-40065-5. URL <http://link.springer.com/10.1007/978-0-387-40065-5>.

- [65] Kevin Sahr. Location coding on icosahedral aperture 3 hexagon discrete global grids. *Computers, Environment and Urban Systems*, 32(3):174–187, 2008. ISSN 0198-9715. doi: <https://doi.org/10.1016/j.compenvurbsys.2007.11.005>. URL <https://www.sciencedirect.com/science/article/pii/S0198971507000889>. Discrete Global Grids.
- [66] Piotr Mazur, Dariusz Gozdowski, and Elżbieta Wójcik-Gront. Soil electrical conductivity and satellite-derived vegetation indices for evaluation of phosphorus, potassium and magnesium content, pH, and delineation of within-field management zones. *Agriculture*, 12(6), 2022. ISSN 2077-0472. doi: 10.3390/agriculture12060883. URL <https://www.mdpi.com/2077-0472/12/6/883>.
- [67] Nathalie Pettorelli. *The normalized difference vegetation index*. Oxford University Press, 2013.
- [68] João Serrano, Shakib Shahidian, José Marques da Silva, Luís Paixão, Francisco Moral, Rafael Carmona-Cabezas, Sónia Garcia, José Palha, and João Noéme. Mapping management zones based on soil apparent electrical conductivity and remote sensing for implementation of variable rate irrigation—case study of corn under a center pivot. *Water*, 12(12), 2020. ISSN 2073-4441. doi: 10.3390/w12123427. URL <https://www.mdpi.com/2073-4441/12/12/3427>.
- [69] Marco Pohle and Ulrike Werban. Near surface geophysical data (Electromagnetic Induction - EMI, Gamma-ray spectrometry), August 2017), Selbitz (Elbe), Germany, 2019. URL <https://doi.org/10.1594/PANGAEA.910272>. Supplement to: Rentschler, Tobias; Werban, Ulrike; Ahner, Mario; Behrens, Thorsten; Gries, Phillipp; Scholten, Thomas; Teuber, Sandra; Schmidt, Karsten (2020): 3D mapping of soil organic carbon content and soil moisture with multiple geophysical sensors and machine learning. *Vadose Zone Journal*, 19(1), <https://doi.org/10.1002/vzj2.20062>.

- [70] Amirhossein Mirtabatabaeipour and John Hall. Resolution adaptive vector rasterization in discrete global grid systems. Technical report, 2022.
- [71] Alec Jacobson, Daniele Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. <https://libigl.github.io/tutorial/#exact-discrete-geodesic-distances>.
- [72] Benjamin Ulmer, John Hall, and Faramarz Samavati. General method for extending discrete global grid systems to three dimensions. *ISPRS International Journal of Geo-Information*, 9:233, 04 2020. doi: 10.3390/ijgi9040233.
- [73] Benjamin Ulmer and Faramarz Samavati. Toward volume preserving spheroid degenerated-octree grid. *GeoInformatica*, 24(3):505–529, 2020.

Appendix

The copyright licence for Figure [3.4](#) and the permission to use co-authored paper [\[15\]](#) as a part of my thesis (signature removed) are attached respectively to the next pages.



Location coding on icosahedral aperture 3 hexagon discrete global grids

Author: Kevin Sahr

Publication: Computers, Environment and Urban Systems

Publisher: Elsevier

Date: May 2008

Copyright © 2007 Elsevier Ltd. All rights reserved.

Order Completed

Thank you for your order.

This Agreement between Mr. Meysam Kazemi ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

Your confirmation email will contain your order number for future reference.

License Number 5612350087487

[Printable Details](#)

License date Aug 19, 2023

Licensed Content

Licensed Content Publisher	Elsevier
Licensed Content Publication	Computers, Environment and Urban Systems
Licensed Content Title	Location coding on icosahedral aperture 3 hexagon discrete global grids
Licensed Content Author	Kevin Sahr
Licensed Content Date	May 1, 2008
Licensed Content Volume	32
Licensed Content Issue	3
Licensed Content Pages	14

Order Details

Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
Format	electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No

About Your Work

Title	Efficient Calculation of Distance Transform on Discrete Global Grid Systems and Its Application in Automatic Soil Sampling Site Selection
Institution name	University of Calgary
Expected presentation date	Sep 2023

Additional Data

Portions	Fig. 26
----------	---------

Requestor Location

Requestor Location

Mr. Meysam Kazemi

\$ Price

Total0.00 CAD

Tax Details

Publisher Tax ID

GB 494 6272 12

Total: 0.00 CAD

CLOSE WINDOW

ORDER MORE

© 2023 Copyright - All Rights Reserved | [Copyright Clearance Center, Inc.](#) | [Privacy statement](#) | [Data Security and Privacy](#)

[For California Residents](#) | [Terms and Conditions](#)

Comments? We would like to hear from you. E-mail us at customer-care@copyright.com

August 6th, 2023

Dear Meysam,

I grant you permission to use our co-authored paper, "Efficient Calculation of Distance Transform on Discrete Global Grid Systems", published in ISPRS International Journal of Geo-Information on 25 May 2022 in your thesis.

You have my full permission to include the paper as part of your research. Please ensure that proper credit is given to me as a co-author and that all citations are done correctly.

If you have any questions or concerns, please do not hesitate to contact me.

Sincerely,

Lakin Wecker