Spatial linear network Voronoi analysis to quantify accessibility of police stations in South Africa

By

Arthur Antonio

18117172

Supervisor: Inger Fabris-Rotelli

Co-supervisors: Renate Thiede, René Stander

Submitted in partial fulfillment for the requirements of the degree

MSc Advanced Data Analytics

in the Faculty of Natural and Agricultural Sciences

University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

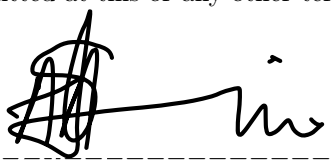
November 29, 2024

Abstract

This study quantifies the overlap between existing police precinct boundaries and theoretically optimal boundaries derived from Voronoi diagrams based on Euclidean and network distances. Spatial similarity measures are used to analyse the relationship between boundary overlap and police station accessibility, hypothesising that reduced overlap corresponds to decreased accessibility. Accessibility, in this mini-dissertation, refers to how easily an individual can reach a police station, with closer points being more accessible. The analysis extends to the potential effects of boundary placement on crime rates, suggesting that greater inaccessibility of police stations may correlate with less crimes reported in that precinct. By quantifying these relationships, this research evaluates the effectiveness of current precinct boundaries and their potential influence on crimes reported. For precincts with low similarity values, indicating low accessibility, we analyse the proportional change in the number of crimes reported after boundary modifications. A decrease in reported crimes within the new boundaries generated by Voronoi models would support our hypothesis. This decrease would suggest that a significant portion of crimes are now being reported to other, nearer, and more accessible police stations.

Declaration

I, *Arthur Antonio,* declare that this mini-dissertation, submitted in partial fulfillment of the degree *MSc Advanced Data Analytics*, at the University of Pretoria, is my own work and has not been previously submitted at this or any other tertiary institution.



Arthur Antonio

November 29, 2024

Acknowledgements

I extend my deepest gratitude to my supervisors, whose invaluable guidance and unwavering support have been instrumental throughout this challenging yet profoundly rewarding journey. Their encouragement has been crucial in shaping the development of this mini-dissertation and my Master's degree.

I also acknowledge the NAS Ethics Committee (NAS116/2019) for granting permission to use the data in this study, as well as Prof. G. Breetzke from the University of Pretoria's Department of Geography, Geoinformatics, and Meteorology for providing the data essential to this research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This study aims to quantify the similarity in spatial lattice data, which are derived by dividing a defined region into subregions where specific outcomes are recorded. Examples include the number of accidents in provinces or the proportion of people living in poverty [133]. We analyse three data sets to explore this overlap. The first data set contains police station precinct boundaries and locations. The data set made use of contains crime statistics for each police station in South Africa. The second data set contains road network information spanning the whole of South Africa. Using the police station locations, we generate two types of Voronoi diagrams [10]: one based on Euclidean distance and the other based on distance along the road network structure (termed network Voronoi). The network Voronoi diagram incorporates the road network structure, by calculating the shortest path distance in the network to quantify proximity to the police station. This means, network Voronoi assigns areas closest to a police station based on the shortest path distance on the road network [10]. Voronoi diagrams are chosen over other spatial modeling techniques due to their ability to provide an intuitive and mathematically rigorous partitioning of space based on proximity [49, 3, 87]. We are interested in whether the police precinct borders arise because of convenience or genuine accessibility to a police station. By comparing the administrative precinct boundaries with those generated through Voronoi diagrams, we assess accessibility. Specifically, we evaluate overlap in two ways: through the polygons themselves and via road networks extracted from the corresponding polygons. A high degree of overlap suggests effective accessibility, whereas areas with minimal overlap may indicate limited access to police services. Herein, we define police station accessibility as the ease with which individuals can reach that police station's services [54]. Our accessibility measure emphasises proximity to police stations because we use Voronoi diagram models. That is assuming other factors remain constant, regions or points within a precinct that are closer to a police station are considered to have easier access (more accessibility). Areas farther from the station are assumed to face greater difficulty in accessing its services, leading to lesser accessibility.

We hypothesise that the location of a precinct's boundaries may impact crime rates in that precinct. For example, we expect that less accessible police stations may correlate with less crime reported in that precinct. This will lead to individuals or residents of that particular precinct reporting to other precincts that are more accessible, which will in turn increase the number of crimes reported in the more accessible station.

The third data set is for the Khayelitsha area in Western Cape. This data set contains crime statistics for three police precincts, with a point pattern aspect representing the locations of reported crimes [34]. We will conduct a separate study for validation purposes on these three police precincts in the Khayelitsha area. This is done by reconstructing both Euclidean and network Voronoi diagrams based on this new area, and thereafter determine if changing the boundaries has an effect on the crime rate or crimes reported at particular police stations.

We adopt several key terminologies that are integrated for this mini-dissertation's objectives. Firstly, **boundary analysis** is used to describe large areas that are divided, into smaller districts based on pre-defined criteria. This helps examine localised accessibility patterns [132]. Secondly, **proximity analysis**, as illustrated in Figure 1.1, measures the distance between police stations and regional landmarks, such as households, using either Euclidean or network distance. Thirdly, **buffer analysis**, with an example shown in Figure 1.2. This process generates zones around police stations to identify areas that fall within these buffer boundaries. This gives an idea or understanding of service reach and coverage [23]. Fourthly, **network analysis** involves examining the structure of networks to optimise service distribution. This process identifies the shortest or least-cost routes to a police station. The "cost" is measured in terms of distance. Here, we use the Network Voronoi Diagram [84] to compute these costs, aiming to quantify accessibility to police stations. Additionally, the costs are calculated to group areas with similar cost ranges into a particular precinct. In other words, areas assigned to the same precinct share the same cost (in terms of distance) to access services from a specific police station, as compared to any other station on the network. Lastly, **spatial overlay** refers to the superimposition of two spatial patterns to quantify their similarity. This enables a comparative analysis of spatial relationships [72].

Figure 1.1 illustrates two police stations and the surrounding neighbourhoods within a specified radius. The locations of the police stations are marked with red circles. The solid blue boundary at the top highlights areas on the map that have easier access to that police station. Similarly, the second police station, encompassed by dashed blue boundaries, represents regions on the map where this station is more accessible. Figure 1.2 demonstrates the concept of buffer analysis. Police station locations are also marked with red circles. The dotted circles around these stations indicate areas closest to each respective station. Note how, as the radius increases, the areas that can be accessed or serviced by a particular station expand accordingly.

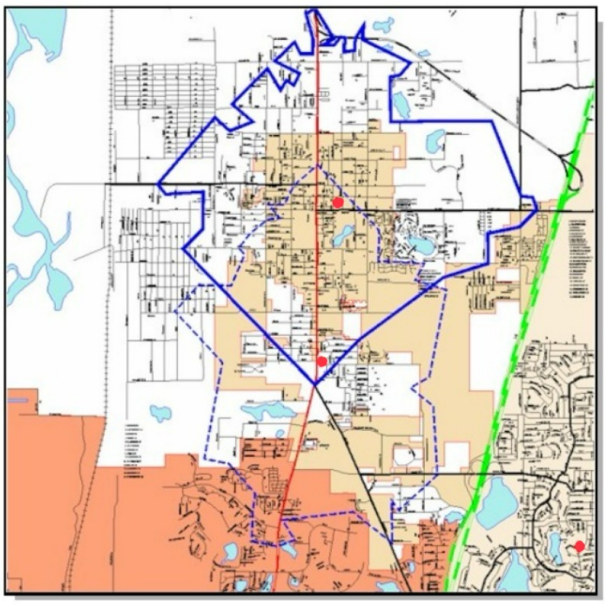


Figure 1.1: Network analysis between two police stations and households. Areas within the solid blue boundary are closest to the top station, while those within the dashed boundary are nearest to the bottom station[1].

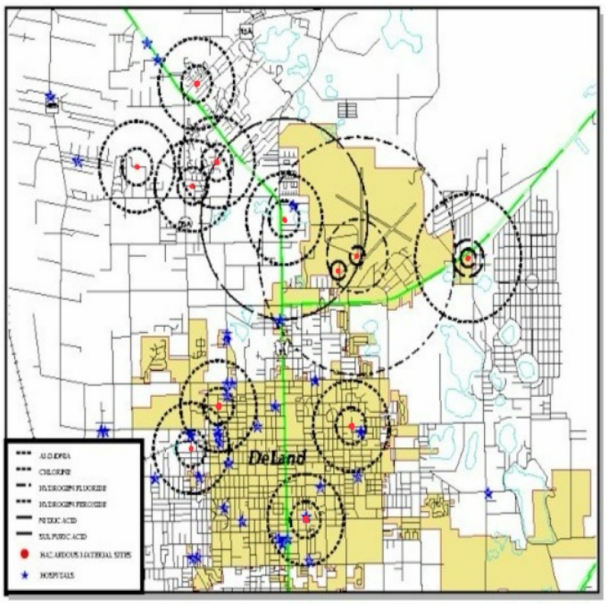


Figure 1.2: Buffer analysis of the area surrounding a police station. Areas within the buffer boundary are closest to this station compared to any other in the region. The buffer also highlights areas that can potentially be served by a specific station[2].

1.1 Literature Review

Voronoi diagrams [83] are a useful tool for delineating spatial regions based on proximity to a set of points, making them relevant for understanding spatial relationships and overlaps. To fully explore their potential applications, we conduct a review of the existing literature. This review also examines how Voronoi diagrams can be applied in various fields, such as criminology, airport management, urban planning, resource management, and ecology. Additionally, we investigate how Voronoi diagrams can be used to uncover spatial patterns. Understanding spatial similarity is also important, therefore, the second part of this review focuses on spatial similarity tests, discussing their significance and surveying the relevant literature. We outline the specific spatial similarity tests employed in various studies and their role in assessing spatial relationships. The last part briefly looks into the existing work on accessibility and criminology.

1.1.1 Voronoi Diagrams

Georgy Voronoi, a Russian mathematician, pioneered the concept of Voronoi diagrams in 1908, initially known as Thiessen polygons [6]. These diagrams were first used to map natural terrain. Later, Voronoi's work found significant application in crystallography, where it was used to analyse crystal structures [6].

[1]Source for Figure 1.1 `https://bit.ly/geographic-spatial-analysis` (accessed on 26 February 2023).
[2]Source for Figure 1.2 `https://bit.ly/BufferAnalysis` (accessed on 26 February 2023).

Given a set of n points of interest on a plane, a Voronoi diagram is created by dividing the plane into convex polygons so that each polygon has strictly one generating point and every point in a specific polygon is nearer to its generating point than to any other point [15]. Figure 1.3 shows an example of a Voronoi diagram, the generator points are the black circles and the convex polygons are made up of the red lines.

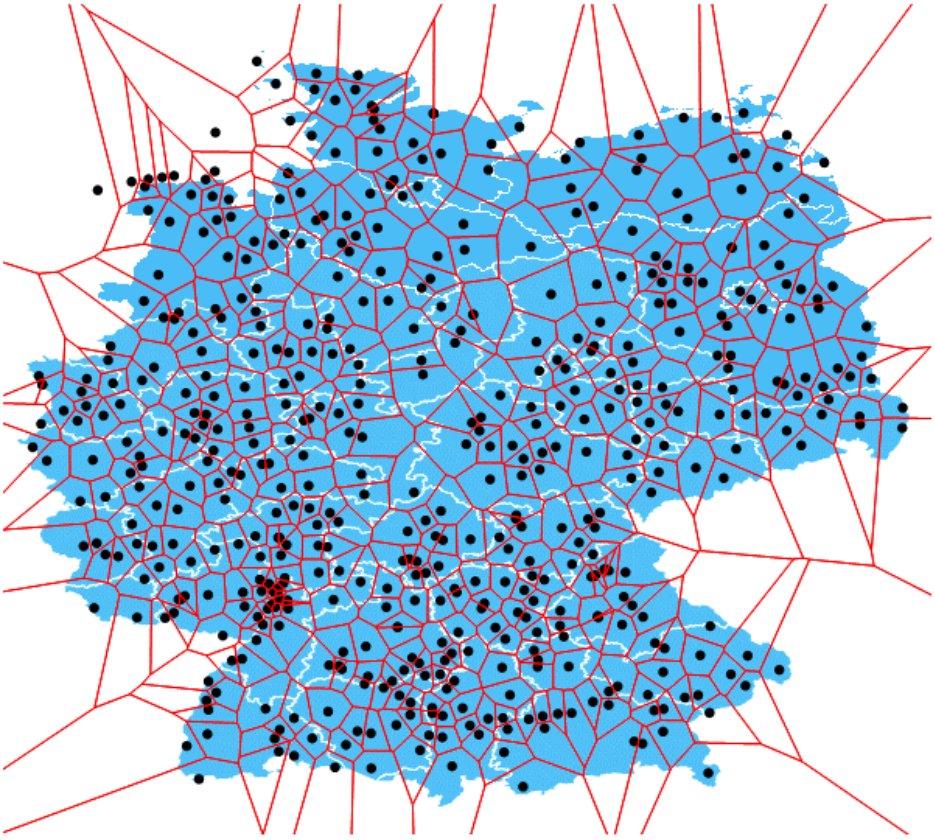


Figure 1.3: An example of a Voronoi diagram on a plane could represent Wi-Fi hotspots, where each black point indicates a hotspot location. The area inside each red polygon includes all points that are closer to that specific hotspot than to any other.

From an algorithmic perspective, Fortune's sweepline algorithm [44] remains a key method for constructing standard Voronoi diagrams. Fortune states in [45] that the Delaunay triangulation, a dual structure of Voronoi diagrams, creates a unique triangulation with no other sites in the circumsphere of each triangle. Voronoi diagrams and their duals are essential to computational geometry due to their many mathematical and natural science applications.

Voronoi diagrams have been underutilised in crime mapping compared to other spatial analysis techniques such as kernel density estimation, Ripley's K-function, and spatial autocorrelation [110]. Previous studies have predominantly focused on these traditional methods to identify crime hotspots, spatial clustering, and patterns of criminal behavior. However, by dividing space according to proximity, Voronoi diagrams provide a novel method that makes it possible to analyse spatial distribution, criminal concentration, and network-based crime patterns. The promise of Voronoi diagrams in crime analysis has been shown by recent studies, such as the one on commercial robbery in Campinas, Brazil, which uncovered spatial correlations

that are difficult to capture using traditional techniques [110]. This increased interest in Voronoi diagrams points to their potential for improving crime geovisualisation and helping criminologists comprehend spatial dynamics [110]. Researchers have also proposed new algorithms that enhance the speed and efficiency of Voronoi diagram computations [69].

Wilebore and Coomes [126] use Voronoi diagrams to forecast village borders in agricultural environments without administrative boundary data. They compare different types of polygons to GPS data for accuracy. This new approach used village size data from surveys to optimise multiplicatively weighted Voronoi polygon weights. They also deduced that weighted Voronoi polygons align with village areas better than unweighted techniques. This technique is used to predict administrative boundaries (which could be replaced by police boundaries) in [126] using weighted Voronoi polygons which in turn is useful for spatial and crime analysis. This can be extended to optimise police station placement to reduce crime rates in underserved areas. The method is useful for land-use studies since researchers and NGOs can use it [70].

Xia et al. [128] use Voronoi diagrams in modeling urban expansion within metropolitan areas. Their approach integrates a refined gravitational field model and bidirectional flow analysis with partitioned cellular automata. Weighted Voronoi diagrams are used to better outline the areas of urban influence, representing actual proximity-based interactions between urban centers and their surroundings. This methodology contributes to the prediction of urban growth patterns. In addition, Voronoi diagrams are used to generate urban layouts that optimise various criteria, such as accessibility and sustainability. For example, Voronoi diagrams can be used to determine the optimal location of public services such as hospitals and schools in a city [79].

The construction of network Voronoi diagrams has undergone numerous improvements. Ai et al. [1] presented a constrained network Voronoi diagram derived from stream flow principles. Tan et al. devised a method for computing network distance with a shortest-path tree [116]. Furthermore, Okabe et al. made a substantial contribution to the discipline by defining six distinct types of generalised network Voronoi diagrams [84]. Okabe et al. [84] demonstrated that for short urban distances under 500 meters, Euclidean measurements can differ significantly from the shortest path distances along a network. This discrepancy makes traditional Voronoi diagrams, which rely on Euclidean distance, unsuitable for accurately depicting service areas in urban environments. Generalised network Voronoi diagrams were introduced [84] to address the distance metric issue. These diagrams use various network elements such as points, lines, and polygons as generators. Depending on the use case, these Voronoi diagrams use improved distance metrics, such as inward/outward distances and weighted shortest path distances [84].

Network Voronoi Diagrams are used for spatial and mobile database range search queries by Xuan et al. [129]. Modern Geographic Information System applications require network distances that reflect real-world pathways, which R-trees cannot provide. Existing algorithms for network distance range searches generally

struggle with processing performance and storage. Network Voronoi Diagrams are suggested to address these restrictions. With increased network distance-based range search efficiency and accuracy.

Luyan et .al [74] uses concurrency and auto-wave distribution in an enhanced Pulse Coupled Neural Network to dynamically create weighted network Voronoi diagrams. The method generates auto-waves from beginning neurons and transmits them along the shortest path between neurons at a predetermined pace until all neurons fire, at which time transmission stops. By weighting facility points and road features, the system better accommodates urban facility points and road networks. Road network nodes are represented as neurons, and facility points are projected onto the adjacent road segments. The weighted network Voronoi diagram better defines service region boundaries.

1.1.2 Overview of Spatial Similarity Analysis

Spatial overlap analysis examines how different geographic layers or maps intersect or coexist [114]. This analysis helps us understand the connections, patterns, and interactions between various geographical features or phenomena. It is useful in fields such as ecology, urban planning, public health, and natural resource management [23]. By showing how different areas overlap or separate, informed decisions can be made about resource allocation, land use, species conservation, disease spread, and other spatial issues [50, 25].

In [28], boundaries are described as the edges that separate different areas or patches. They mark the transition where one ecological unit ends and another begins. The study points out that analysing these boundaries is important for understanding spatial dynamics and interactions occurring at the interfaces between units. The paper also emphasises the application of spatial overlap analysis to measure the degree of overlap between patches along these boundaries. This methodology was used to assist ecologists in examining interaction patterns, edge effects, and habitat appropriateness in these transitional zones [28].

Pontius and Schneider [97] assess a land-cover change model in the Ipswich watershed, Massachusetts, by applying the Relative Operating Characteristic approach. In order to evaluate how well a model predicts real land-cover data, this method uses spatial overlap analysis of maps, including metrics like area of overlap and the Kappa statistic. The research evaluates the model's accuracy in tracking changes in land cover in the region.

The Kappa metric has been widely used for accuracy assessment (spatial overlay), but Pontius and Millones, in their "Death to Kappa" paper, advocated against its use [98]. Pontius recommended interpreting quantity and allocation disagreements for error assessment [96]. Despite this criticism, approximately half of the 2000 papers citing "Death to Kappa" continued to use Kappa. Pontius emphasises the value of learning from differences rather than agreements, elaborating on appropriate metrics for various spatial overlay scenarios in [96]. While half of these papers reported quantity and allocation differences, many failed to relate these

differences meaningfully to their research questions. Many papers simply presented metrics and considered their findings satisfactory by citing Pontius's claims regarding particular metric values, frequently lacking thorough investigation. The "Death to Kappa" critiques the indices, asserting that they are "useless, misleading, and/or flawed" for practical remote sensing applications [98]. Following more than ten years of engagement with these indices, the authors recommend discarding Kappa in favor of more straightforward summary metrics, that is "quantity disagreement and allocation disagreement" [96]. In light of this, we follow their recommendation to replace Kappa with quantity and allocation disagreement while also using Structural Similarity Index (SSIM) to enhance spatial similarity assessment.

In domains like image analysis, spatial epidemiology, and ecological studies, where local patterns differ, similarity measurements are important [43]. Wang et al.'s SSIM has become a standard for assessing image and video quality by measuring structural similarity [123]. This is an alternative to error-summation methods like Mean Squared Error, which often mismatch human visual perception. SSIM is used in image processing, computer vision, and multimedia applications. This is because its structural element focus matches human visual sensitivity. SSIM was developed by Wang et al. [123] to assess image quality by evaluating brightness, contrast, and structure. SSIM calculated image similarity by independently analysing these components and integrating them into a single score. Multiscale SSIM [124] provides a higher connection with perceptual quality in high-resolution images by assessing similarity across many scales, capturing global and local information. Zeng and Wang [130] enhanced SSIM for 3D data, enabling volumetric and 3D imaging applications. Sampat et al. [107] created Complex Wavelet SSIM, which improves resilience against geometric distortions like scaling, rotation, and translation. SSIM outperforms Peak Signal-to-Noise Ratio and MSE in visual perception and computational performance, making it suitable for real-time applications. To compare structural similarities in non-image data, such as road networks, SSIM can be applied through preprocessing methods outlined in [61]. This will be covered in Sections 4.2 and 5.2.

SSIM-based similarity mapping has diverse applications beyond image analysis. In urban analysis, it can compare land use patterns or monitor changes in urban morphology over time. In ecological studies, SSIM is used to assess habitat structures or species distribution patterns [43]. Epidemiological studies employ SSIM to compare spatial distributions of disease outbreaks, identifying regions with similar patterns of disease incidence [117]. Local SSIM mapping requires careful consideration of the sliding window size, as it impacts spatial resolution. Smaller windows yield detailed similarity maps but are more sensitive to noise, while larger windows offer generalised comparisons, potentially smoothing local details. Window size should thus align with the spatial scale of the patterns under study [16]. Though SSIM offers a good local similarity measures, it can be computationally intensive, for high-resolution raster images or large data sets. To address this, Multi-Scale SSIM provides an optimised approach by evaluating similarity across scales, offering comprehensive comparisons with reduced computational demands [124].

Edge effects present another challenge when applying SSIM to spatial data. These effects occur when windows near raster boundaries lack sufficient neighbouring pixels, potentially skewing similarity scores. Padding techniques or edge correction methods can minimise this bias, ensuring consistent treatment of boundary regions with the interior of the raster grid [33]. As part of our spatial similarity tests in this research, we additionally examine the degree of overlap between polygons. The Polygonal Dissimilarity Function (PDF), a unique statistic created to quantify variations across geographical polygons, was discovered in [58]. Both spatial and non-spatial factors are taken into account by this function. In addition to exterior criteria like distance to other features, the PDF contains spatial data like shape and area. Non-spatial characteristics like population and land use are also taken into consideration. The PDF gives users the freedom to concentrate on particular polygon properties for a range of applications by allocating weights to these distinct aspects. When doing tasks like clustering, which involves classifying watersheds according to both spatial and non-spatial characteristics, the PDF records polygon shapes and border data. It has been useful in domains like environmental studies and regional planning, where precise polygon similarity metrics are important, and has been successfully applied in clustering techniques like k-medoids. The PDF offers a tailored way for determining polygon similarity in intricate spatial data sets [58].

In [61], a simplified method is presented for measuring spatial overlap between polygonal structures. This approach involves converting polygons into images, allowing for an overlap comparison similar to methods in image analysis. Detailed steps of this process are outlined in Sections 4.1 and 5.2. This technique shares similarities with the Jaccard Index [102, 42], which measures the similarity between two images by evaluating pixel matches: a value of zero is assigned where pixels do not align, and a value of one is assigned where they do. The final value or extent of similarity will be the proportion of the comparisons that are equal to 1.

Fan et. al [41] also makes use of polygon overlap or similarity, proposing a polygon-based approach for comparing the OpenStreetMap road network with authority data. The first step in this method is to extract urban blocks, which are important components of urban planning and are shown as polygons with roadways encircling them. The system then examines the topological relationships between these neighbourhoods to associate road lines with their edges. Finding overlapping areas between urban block polygons is the initial step in the matching process. The edges of neighbourhood block pairs are further aligned after they have been matched. After that, road lines connected to the same matched metropolitan block edges are matched. By using matching polygons rather than individual road lines, this technique significantly reduces processing costs.

Duong et. al, proposed a formal method for evaluating the similarity of spatial patterns, using a test statistic derived from kernel smoothing techniques [37]. The kernel estimator relies on the distances among points. To overcome this limitation, it was enhanced in [46] through the integration of bootstrap calibration,

which facilitates the comparison of smaller patterns as well. However, this test may not achieve a nominal significance level. The primary drawback of the method is it requires a lot of computing power, especially when analysing larger patterns. This approach is used in [21] to examine covariate measurements at various points within the point pattern.

A non-parametric test was developed by Andresen in 2009 to evaluate how similar two spatial point patterns are to one another [4]. The test uses an area-based methodology to assess similarity. The test concentrates on whether the points in the two patterns are distributed similarly rather than the statistical distribution of the points inside the spatial patterns. This test yields a percentage, known as the S-index, that measures how similar the two spatial point patterns are to one another [4]. Andresen's S-Index in criminology is used to quantify the spatial similarity or difference in crime patterns over time. It is widely used in criminology to analyse changes in the geographic distribution of crimes across different periods or between different types of crimes [4].

1.1.3 Accessibility and Criminology

Accessibility is a concept frequently used in transportation and land-use research [57]. A variety of accessibility measures have been proposed. Accessibility pertains to the opportunities afforded to an individual or specific group at a particular area to engage in a specific activity or range of activities [57]. Various interpretations of the phrase "accessibility" are in literature. This necessitates an examination of the various indices to which this term has been applied, as well as the range of definitions attributed to it. There isn't a one "best" metric; instead, the selection is based on the resources available and the nature of the problem being examined [57]. Research on police station accessibility in criminology often looks at how spatial factors affect public access to police services. A consolidation of regional police bodies into a national authority raised concerns about police service distribution in Sweden [113]. They applied central place theory and geographic information systems to analyse accessibility using proximity and network distance [113]. An interesting finding is that while most residents can reach a station within 20 minutes by car, remote populations may face over two-hour travel times for specialised services [113].

The National Institute of Justice in [39], demonstrates that spatial similarity and dissimilarity are important in crime hot spot analysis. It helps show patterns where criminal activity is more likely to occur in "hot spots". The report also pointed out "hot spots" are vulnerable because of environmental or sociological factors. Areas regarded as "cool spots" benefit from their excellent community organisation. A set of techniques, including gradient mapping, point mapping, and spatial ellipses helps them locate high-risk locations. The report uses point mapping for micro locations. The paper also discusses how different types of barriers affect spatial relationships, making accessibility to other areas difficult.

1.2 Research Objectives

The primary objective of this study is to evaluate the relationship between police precinct boundaries, accessibility to police stations, and crime reporting. In more detail, this mini-dissertation aims to

1. Develop and apply spatial methods to assess police precinct accessibility using Voronoi diagrams based on Euclidean and network distances.
2. Investigate the extent to which administrative police precinct boundaries align with accessibility-based precinct definitions derived from spatial models.
3. Quantify the degree of overlap between administrative police precinct boundaries and Voronoi-based precincts to determine the effectiveness of current police precinct delineations in terms of accessibility.
4. Assess the impact of police station accessibility on crime reporting by evaluating whether precincts with lower accessibility have lower reported crime rates.
5. Determine whether discrepancies between police precinct boundaries and accessibility-based precincts lead to crime being reported at different police stations than the one within the designated precinct.
6. Validate findings using a case study of the Khayelitsha area in the Western Cape, where crime statistics and spatial data for three police precincts will be analysed.

Having outlined the research objectives, the study seeks to answer the following research questions

1. With the increasing under-reporting of crimes to police stations in South Africa [76], is it possible to determine whether accessibility to police stations quantitatively influences this phenomenon?
2. Impact of police precinct boundaries on accessibility:
    - What effect do police precinct boundaries have on accessibility?
    - Do these boundaries influence an individual's ease of access to a police station?
    - Can precinct boundaries be evaluated in terms of their contribution to an individual's accessibility to police stations?
    - How can this relationship be quantified?
3. Precinct boundaries and crime rates:
    - How do precinct boundaries affect crime rates or crime counts?
    - If overlaps exist between Voronoi-generated boundaries and police precinct boundaries, can these overlaps be attributed to accessibility issues?

Chapter 2

Data

2.1 Map Data

This research uses data from the police precincts across South Africa to see how police district boundaries delineate the police precincts. The data set is comprised of three most common types of crime, namely violent crimes, property crimes, and sexual offenses. The data has aggregated crime counts collected each year at precinct level. Figure 2.1 shows a visual summary of the crime counts in each police precint.

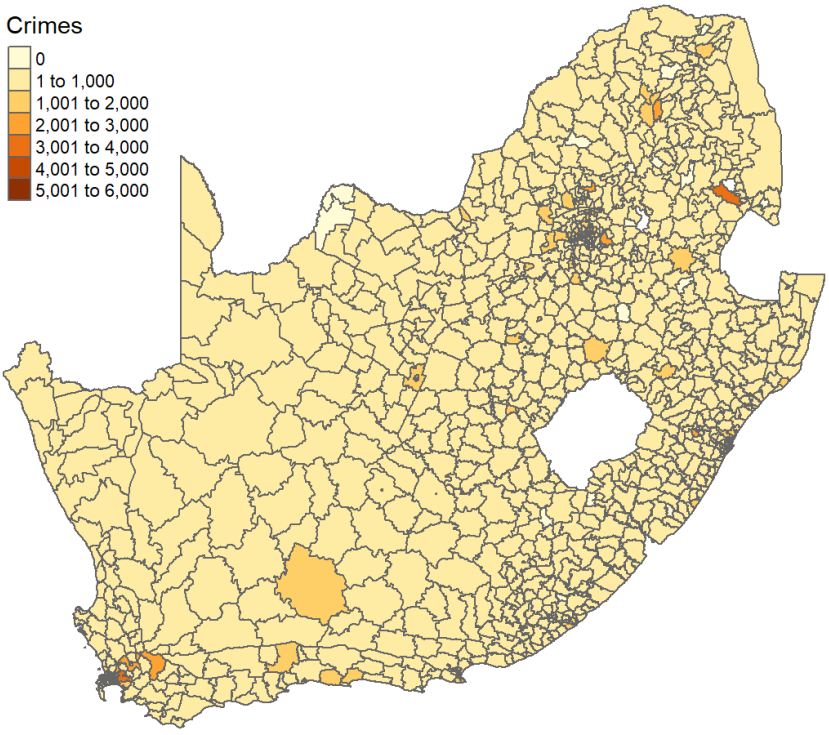
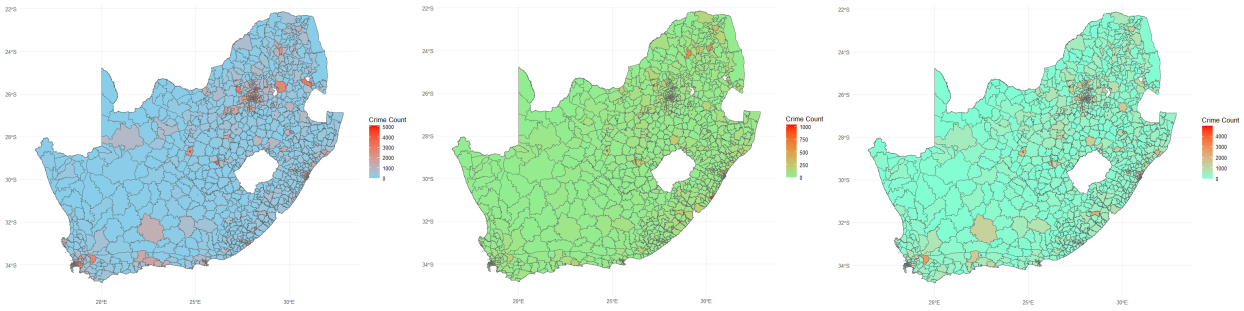


Figure 2.1: The map provides an overview of crime numbers, showing the total incidents of all crime types that occurred and were reported in each police precinct.

The maps in Figure 2.2 illustrate the prevalence of three types of crime separately. Figure 2.2b reveal that sexual offenses show less frequence than property and violent crimes. This is due to significant under reporting of the sexual crimes. Research suggests some of the reasons for this under reporting [125, 53, 105]. Victims of sexual violence may be discouraged from reporting due to close relationships with the offender, feelings of shame, guilt, or self-blame, and distrust in law enforcement. Social stigma, fear of retaliation, and concerns over confidentiality further contribute to this reluctance. Additionally, some victims may not perceive their experiences as "serious enough" if they don't align with stereotypical violent stranger assaults. In many cases, victims choose to cope independently, seeking to avoid the emotional toll of legal proceedings. The map in Figure 2.2c reveal that Hillbrow, Johannesburg Central, and Mitchells Plain had consistently high levels of violent crime. Figure 2.2a Cape Town Central, Park Road, and Honeydew had the highest rates of property crime. These places are flagged by the intense red colour in the respective maps.



(a) Property Crimes (b) Sexual Crimes (c) Violent Crimes

Figure 2.2: Overview of crime types: (a) property crimes, (b) sexual crimes, and (c) violent crimes, each shown with a corresponding map for the selected crime category.

2.2 Road Network Data

2.2.1 Network Data Preparation

Road network data is extracted from OpenStreetMap using the `osmextract` package in R [48]. The selection of a specific data set, such as the road network in South Africa, is done using the `oe_match()` function[1]. Preprocessing steps are undertaken to ready the data for analysis. This includes an initial inspection and cropping using `GIS` tools in `QGIS` [100], followed by further transformations in `R`.

2.2.2 Handling Planar and One-layer Networks

Road network data is typically modeled as a planar graph [112], meaning that edges (roads) do not cross each other without a node (intersection). One-layer networks, applied to road systems, assume that all

[1]Subsequently, the data is converted from its original format (.osm.pbf) to a more manageable format (.gpkg).

connections occur on a single plane. This is similar to the real-world layout where roads meet at defined intersections. In this context, it is important to handle road overlaps appropriately. There is need to differentiate between roads that intersect at a junction and those that pass over or under each other without a connection (e.g., bridges or tunnels). These relationships are managed by refining the data into a topologically sound network where intersections represent true connectivity [14].

The data is imported into `R` as `sf` objects [93], and the projection is adjusted to a suitable coordinate reference system (CRS), ensuring spatial accuracy across the region of interest. For instance, the UTM Zone 35S (EPSG code 32736) is applied for the Gauteng region to match local spatial reference standards. Road types are filtered to retain only relevant categories, such as motorways, primary roads, and other significant pathways, by filtering based on the attributes. This step ensures that only parts of the network which are most likely to influence traffic flow and connectivity, are retained [75].

2.2.3 Network Construction and Cleaning

We construct an undirected weighted graph where road segments are treated as edges, intersections as nodes, and the distance between intersections, as weights. The weights are derived from the spatial geometry of the road segments. This ensures that each road's length is accurately reflected in the network model [35]. The network is cleaned before analysis. The cleaning steps include simplifying the network by removing redundant edges and loops, which arise due to mapping errors. Edges are subdivided to ensure that nodes are created at all genuine intersections, to improve the accuracy of the network representation. Pseudo nodes, or nodes that do not represent true intersections, are smoothed out by eliminating unnecessary points. In cases where intersections are overly dense or geographically close, they are simplified by clustering these nodes, thus consolidating them into single intersection points [47].

The cleaned network is then visually inspected to confirm its spatial accuracy. This involves overlaying the road network with administrative boundaries, such as ward boundaries, to verify that the network coincides with the correct geographic regions. Additionally, the connectivity of the network is checked to ensure that all road segments are part of a fully connected graph. Any disconnected segments, are identified and discarded, leaving only the connected components for further analysis [12].

2.2.4 Exploratory Data Analysis

The network is composed of (line) road segments. In our setting, the network contains 311 632 road segments in Gauteng, South Africa, represented as a Simple Feature Collection in the WGS 84 / UTM zone 36S spatial reference system. This data set provides a detailed view of road segments, each represented by multiple fields detailing the road's characteristics.

Table 2.1 shows the first 10 rows of the data set. Each road segment includes a unique identifier `osm_id` from OpenStreetMap[2] and a `code` field, that encodes different types of roads. The `fclass` field groups the road segments into general categories, such as primary, secondary, and motorway roads. The grouping is essential for understanding the functional purpose of each road within each precinct. When available, the `name` field provides the official name of a road segment. The `ref` field has the route numbers or road identifiers, such as "R24".

Table 2.1: Sample data from Gauteng road network.

osm_id	code	fclass	name	ref	oneway	maxspeed	layer	bridge	tunnel
4339842	5113	primary	Kitchener Ave	R24	B	80	0	F	F
11624565	5114	secondary	Nicol Rd	M97	B	60	0	F	F
11624875	5114	secondary	Queen St	M33	B	60	0	F	F
11625221	5115	tertiary	Bradford Rd	-	B	60	0	F	F
11633680	5122	residential	Kloof St	-	B	30	0	F	F
11638702	5111	motorway	Albert Sisulu Rd	R24	F	120	0	F	F
11640047	5115	tertiary	Boeing Rd W	-	B	60	0	F	F
11640227	5135	tertiary link	-	-	F	40	0	F	F
11640298	5131	motorway link	-	-	F	80	0	F	F
11640394	5135	tertiary link	-	-	F	40	0	F	F

To further examine traffic flow and road functionality, the data set includes fields such as `oneway`, indicating if the road segment is unidirectional or bidirectional. The `maxspeed` specifies speed limits where available. We may learn a lot about possible traffic flow and congestion patterns by looking at these characteristics, particularly in places with dense populations and large traffic volumes. Variables like `layer`, `bridge`, and `tunnel` show road network aspects such as elevated or underground roadways and vertical stacking in regions, or if that segment has a bridge or a tunnel.

The road classifications in the Gauteng region are shown in Figures 2.3 and 2.4. A wide range of classifications are included in the entire network shown in Figure 2.3. Along with main roads that support high traffic flow, such as "primary", "secondary", "tertiary", and "motorway" highways, the data set includes classifications that are appropriate for local and specialised usage. These include "residential" roads, which are mostly used for access to nearby neighbourhoods, and "service" roads, which often have access to particular buildings, such as police stations, which are a major focus of our study.

[2] `https://www.openstreetmap.org`

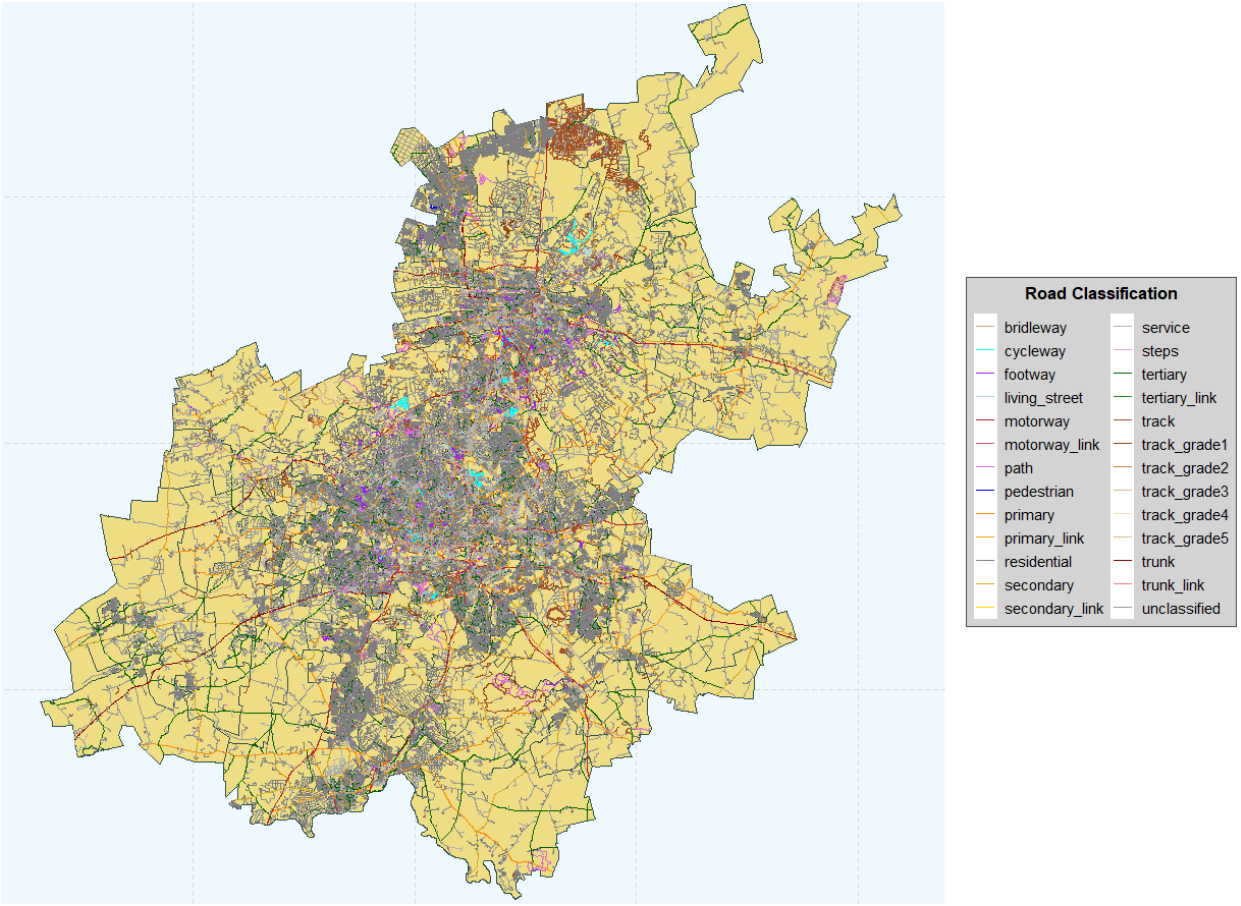


Figure 2.3: Gauteng road network with road classifications.

A primary focus of this mini-dissertaion is to quantify accessibility to police stations, which play a critical role in public safety. Therefore, understanding the distribution and connectivity of roads serving the stations is essential. Other classifications, such as "unclassified" roads, refer to smaller routes that don't fall into one of the established categories. The fact that "footways" and "paths" are just for pedestrians may affect how easily pedestrians can reach the police stations. Prioritising non-vehicular use through cycleways and "living streets" might affect accessibility to police stations in different precincts.

The subset of roads, in Figure 2.3, details additional specialised paths, including "track" classifications from "track grade1" to "track grade5", which may represent unpaved paths of varying conditions, these are often used for agricultural or recreational purposes but potentially relevant in understanding indirect access routes to police stations in rural areas. "Bridleways" denote paths designated for horseback riding. Moreover, the data set includes "pedestrian" zones and "steps", exclusively for foot traffic, and "trunk" roads and "trunk links", which facilitate major traffic movement between high-capacity routes. The primary and secondary routes but also reflects accessibility and safety by supporting diverse modes of transportation, including routes that connect directly or indirectly to police stations.

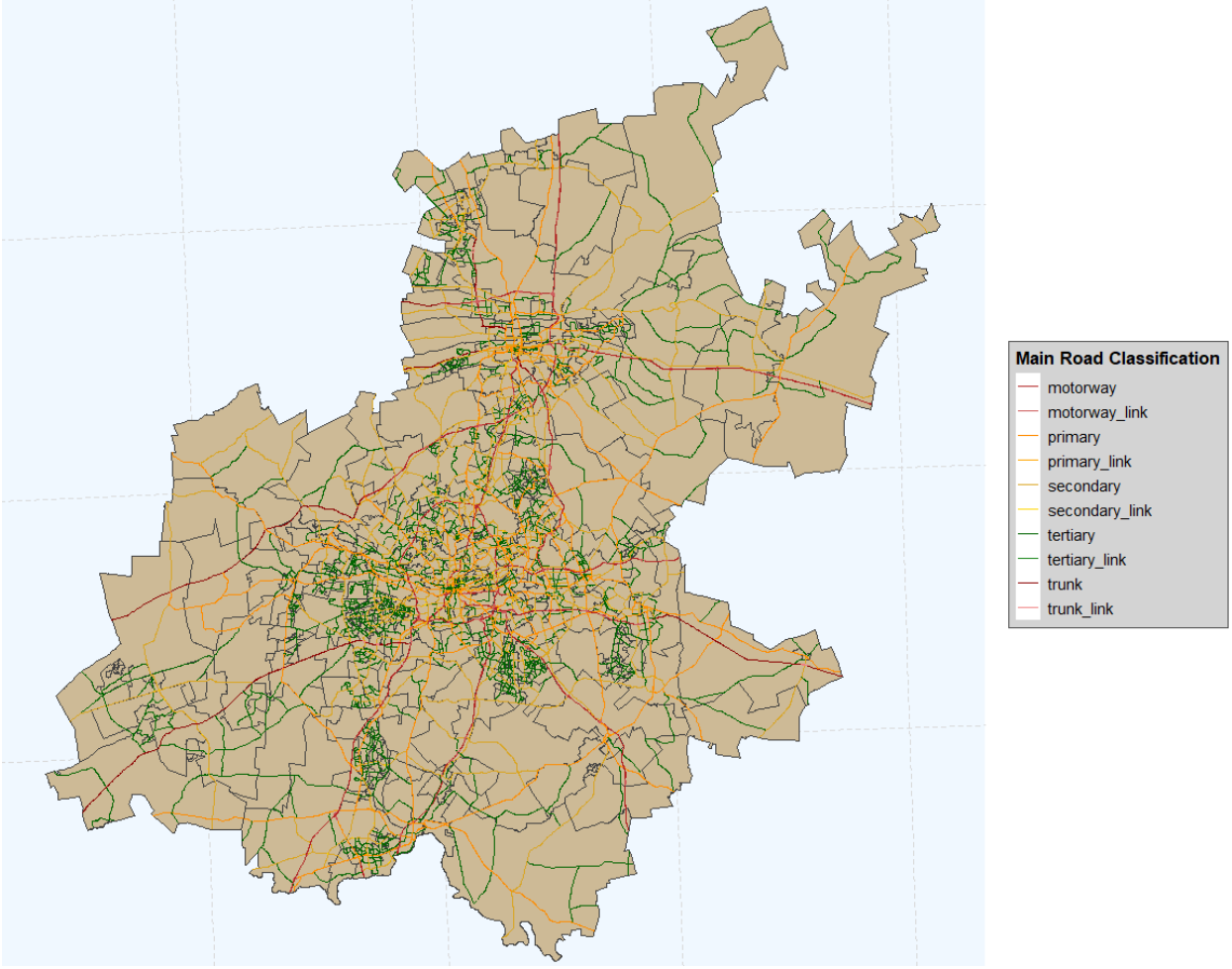


Figure 2.4: Subset of Gauteng road network focusing on fewer road types.

2.3 Crime Point Data (Validation Data)

The third data set serves as a validation tool for the spatial similarity test results presented in Chapter 5. It is based on three precincts in the Khayelitsha area of the Western Cape, South Africa. The data set contains precise coordinates, the type of offense or crime, and the time of occurrence for each reported incident. Spanning the years 2006 to 2016, it includes various types of crimes reported within the precincts. Table 2.2 displays a sample of rows from this dataset.

Table 2.2: The first and last 5 rows from the crime count data set.

Incident	X_Coord	Y_Coord	Offence	Year	Month	Day	Weekday	SatSun	Hour	Minute
1	18.6381	-34.0144	Abduction	2006	4	28	6	0	9	0
2	18.6538	-34.0136	Abduction	2006	5	15	2	0	8	0
3	18.6394	-34.0142	Abduction	2006	5	19	6	0	19	0
4	18.6469	-34.0140	Abduction	2006	5	30	3	0	20	5
5	18.6572	-34.0159	Abduction	2006	6	1	5	0	6	0
4949	18.6772	-34.0239	Sexual Assault	2015	11	19	5	0	8	30
5719	18.6786	-34.0334	Sexual Assault	2016	2	13	7	1	12	0
5857	18.6537	-34.0258	Sexual Assault	2016	2	29	2	0	18	30
2877	18.6565	-34.0219	Sexual Exploitation Of Mentally Disabled Person	2008	2	8	6	0	13	0
2878	18.6744	-34.0269	Sexual Grooming Of Mentally Disabled Person	2011	10	2	1	1	22	0

Figure 2.5 shows the distribution of the crime point pattern.

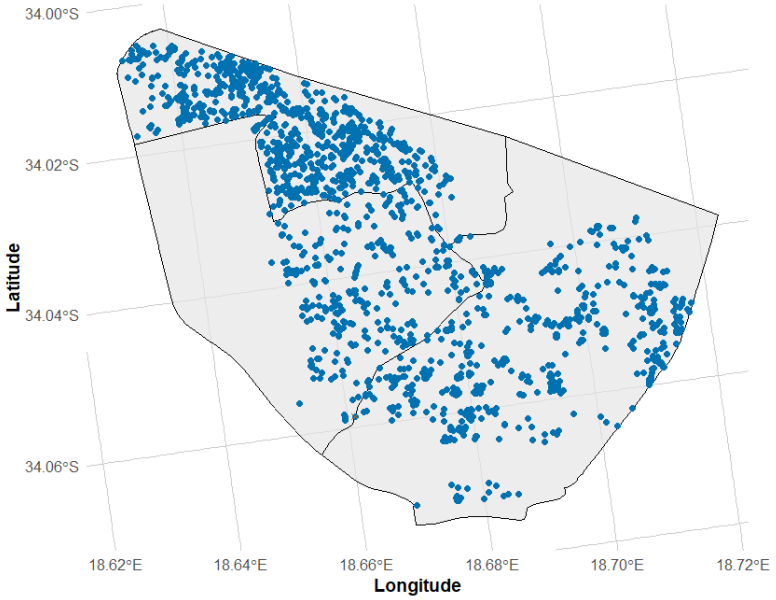


Figure 2.5: Map of the three precincts in the Khayelitsha area, showing the point pattern distribution of reported crime incidents.

Table 2.3 has the summary statistics of the crimes per police precinct.

Table 2.3: Crime categories and totals by police precinct.

Location	Category	Total Crimes	Total per Location
Harare	Other Crimes	335	723
	Property Crimes	135	
	Sexual Crimes	33	
	Violent Crimes	220	
Khayelitsha	Other Crimes	365	780
	Property Crimes	112	
	Sexual Crimes	30	
	Violent Crimes	273	
Lingelethu	Other Crimes	189	495
	Property Crimes	64	
	Sexual Crimes	18	
	Violent Crimes	224	

2.4 Conclusion

This chapter described the data sets used in this study. The map Data provided the police stations' spatial structure, while the road network data was cleaned and prepared to ensure accurate results. Processes like handling planar and one-layer networks were important for building a reliable network. Finally, the crime point data was briefly introduced as a validation tool. These data sets together provide a strong base for the analyses in the next chapters.

Chapter 3

Methodology

This section explores the theoretical background of Voronoi diagrams, including the algorithms used for their creation. Additionally, it examines the foundational concepts of network Voronoi diagrams. The properties and characteristics of Voronoi diagrams relevant to the application Chapter 5 are also described. Finally, we introduce a novel approach for generating polygon boundaries in network Voronoi diagrams.

3.1 The Traditional Voronoi Diagram

Assume a set of distinct points in the Euclidean plane, where the number of points is finite and at least two. The trivial case of a single-point set is disregarded, as its Voronoi diagram consists solely of the entire plane as one Voronoi polygon. Each point in this set is unique, meaning no two points overlap. In a Voronoi diagram, every location in the plane is assigned to the nearest point in the set. If a location is equidistant to two or more points, it is considered part of all those points' regions. Consequently, each point in the set defines a unique region consisting of locations closest to it. These regions collectively cover the entire plane, ensuring every location is assigned to at least one point [17]. The boundaries between regions consist of locations equidistant to the adjacent points, and these boundaries are shared by the neighbouring regions. In Voronoi diagrams, this partitioning of the plane into distinct regions, or Voronoi polygons, forms a tessellation [1]. Apart from the boundaries, the regions create a complete tessellation of the plane, known as a planar Voronoi diagram [115]. The regions themselves are referred to as Voronoi polygons. Voronoi polygons can be thought of as cells in a growth process. We will refer to Figure 3.1 to illustrate the definitions described above. The set of distinct points $\{q_1, q_2, \ldots, q_5\}$, represented by solid blue circles, are clearly shown in the diagram. Notice that none of the points coincide. Each location in the plane is assigned to its nearest point from this set.

[1] A tessellation is when one or more geometric shapes, referred to as tiles, are used to cover a surface without any overlap or spaces between them.

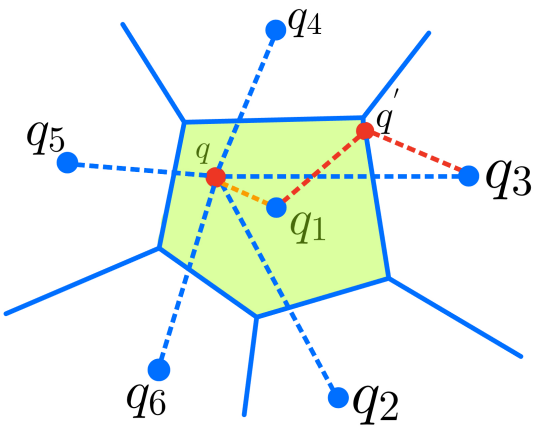


Figure 3.1: An example of a traditional Voronoi diagram polygon construction.

The point q is assigned to the solid red circle. This red circle lies on the orange dashed line. Additionally, there are locations equidistant from two points in the set (formally introduced shortly); as a result, these locations are assigned to both points. In the diagram, point q' is assigned to the two blue circles q_1 and q_3 along the red dashed lines. Lastly, the collection of locations assigned to each point forms a distinct region. In the diagram, this is represented by the shaded green region. Note that the solid blue lines mark the boundaries between regions, which are formed by locations equidistant from two or more points in the set. We define the Voronoi diagram formally next.

3.1.1 Mathematical Definition of a traditional Voronoi Diagram

Consider a finite set $Q = \{q_1, q_2, \ldots, q_m\}$ of m distinct points in the Euclidean plane, where $2 \leq m < \infty$. Each point q_i has Cartesian coordinates (y_{i1}, y_{i2}) or, equivalently, a location vector $\mathbf{y}_i$. For any two points q_i and q_j in Q, we have $\mathbf{y}_i \neq \mathbf{y}_j$ for $i \neq j$.

Given an arbitrary point q in the Euclidean plane with coordinates (y_1, y_2) or location vector $\mathbf{y}$, the Euclidean distance between q and q_i is defined as

$$d(q, q_i) = \|\mathbf{y} - \mathbf{y}_i\| = \sqrt{(y_1 - y_{i1})^2 + (y_2 - y_{i2})^2}.$$

If q_i is the closest point to q or one of the closest points, it satisfies

$$\|\mathbf{y} - \mathbf{y}_i\| \leq \|\mathbf{y} - \mathbf{y}_j\| \text{ for all } j \neq i.$$

Mathematically [17, 7], let $Q = \{q_1, \ldots, q_m\} \subset \mathbb{R}^2$, where $2 \leq m < \infty$ and $\mathbf{y}_i \neq \mathbf{y}_j$ for $i \neq j$. The Voronoi

polygon associated with q_i is the region defined as:

$$V(q_i) = \left\{ \mathbf{y} \in \mathbb{R}^2 \mid \|\mathbf{y} - \mathbf{y}_i\| \leq \|\mathbf{y} - \mathbf{y}_j\| \text{ for all } j \neq i \right\}. \tag{3.1}$$

The collection of these regions [84],

$$\mathscr{V} = \{V(q_1), \ldots, V(q_m)\},$$

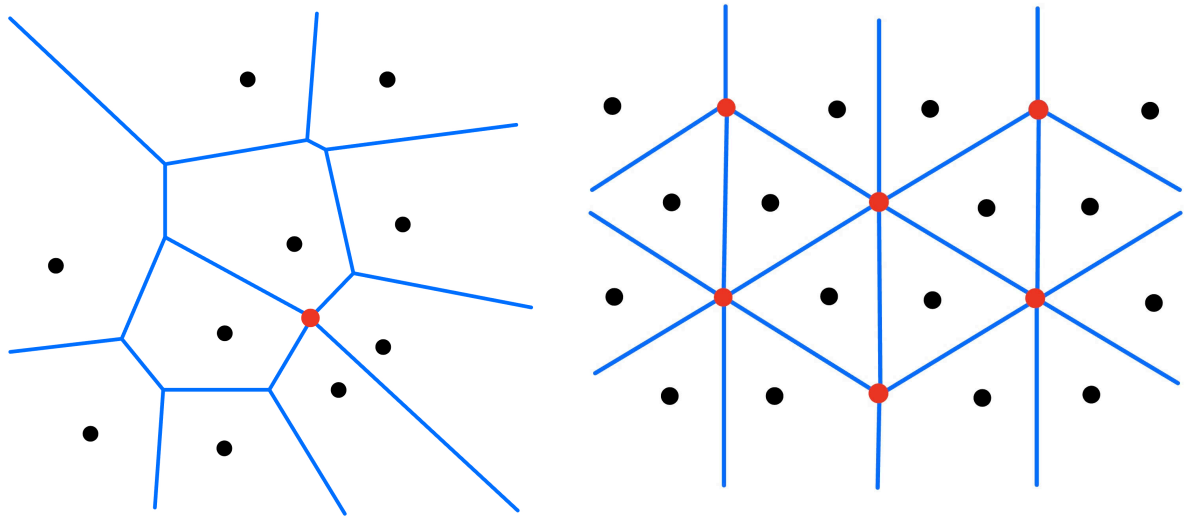
constitutes the planar Voronoi diagram generated by Q. The point q_i is referred to as the generator point of the i-th Voronoi polygon $V(q_i)$, and the set Q is called the generator set of the Voronoi diagram V.

Voronoi polygons can be conceptualised as either closed or open sets; however, for our purposes, we will assume they are closed sets. Since a Voronoi polygon is a closed set, it includes its boundary, denoted by $\partial(V(q_i))$.

A Voronoi diagram can alternatively be described by the union of its edges [17], expressed as $\bigcup_{i=1}^{m} \partial(V(q_i))$, rather than by the set of polygons $\{V(q_1), \ldots, V(q_m)\}$. The definition includes the boundaries, so a Voronoi edge may take the form of a line segment, half-line, or infinite line shared between two adjacent Voronoi polygons, along with its endpoints. Formally, if $V(q_i) \cap V(q_j) \neq \emptyset$, the intersection $V(q_i) \cap V(q_j)$ forms a Voronoi edge. We denote this edge as $v_e(q_i, q_j)$, representing the edge generated by points q_i and q_j. Although the edge could be empty, if it is neither empty nor a single point, the polygons $V(q_i)$ and $V(q_j)$ are considered adjacent [17]. A Voronoi vertex is defined as a point where multiple Voronoi edges meet, or equivalently, as a point shared by three or more Voronoi polygons [17].

In the context of Voronoi diagrams, a diagram V is considered degenerate if there exists at least one vertex where four or more Voronoi edges converge, as indicated by the red circle(s) in Figure 3.2. On the other hand, if all vertices are formed by exactly three converging edges, as shown in Figure 3.1, the diagram is classified as non-degenerate [17].

For instance, the Voronoi diagrams in Figure 3.2 are examples of degeneracy, whereas Figure 3.1 illustrates a non-degenerate case. Degeneracy typically arises when generator points are regularly distributed, as shown in Figure 3.2b. The concept of degeneracy is important in the construction of Voronoi diagrams because it affects the stability and uniqueness of the diagram. Non-degenerate diagrams generally ensure a more stable and unique tessellation, which is essential in applications requiring precise spatial partitioning [17].



(a) Voronoi diagram with irregularly spaced generator points. (b) Voronoi diagram with regularly spaced generator points.

Figure 3.2: Comparison of degenerate Voronoi diagrams with different distributions of generator points showing a Voronoi diagram with irregularly spaced generator points, resulting in cells of varying sizes and shapes, and a Voronoi diagram with regularly spaced generator points, producing a more uniform cell structure.

3.1.2 Half-Plane Representation of Voronoi Diagrams

As illustrated in Figure 3.1, the traditional Voronoi diagram consists of polygons. Since polygons can be described by half-planes [111], we can alternatively represent a Voronoi diagram in terms of half-planes. A half-plane in Voronoi diagrams is one side of the plane divided by the perpendicular bisector of two points or sites. It represents the region closer to one site than the other [40]. We need the half-plane alternative to get a deeper understanding of the characteristics of the Voronoi diagrams. To formulate this alternative representation, consider the perpendicular bisector of the line segment $\overline{q_i q_j}$ connecting two generator points q_i and q_j. In Figure 3.3, this is illustrated for generator points q_1 and q_2 (for $i = 1$ and $j = 2$) as the thick blue line. This bisector, denoted as $b(q_i, q_j)$, describes the set of points $\mathbf{y}$ that are equally distant from both generator points $\mathbf{y}_i$ and $\mathbf{y}_j$ (corresponding to q_i and q_j), and can be expressed as

$$b(q_i, q_j) = \{\mathbf{y} \mid \|\mathbf{y} - \mathbf{y}_i\| = \|\mathbf{y} - \mathbf{y}_j\|\}.$$

This bisector $b(q_i, q_j)$ splits the plane into two distinct half-planes. The half-plane containing q_i is defined as

$$H(q_i, q_j) = \{\mathbf{y} \mid \|\mathbf{y} - \mathbf{y}_i\| \leq \|\mathbf{y} - \mathbf{y}_j\|\},$$

which is known as the dominance region of q_i over q_j.

In Figure 3.3, the dominance regions of q_1 over q_2, q_3, and q_4 are highlighted in red, green, and blue, respectively. Inside the dominance region $H(q_i, q_j)$, the distance to q_i is consistently less than or equal to the distance to q_j.

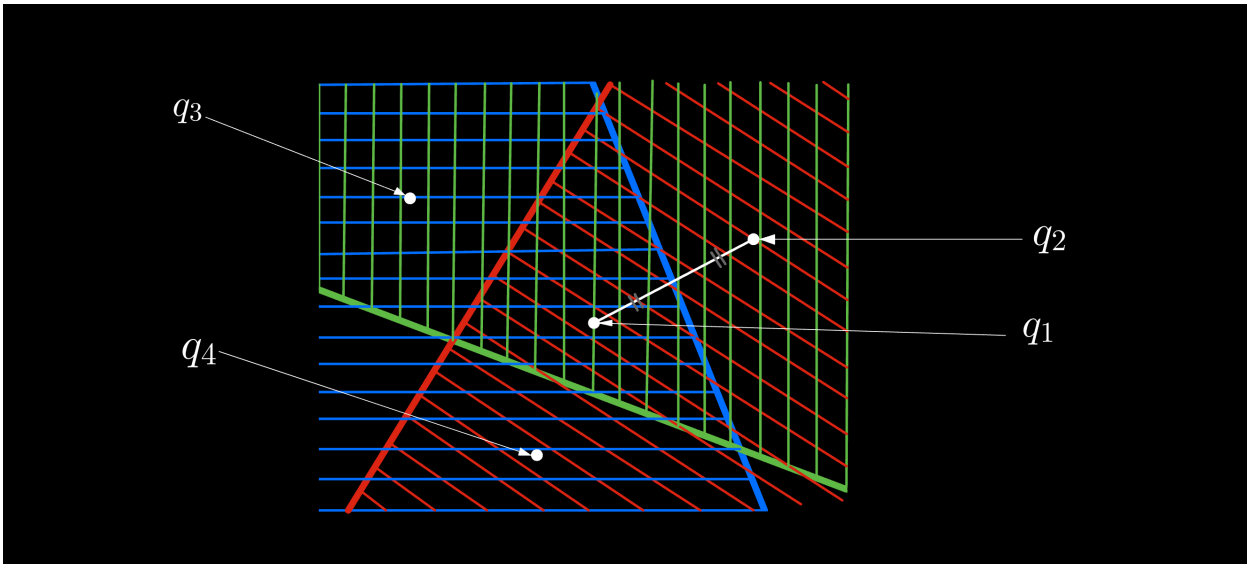


Figure 3.3: Visualising the Voronoi polygon formation through the intersection of half-planes.

Therefore, in Figure 3.3, any point q within the intersection of these different colour regions is closer to q_1 than to q_2, q_3, or q_4. This relationship is mathematically represented as:

$$H(q_1, q_2) \cap H(q_1, q_3) \cap H(q_1, q_4),$$

which corresponds to the Voronoi polygon associated with q_1. Based on this, the original distance-based definition can be alternatively expressed using the half-plane definition.

Let $Q = \{q_1, \ldots, q_m\} \subset \mathbb{R}^2$, where $2 \leq m < \infty$ and $\mathbf{y}_i \neq \mathbf{y}_j$ for $i \neq j$, with $i, j \in I_m$, where $I_m = \{1, 2, \ldots, m\}$. The region defined as

$$V(q_i) = \bigcap_{j \in I_m \setminus \{i\}} H(q_i, q_j) \tag{3.2}$$

represents the Voronoi polygon associated with point q_i, where $H(q_i, q_j)$ is the half-plane containing q_i and bounded by the bisector between q_i and q_j is called the Voronoi polygon associated with q_i. The set $\mathscr{V}(Q) = \{V(q_1), \ldots, V(q_m)\}$ constitutes the traditional Voronoi diagram generated by Q. The equivalence between this half-plane-based definition and the earlier distance-based definition is evident, as $\|\mathbf{y} - \mathbf{y}_i\| \leq \|\mathbf{y} - \mathbf{y}_j\|$ holds if and only if $\mathbf{y} \in H(q_i, q_j)$ for all $j \neq i$ [18].

3.1.3 Fundamental Characteristics of Voronoi Diagrams

Having established the concept of a Voronoi diagram, we now turn our attention to examining some of their fundamental characteristics from [18] that we base the creation of the Voronoi diagrams in our methodolody on. While our focus will primarily be on planar Voronoi diagrams, many of these characteristics can be extended to higher dimensions.

The region $V(q_i)$ is defined and described using Equation (3.1), which notes that it forms a polygon. This can be more intuitively understood via the other definition, which constructs a Voronoi polygon through the intersection of half-planes using Equation (3.2). Since half-planes are convex and the intersection of convex sets remains convex [89], each Voronoi polygon is a convex set. Moreover, because the generator points are distinct, each Voronoi polygon is non-empty [19].

Every point in $\mathbb{R}^2$ is assigned to at least one generator point according to the definition in Equation (3.1), making the Voronoi polygons in $\mathscr{V}(Q)$ collectively exhaustive over $\mathbb{R}^2$. Furthermore, since $V(q_i) \subset H(q_i, q_j)$ and $V(q_j) \subset H(q_j, q_i)$ with $H(q_i, q_j) \cap H(q_j, q_i) = \emptyset$ (excluding boundaries), Voronoi polygons are mutually exclusive except at their boundaries [18]. Although these characteristics are straightforward for traditional Voronoi diagrams, they may not hold for the network Voronoi diagrams, which will be discussed in Sections 3.2 and 3.3. To highlight these basic characteristics, we summarise them as follows.

Characteristic 1. Let $Q = \{q_1, \ldots, q_m\} \subset \mathbb{R}^2$ where $2 \leq m < \infty$ and the points are distinct. The set $V(q_i)$ defined as

$$V(q_i) = \{y \mid \|y - y_i\| \leq \|y - y_j\| \text{ for all } j \neq i, j \in I_m\}$$

is a non-empty convex polygon. Furthermore, the collection $\mathscr{V}(Q) = \{V(q_1), \ldots, V(q_m)\}$ satisfies

$$\bigcup_{i=1}^{m} V(q_i) = \mathbb{R}^2,$$

$$[V(q_i) \setminus \partial V(q_i)] \cap [V(q_j) \setminus \partial V(q_j)] = \emptyset \quad \text{for} \quad i \neq j, \quad i, j \in I_m.$$

Thus, the Voronoi diagram $V(Q)$ represents a unique partitioning of $\mathbb{R}^2$ for the set Q.

Given that Voronoi polygons collectively cover the Euclidean plane, at least one Voronoi polygon must be unbounded. With the condition $m \geq 2$, there are at least two Voronoi polygons, some of which may be bounded. When the generator points are collinear[2], all Voronoi polygons are unbounded. For non-collinear generator points, we state Characteristic 2.

Characteristic 2. Consider the Voronoi diagram generated by a set of distinct points $Q = \{q_1, \ldots, q_m\} \subset \mathbb{R}^2$, where $2 \leq m < \infty$. The Voronoi polygon $V(q_i)$ is unbounded if and only if the point q_i lies on the

[2] In geometry, a set of points is said to be collinear if they all lie on the same straight line. This property is known as collinearity [22]

boundary of the convex hull of Q, denoted as $q_i \in \partial(\text{Conv}(Q))$. Here, $\text{Conv}(Q)$ represents the convex hull of Q, and ∂ denotes the boundary of the convex hull.

The Voronoi diagram is defined for Q, and the Voronoi polygon $V(q_i)$ corresponding to a point $q_i \in Q$ is unbounded if and only if q_i is located on the boundary of the convex hull of Q, denoted as $\partial(\text{Conv}(Q))$. The proof of this property is thoroughly presented in [19]. Given the Voronoi diagram $V(Q)$, one can determine the convex hull of the point set Q by applying Characteristic 2. Specifically, the convex hull $\text{Conv}(Q)$ is formed by connecting the generator points corresponding to unbounded Voronoi polygons that share a common boundary. From Characteristic 2, the following is derived.

Characteristic 3. For the Voronoi diagram generated by a set of distinct points $Q = \{q_1, \ldots, q_m\} \subset \mathbb{R}^2$ where $2 \leq m < \infty$:

1. The Voronoi edges are infinite lines if and only if the points in P are collinear.
2. The Voronoi edge $v_e(q_i, q_j) \neq \emptyset$ is a half-line if and only if Q is non-collinear and q_i and q_j are consecutive points on the boundary of $\text{Conv}(Q)$.
3. The Voronoi edge $v_e(q_i, q_j)$ is a finite line segment if and only if the segment $\overline{q_i q_j}$ does not form an edge of $\text{Conv}(Q)$.

It should be noted that while Voronoi edges are segments of bisectors, not every bisector corresponds to a Voronoi edge. The following characteristics provide sufficient conditions for the creation of these Voronoi edges [19].

Characteristic 4. The Voronoi edge corresponding to the Voronoi cell $V(q_i)$ is uniquely defined by the generator point that is closest to q_i. The proof of this characteristic is elegantly demonstrated in [19].

Characteristic 5. The nearest generator point to q_i is among the set of generator points whose Voronoi polygons have a common Voronoi edge with $V(q_i)$.

Characteristic 6. The generator point q_i is the closest generator to a point q if and only if q is located within the Voronoi polygon $V(q_i)$.

While characteristic 6 is a fundamental property of a Voronoi polygon, it is explicitly stated here. This is because it has practical applications in addressing the problem of creating network Voronoi diagram polygons, as discussed in Section 3.3. Characteristics 7 to 10 are provided to facilitate a sanity check on the shapes of the resulting Voronoi polygons. Detailed proofs for these properties can be found in [19].

Characteristic 7. In the Voronoi diagram within $\mathbb{R}^2$, m, m_e, and m_v represent the number of generator points, Voronoi edges, and Voronoi vertices, respectively, where $2 \leq m < \infty$. The relationship among these quantities is given by:

$$m_v + m = m_e + 1.$$

Characteristic 8. In the n-dimensional Voronoi diagram, m_k denotes the number of k-dimensional Voronoi faces. The relationship among these quantities is given by

$$\sum_{k=0}^{n}(-1)^k m_k = (-1)^n.$$

Characteristic 9. In the Voronoi diagram in $\mathbb{R}^2$, m, m_e, and m_v represent the number of generator points, Voronoi edges, and Voronoi vertices, respectively, with $3 \leq m < \infty$. The following relations hold:

$$m_e \leq 3m - 6,$$

$$m_v \leq 2m - 5.$$

The Voronoi polygon is defined by the intersection of $m-1$ half-planes. Consequently, the Voronoi polygon can have at most $m-1$ Voronoi edges.

Characteristic 10. The average number of edges per Voronoi polygon is at most six.

3.1.4 Practical construction methods for traditional Voronoi diagrams

Having considered the theory and characteristics of Voronoi diagrams, practical approaches for their construction are examined, with a primary focus on the fundamental case of points in a plane. Algorithm 1 introduces a straightforward technique based directly on the definition of Voronoi diagrams [20]. The algorithm iteratively computes the half-planes between each generator point and all others, then finds the intersection of these half-planes to form the Voronoi polygons. This process is repeated for each generator, constructing the full Voronoi diagram step by step. After generating the polygons, the set of polygons is combined to produce the complete Voronoi diagram. This basic method, while inherently inefficient, is grounded in several equivalent conditions for characterising the Voronoi diagram, including the half-plane characterisation. This characterisation provides a method where the Voronoi polygon for a generator q_i is determined by the intersection of all half-planes defined by the perpendicular bisectors of q_i and the other generators. Utilising this definition, the Voronoi polygons are sequentially constructed.

Algorithm 1 Basic Voronoi diagram construction.

Input: A set of m generators $Q = \{q_1, q_2, \ldots, q_m\}$
Output: The Voronoi diagram $V = \{V(q_1), V(q_2), \ldots, V(q_m)\}$
Create an empty set V to store the Voronoi polygons
for each generator q_i (where $i = 1, 2, \ldots, m$) **do**
    Generate half-planes:
    for each j (where $j = 1, 2, \ldots, m$ and $j \neq i$) **do**
        Compute the half-plane $H(q_i, q_j)$ using the perpendicular bisector of q_i and q_j
    end for
    Compute intersection:
    Determine the intersection of all half-planes $H(q_i, q_j)$ to obtain the Voronoi polygon $V(q_i)$
    Store the result:
    Add $V(q_i)$ to the set V
end for
Output the Voronoi diagram:
The set V containing all Voronoi polygons $\{V(q_1), V(q_2), \ldots, V(q_m)\}$

The method described in Algorithm 1 is among the simplest to grasp [103] as it directly translates the definition of the Voronoi diagram into practice. In theory, a Voronoi diagram should result in a tessellation of the plane, where the entire plane is divided into non-overlapping polygons, each associated with a generator point. This tessellation ensures that every point in the plane belongs to exactly one polygon. However, this straightforward method may not always produce a true tessellation [20]. Factors such as numerical precision and floating-point errors can lead to slight overlaps or gaps between polygons, preventing a perfect division of the plane [99]. Consequently, when implemented directly in a computer program, the method can be prone to numerical errors, potentially leading to topological inconsistencies [99]. To ensure numerical robustness, several algorithms have been developed over the years to address these issues effectively.

- **Divide and conquer [60, 68, 38, 52, 108]:** This algorithm starts by recursively splitting the point set into smaller subsets based on their coordinates. Once the subsets are small enough, the Voronoi diagram is constructed for subset each individually. The challenge lies in merging these smaller diagrams to form the overall Voronoi diagram. This merging step combines the results by maintaining the geometric properties of the Voronoi diagram. The method is highly efficient, achieving a time complexity of $\mathcal{O}(n \log n)$, where n is the total number of points. It is suitable for large data sets due to its scalability and efficiency.

- **Incremental [64, 51, 80]:** In this approach, the Voronoi diagram is constructed incrementally by adding one generator point at a time. Each time a new point is added, the existing diagram is updated to include the new point's influence, which may alter the structure of the diagram. Although the naive implementation of this method has a quadratic time complexity of $\mathcal{O}(n^2)$, which can be slow for large data sets, optimised versions improve this to $\mathcal{O}(n \log n)$ by using advanced data structures and more efficient updating strategies. This method is intuitive and relatively straightforward to implement, making it a popular choice for smaller or moderately sized data sets.

- **Plane sweep [44]:** The plane sweep algorithm processes points by conceptually sweeping a line (usually vertical) across the plane. As the line moves, it encounters events, such as a point being added or edges of the Voronoi diagram forming. The algorithm keeps track of active events in a structured manner, allowing it to build the diagram dynamically as the line progresses. This method is efficient and has a time complexity of $\mathcal{O}(n \log n)$, making it well-suited for both theoretical and practical applications where computational efficiency is critical.

We begin by exploring the **incremental algorithm**, with particular attention to the iterative approach detailed in [67]. Since this algorithm is closely tied to Delaunay triangulation, we will first introduce the fundamental concepts of the Delaunay triangulation before going into the algorithm itself. We want to address the problem of triangulating a set of points in the plane. Let Q be a set of m distinct points in the Euclidean plane, where $m \geqslant 3$, and assume the points are not collinear. Define V_E as the set of edges formed by line segments connecting pairs of vertices in Q. Two edges v_{e1} and $v_{e2} \in V_E$, where $v_{e1} \neq v_{e2}$, are said to properly intersect if they meet at a point other than their endpoints [71]. The straight-line graph in $\mathbb{R}^2$, $G(Q, V'_E)$, is a triangulation of Q where E' is the maximal subset of V_E such that no two edges in V'_E properly intersect [24].

In Figure 3.4 we observe the Delaunay triangulation alongside its corresponding Voronoi diagram, based on a set of 16 generator points (orange circles). The Delaunay triangulation (red dashed lines) is a type of triangulation that can be created by connecting points with edges, ensuring that no new edge intersects an existing edge.

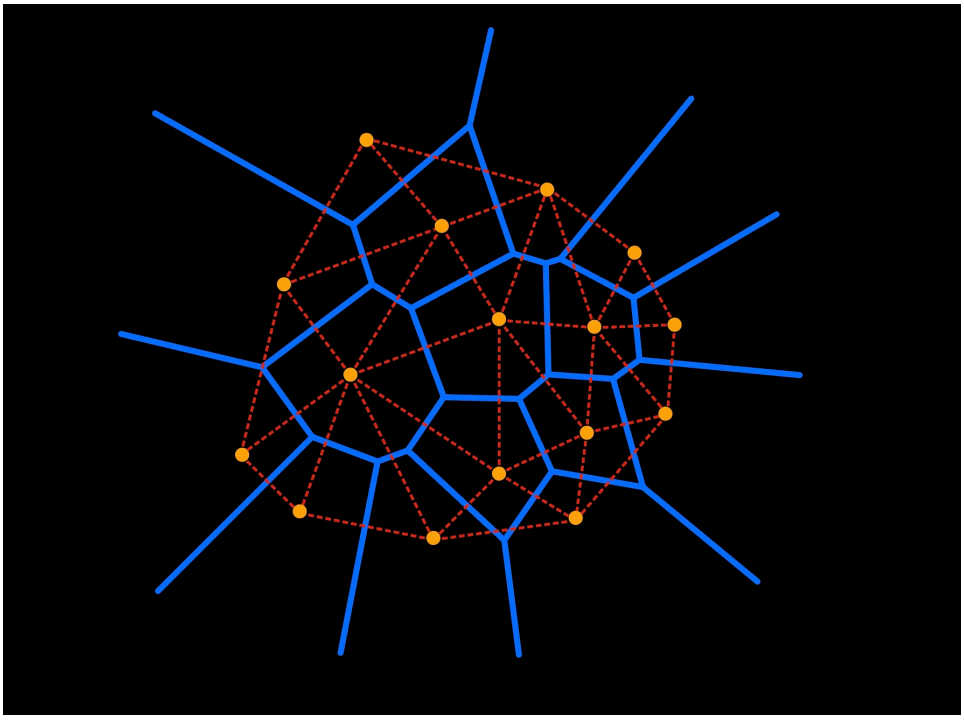


Figure 3.4: An illustration of the Delaunay triangulation shown by red lines and the corresponding Voronoi diagram represented by blue lines, based on 16 generator points (denoted by orange circles).

This specific triangulation type has the characteristic that the circumcircle of any triangle in the triangulation does not contain any other point of Q within its interior [31]. Figure 3.4 illustrates this principle by showing how each triangle in the Delaunay triangulation adheres to this circumcircle property, which maximises the minimum angle of the triangles, avoiding skinny triangles leading to a more stable structure. Next, we formally introduce the Delaunay triangulation and discuss its essential geometric properties before moving on to the algorithm used for its construction. This foundational understanding of the Delaunay triangulation and its relationship with the Voronoi diagram is important, as it sets up the geometric basis for subsequent applications and methods that rely on this structure.

The Delaunay triangulation, denoted as $DT(Q)$, is formed by connecting the growth centers of all adjacent Voronoi polygons, resulting in a tessellation of the region within the convex hull of the point set.

Property DT 1. For a given set Q of m points, any triangulation $T(Q)$ will consist of the same number of triangles, $m_t = 2(m-1) - m_x$, and the same number of edges, $m_e = 3(m-1) - m_x$, where m_x is the number of points on the convex hull of Q. This property is proven in [65].

Property DT 2. For a set $Q = \{q_1, \ldots, q_m\}$, any edge (v_{ei}, v_{ej}) forms part of a circle centered at a point x that passes through v_{ei} and v_{ej}, with no other points from Q lying inside the circle. This result can be found in [66].

Property DT 3. An edge (v_{ei}, v_{ej}) is a Delaunay edge if and only if its circumcircle[3] does not contain any other points from Q in its interior. This property, also known as the circle criterion, serves as a rule for constructing the Delaunay triangulation and is proven in [68].

Property DT 4. The Delaunay triangulation of a set of points satisfies the MAX-MIN angle criterion, meaning that it maximises the minimum angle across all triangles in the triangulation. However, a triangulation that satisfies this criterion is not necessarily a Delaunay triangulation [68].

The construction algorithm we use is based on the incremental method as described in [63, 78]. This approach incrementally adds points to the triangulation and restructures the mesh to maintain the Delaunay property. Algorithms 2 and 3 briefly outline the steps involved in this iterative procedure.

[3]The circumcircle of a triangle is the unique circle that passes through all three of its vertices, making it the triangle's circumscribed circle For more details, see MathWorld's entry on Circumcircle.

Algorithm 2 Delaunay construction

Input: A set Q of m points contained within a rectangular boundary.
Output: A Delaunay triangulation satisfying the MAX-MIN angle criterion.
procedure INITIALISATION
    Clean the input set Q by removing any redundant points.
    Divide the rectangular area into approximately $\sqrt{m}$ grid cells.
    Rearrange the points in Q according to their corresponding grid indices.
    Insert the first point into the rectangle and connect it to the four corners to create an initial triangulation.
end procedure
procedure ITERATIVE PROCESS
    for each remaining point $q \in Q$ **do**
        Insert point q into the current triangulation.
        Connect point q to the vertices of the triangle that encloses it.
        Up to four convex quadrilaterals may form.
        Perform diagonal flips using ANGLE_SWAP to satisfy the MAX-MIN angle condition.
        Use QUAD_CHECK to evaluate neighbouring quadrilaterals after each flip.
        Further flips may be required if the condition is not satisfied.
        The flip process may propagate outward but will eventually terminate, as proven by Lawson's theorem.
    end for
    Once all points in Q have been processed, the triangulation is complete.
end procedure

Figures 3.5 and 3.6 present two examples demonstrating the reordering of points by sequentially progressing through bins, starting from a designated bin and continuing to adjacent bins, as described in Algorithm 2.

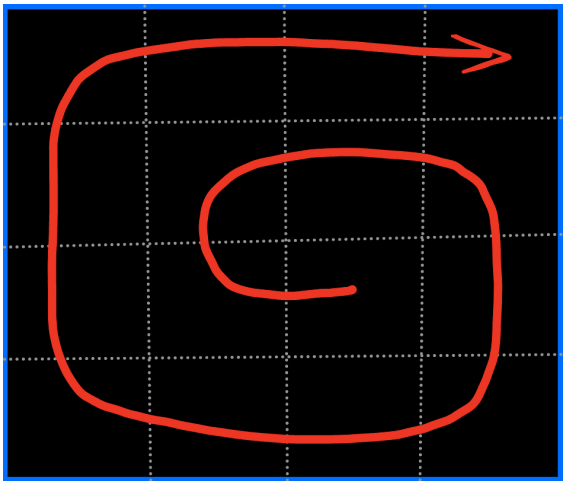


Figure 3.5: First possible ordering scheme.

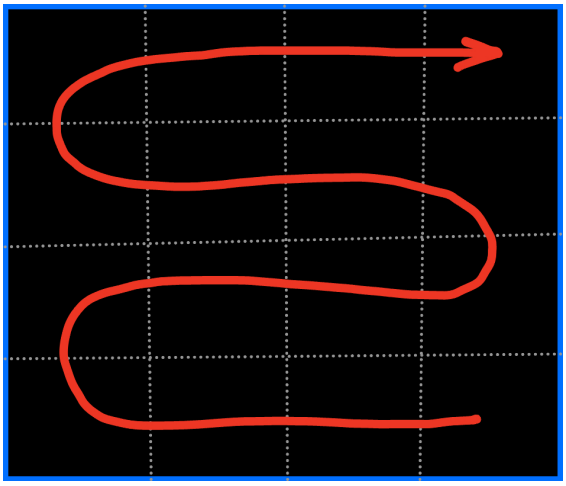


Figure 3.6: Second possible ordering scheme.

Algorithm 3 Locate triangle.

Input: A point (x_0, y_0) to be located within the triangulation.
Output: The triangle Δ that encloses the point.
procedure LOCTRI
    Begin by considering the last triangle Δ that was created.
    while (x_0, y_0) is not inside the current triangle Δ **do**
        for $i = 1$ to 3 **do** ▷ Iterate over the triangle's vertices and edges
            Set $i_{\text{next}} \leftarrow (i \mod 3) + 1$
            if $(y_0 - y_i) \times (x_{i_{\text{next}}} - x_0) > (x_0 - x_i) \times (y_{i_{\text{next}}} - y_0)$ **then**
                Jump to the neighbouring triangle $\Delta_{i_{\text{next}}}$.
                Update $\Delta \leftarrow \Delta_{i_{\text{next}}}$
                Continue checking with this new triangle.
            end if
        end for
    end while
    Return the triangle Δ that contains the point.
end procedure

The data structure used by the iterative algorithm is illustrated through an example. This example is shown in Figure 3.7 and Tables 3.1 and 3.2. The triangulation is stored in this specific format [63].

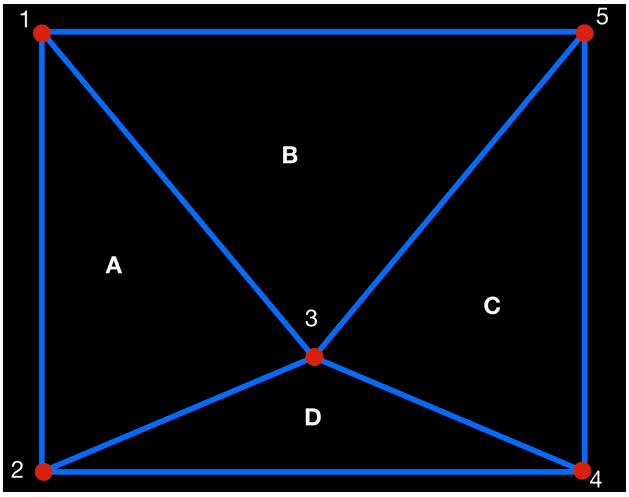


Figure 3.7: Vertices and edges.

Table 3.1: Vertex coordinates.

Vertex	1	2	3	4	5
x	1	1	18	31	31
y	1	31	17	31	1

Table 3.2: Adjacent triangles and vertices (counterclockwise).

	Triangle A	Triangle B	Triangle C	Triangle D
Adjacent $\mathcal{N}(\Delta, 1)$	2	1	4	1
Vertex $\mathcal{V}(\Delta, 1)$	1	3	4	2
Adjacent $\mathcal{N}(\Delta, 2)$	0	3	0	0
Vertex $\mathcal{V}(\Delta, 2)$	2	5	5	4
Adjacent $\mathcal{N}(\Delta, 3)$	4	0	2	3
Vertex $\mathcal{V}(\Delta, 3)$	3	1	3	3

In this triangulation, each triangle Δ has three adjacent triangles, denoted as $\mathcal{N}(\Delta, 1)$, $\mathcal{N}(\Delta, 2)$, and $\mathcal{N}(\Delta, 3)$, as shown Table 3.2. These adjacent triangles are arranged in a counterclockwise manner around Δ. Each triangle Δ also has three vertices, labeled as $\mathcal{V}(\Delta, 1)$, $\mathcal{V}(\Delta, 2)$, and $\mathcal{V}(\Delta, 3)$, also listed in Table 3.2.

Mathematically, the vertex $\mathcal{V}(\Delta, 1)$ is the point where triangles $\mathcal{N}(\Delta, 1)$, $\mathcal{N}(\Delta, 2)$, and Δ meet. This implies that all three triangles are incident at the common vertex $\mathcal{V}(\Delta, 1)$. The same rule applies to vertices $\mathcal{V}(\Delta, 2)$ and $\mathcal{V}(\Delta, 3)$, with respect to their neighbouring triangles.

In the table, a value of 0 for $\mathcal{N}(\Delta, i)$ indicates that there is no adjacent triangle across the edge opposite vertex $\mathcal{V}(\Delta, i)$. This means triangle Δ shares this edge with the boundary of the triangulated region, representing a degenerate or null triangle at that position. Therefore, in a triangulation over the convex hull, such "null triangles" occur wherever there is no adjacent triangle across a particular edge.

For example, in Triangle A, the adjacent triangles are $\mathcal{N}(\mathrm{A}, 1) = 2$, $\mathcal{N}(\mathrm{A}, 3) = 4$, and $\mathcal{N}(\mathrm{A}, 2) = 0$. These are incident at vertex $\mathcal{V}(\mathrm{A}, 1) = 1$. Since $\mathcal{N}(\mathrm{A}, 2) = 0$, this indicates that Triangle A shares one of its edges with the boundary at vertex 2.

We also discuss Fortune's algorithm, also known as the plane sweep algorithm [44], which efficiently constructs the Voronoi diagram for a given set of points. This algorithm dynamically maintains a structure called the beach line, which evolves as new points are processed. The beach line represents the boundary between processed and unprocessed points, and the algorithm manages events using a priority queue. Each event either introduces a new site to the beach line or removes a circle event, which corresponds to the removal of an arc from the beach line. These steps allow the Voronoi diagram to be updated incrementally, achieving a time complexity of $\mathcal{O}(n \log n)$ [29]. This makes Fortune's algorithm one of the most efficient methods for constructing the Voronoi diagrams.

While other algorithms are still useful, Fortune's algorithm, is adopted from the implementations in the `Foronoi` Python package [120], the `scipy.spatial` module, the `scipy.spatial.Voronoi` function [32, 121], and in `C++`[4]. This algorithm combines the efficiency of Divide and Conquer with the incremental nature of Incremental Construction, without the complexities involved in merging or maintaining the diagram at every step. Its plane-sweep approach is highly efficient in managing events and dynamically updating the diagram, making it ideal for larger data sets due to its $\mathcal{O}(n \log n)$ complexity, which scales effectively with the number of points.

The Voronoi diagram is constructed by sweeping a vertical line across the plane from left to right. As the sweep line progresses, it incorporates points to its left into the diagram while leaving points to its right out. The beach line, to the left of the sweep line, is a piecewise parabolic curve that separates the area known to have a Voronoi diagram from the area not yet processed. Each parabola on the beach line represents points that are equidistant from the sweep line and a known point. The edges of the Voronoi diagram are determined by the intersection points of these parabolas [30]. To ensure that the base of every parabola remains correctly positioned halfway between its associated point and the sweep line, the beach line is updated as the sweep line advances [127].

[4] `https://blog.ivank.net/fortunes-algorithm-and-implementation.html` accessed August 2024.

We present an example of Fortune's algorithm in Voronoi diagram construction. Think of the process of constructing the Voronoi diagram using Fortune's algorithm as a dynamic progression. The following explanation walks through some of the stages, to illustrate how the algorithm unfolds over time.

In Figure 3.8, we begin with a set of 11 points, the generator points (sites) in the algorithm. These are the points from which the Voronoi diagram will be constructed. Without loss of generality, the diagram shows these sites distributed across the plane. These generator points are the initial input for Fortune's algorithm.

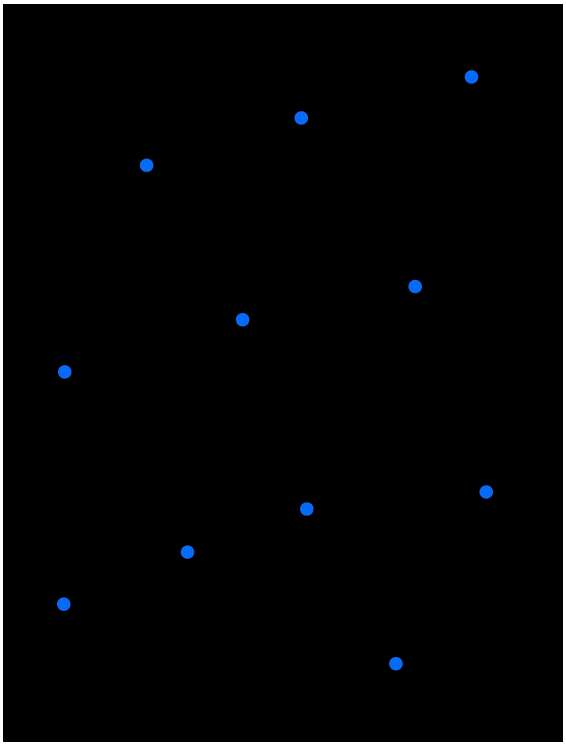


Figure 3.8: Initial state of the plane with generator points (sites).

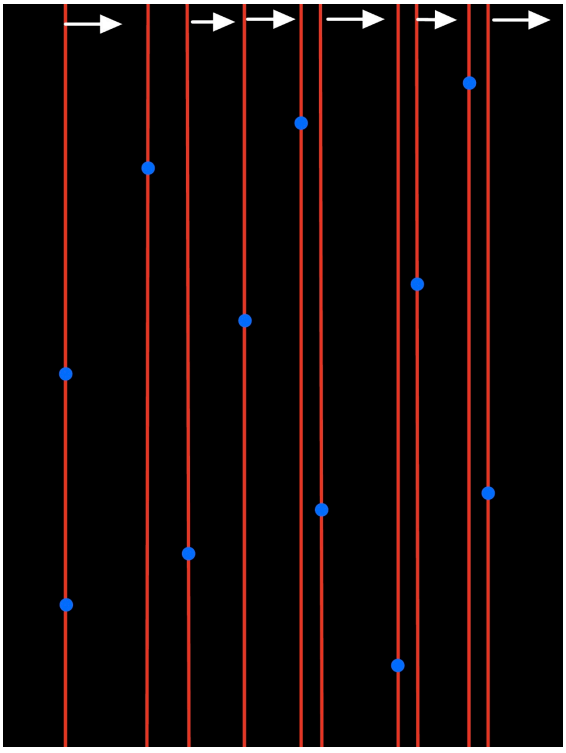


Figure 3.9: Sweep line (red) moving left to right across the generator points.

Figure 3.9 illustrates the sweep line, shown as a red vertical line, as it moves from left to right across the plane. The sweep line captures specific moments where it intersects the generator points (sites). This is showcasing how the algorithm progresses through different instances. At this stage, the sweep line begins interacting with the generator points.

In Figure 3.10, we introduce the beach line, shown as a yellow curve. The beach line represents the boundary between the processed region (to the left of the sweep line) and the unprocessed region (to the right). In this moment, the beach line is still straight, but as more points are processed, it gradually develops into the parabolic curves you see in Figure 3.11. These parabolic arcs represent the region equidistant between the sweep line and each generator point.

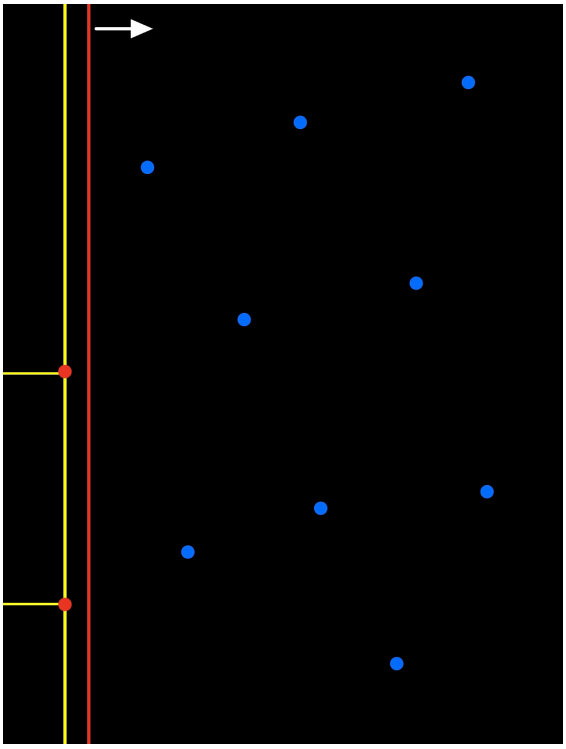


Figure 3.10: Beach line shown at an initial straight configuration.

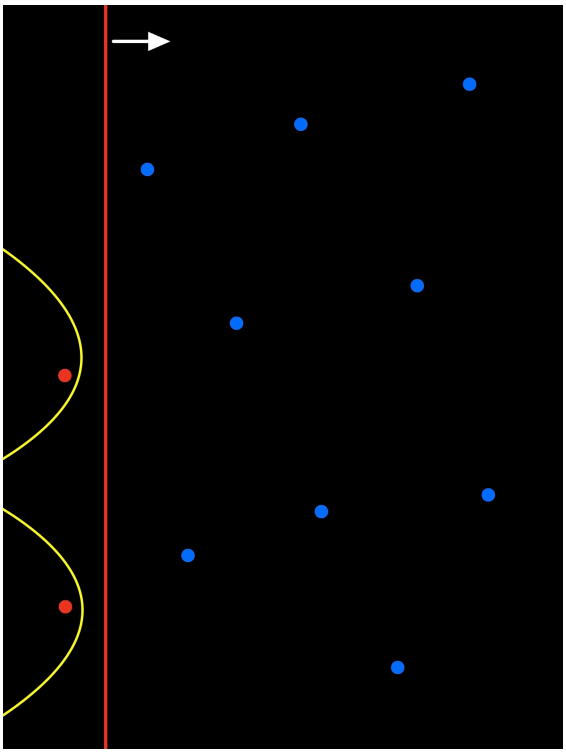


Figure 3.11: Beach line transitioning into a parabolic shape.

Notice in both Figures 3.10 and 3.11, the blue solid circles represent generator points that have yet to be processed by the sweep line. The red solid circles represent the sites that have already been processed. The beach line evolves as more points are visited, creating the characteristic arcs.

By the time we reach Figure 3.12, the sweep line has progressed further to the right. At this stage, focus on the specific generator point, highlighted by a red circle filled with blue. This signifies that the point is currently being processed, but not yet fully incorporated into the Voronoi diagram (it isn't fully red like the already processed points). When the sweep line touches this generator point, its associated parabolic arc degenerates into a vertical ray shooting upwards from the point to the current beach line, as shown in Figure 3.13.

As the sweep line moves further right, the vertical ray begins to widen, transforming into an arc along the beach line, which you can observe in Figure 3.14. This is the natural progression of the beach line as more sites are processed.

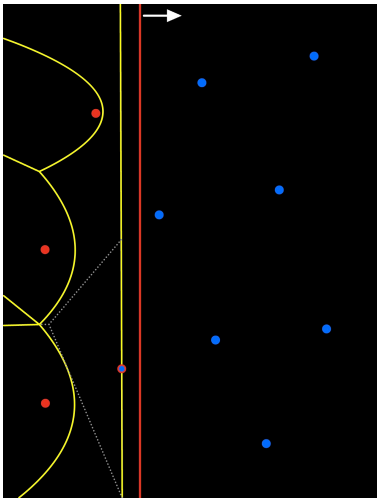


Figure 3.12: Focus on a specific generator point being processed.

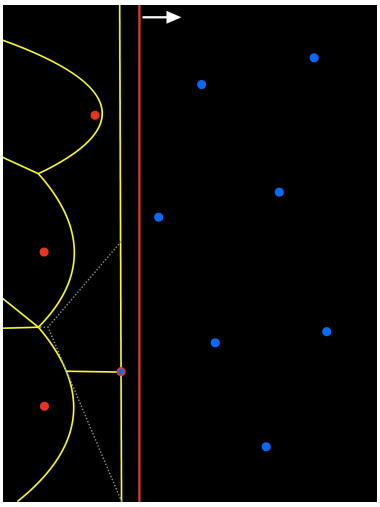


Figure 3.13: Vertical ray forming from the generator point.

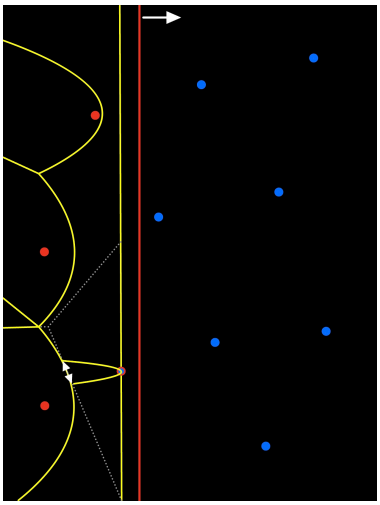


Figure 3.14: Arc forming along the beach line.

The construction of the Voronoi diagram begins to take shape in Figures 3.15 and 3.16. At this point, we can see the orange lines forming, which represent the edges of the Voronoi polygons that have already been fully constructed. Notice that some edges have components that are both orange and yellow. This indicates that the orange segments are already processed, while the yellow parts are still being updated as the beach line continues to evolve.

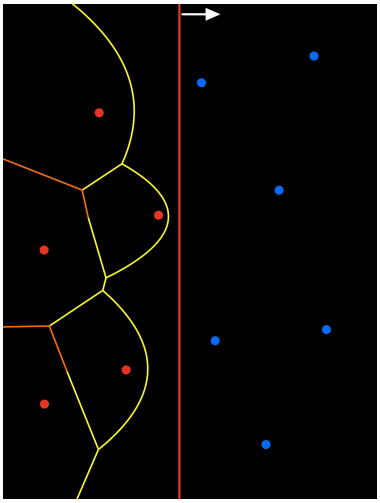


Figure 3.15: Early construction of Voronoi edges.

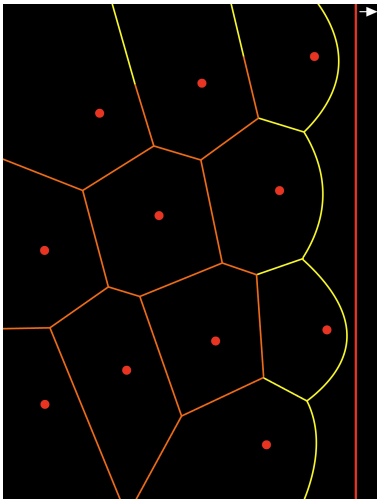


Figure 3.16: Voronoi diagram with incomplete edges.

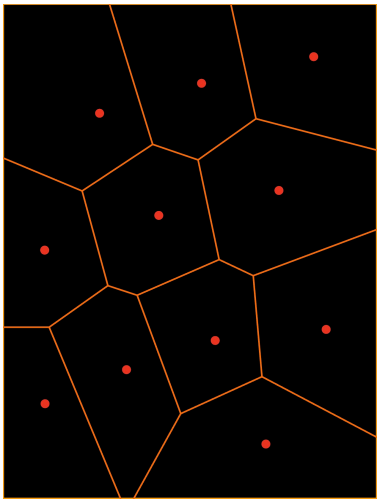


Figure 3.17: Final Voronoi diagram with all regions complete.

Finally, in Figure 3.17, the Voronoi diagram is complete. The full set of Voronoi polygons is now visible, and all the generator points have been processed. The diagram at this stage has no remaining beach line or sweep line, and the edges of the Voronoi polygons are fully constructed.

The algorithm's process can be summarised as follows. We define $\phi(a)$ as the transformation $\phi(a) = (a_x, a_y + \delta(a))$, where $\delta(a)$ represents the Euclidean distance between point a and the nearest site. The

beach line is denoted by $\mathcal{B}$, and S_u represents the region covered by site u. The boundary ray between sites u and v is indicated by L_{uv}, while Q is the set of points on which the algorithm is applied. The points $u_1, u_2, \ldots, u_m$ are extracted from Q based on the smallest y-coordinate and are ordered by x-coordinate. The operation LLP(X) removes the lowest and leftmost point from X, sorting by y-coordinate and by x-coordinate if the y-coordinates are identical. Finally, $V(Q)$ is the Voronoi diagram of Q, which the algorithm aims to construct, with H initialised as $\{u_1, u_2, \ldots, u_m, Q\}$.

Algorithm 4 Fortune's algorithm for constructing a Voronoi diagram.

Input: Set of points Q.

Initialise $H \leftarrow Q$
Create initial vertical boundary rays $L^0_{u_1,u_2}, L^0_{u_2,u_3}, \ldots, L^0_{u_{m-1},u_m}$
Initialise $\mathcal{B} \leftarrow \phi(S_{u_1}), L^0_{u_1,u_2}, \phi(S_{u_2}), L^0_{u_2,u_3}, \ldots, \phi(S_{u_{m-1}}), L^0_{u_{m-1},u_m}, \phi(S_{u_m})$
while H is not empty **do**
    $u \leftarrow \texttt{LLP}(H)$
    if u is a site in $\phi(V(Q))$ **then**
        Locate the region $\phi(S_v)$ in $\mathcal{B}$ containing u, bounded by L_{wv} on the left and L_{vs} on the right
        Create new boundary rays L^-_{uv} and L^+_{uv} with bases at u
        Replace $\phi(S_v)$ with $\phi(S_v), L^-_{uv}, \phi(S_u), L^+_{uv}, \phi(S_v)$ in $\mathcal{B}$
        Remove from H any intersection between L_{wv} and L_{vs}
        Add to H any intersection between L_{wv} and L^-_{uv}
        Add to H any intersection between L^+_{uv} and L_{vs}
    else
        u is a Voronoi vertex in $\phi(V(Q))$
        u is the intersection of L_{vw} on the left and L_{ws} on the right
        L_{tv} is the left neighbour of L_{vw} and L_{sz} is the right neighbour of L_{ws} in $\mathcal{B}$
        if $v_y = s_y$ **then**
            Create a new boundary ray L^0_{vs}
        else if u is to the right of the higher of v and s **then**
            Create L^+_{vs}
        else
            Create L^-_{vs}
        end if
        Replace $L_{vw}, \phi(S_w), L_{ws}$ with the newly created L_{vs} in $\mathcal{B}$
        Remove from H any intersection between L_{tv} and L_{vw}
        Remove from H any intersection between L_{ws} and L_{sz}
        Add to H any intersection between L_{tv} and L_{vs}
        Add to H any intersection between L_{vs} and L_{sz}
        Record u as the summit of L_{vw} and L_{ws} and the base of L_{vs}
        Output the boundary segments L_{vw} and L_{ws}
    end if
end while
Output any remaining boundary rays in $\mathcal{B}$

Output: The constructed Voronoi diagram $V(Q)$.

Herein, planar Voronoi diagrams are generated using the `R` programming language [101] and the `deldir` and `spatstat` packages [11, 119]. The `deldir` package (short for Delaunay Dirichlet) in `R` uses the iterative algorithm we previously discussed. While there are other packages available, such as `terra` [55], `sf`, and `ggforce` [94], we won't look into them here, as the algorithms covered so far (iterative and Fortune's) are sufficient for our practical applications. The Voronoi diagram shown in Figure 3.18 is unbounded, meaning that some cells extend infinitely without a defined outer boundary. For practical applications, this is less

useful because we need to analyse relationships within a finite region. To address this, we introduce a bounded region $R \subseteq \mathbb{R}^2$, containing the generator points, and focus on the subset $\mathcal{V}_{\cap R} \subseteq \mathbb{R}^2$, defined as

$$\mathscr{V}_{\cap R} = \{V(q_1) \cap R, \ldots, V(q_m) \cap R\}\,.$$

This subset, the bounded Voronoi diagram, restricts each cell $V(q_i)$ to the region R, creating finite, closed regions with a clear boundary. This enhances spatial analysis near edges, accurately representing proximity relationships within the area of interest. Figure 3.19, shows an example of a bounded Voronoi diagram.

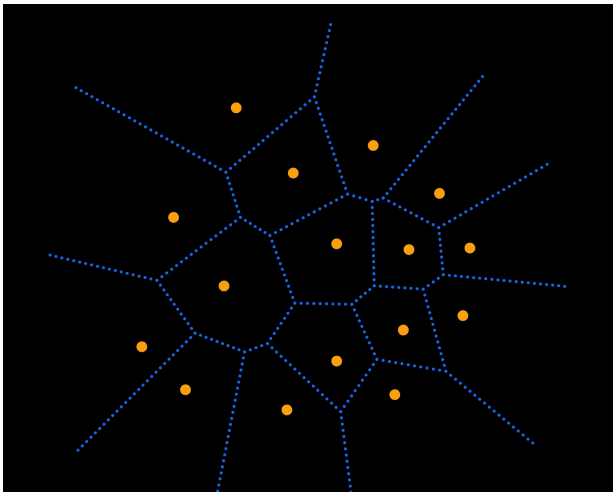


Figure 3.18: Unbounded Voronoi diagram within an arbitrary rectangular window.

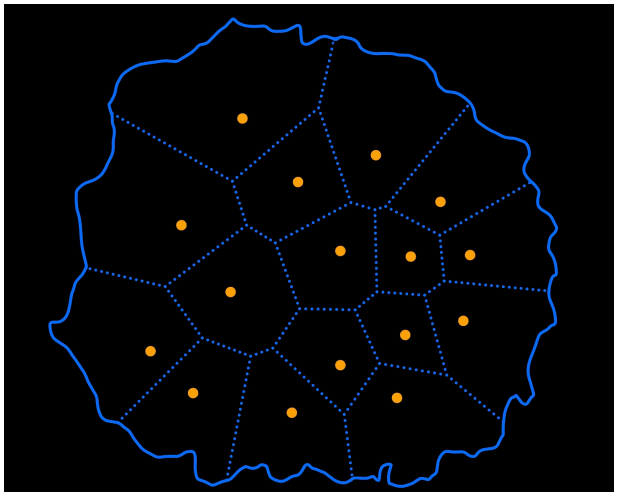


Figure 3.19: Bounded Voronoi diagram restricted to the region $R \subseteq \mathbb{R}^2$.

3.2 Network Voronoi Diagram (N-VD)

The aim of this section is to extend the concept of the planar Voronoi Diagram by substituting the plane with a network space $\mathcal{N} = (\mathcal{V}, \mathcal{L})$, consisting of a set of nodes $\mathcal{V}$ and a set of links $\mathcal{L}$, such as those found in a road network, and by replacing the Euclidean distance with distances defined on the network, specifically the shortest-path distance. This Voronoi diagram is referred to as the network Voronoi diagram [85]. Figure 3.20 is an example of a network Voronoi diagram. Different regions of the network are colour-coded, with each colour representing parts of the road network closer to the red generator points than to any other generator points on the network [81].

Figure 3.20: This network Voronoi diagram showcases the boundary points of Voronoi sub-networks (small circles) and the dominant generator points (large circles) [81].

The Voronoi diagram on a network is defined for a set of points $Q = \{q_1, \ldots, q_m\}$ on the network, where the distance $d_{\mathcal{N}}(q, q_i)$ between an arbitrary point q and a generator point q_i on the network is measured using the shortest-path distance, $d_{\mathcal{N}}(q, q_i)$ [82]. In these terms, the network Voronoi Diagram is defined as a collection of subnetworks, known as Voronoi subnetworks, $\text{Vor} = \{\text{Vor}_1, \ldots, \text{Vor}_m\}$, where the Voronoi subnetwork Vor_i is given by

$$\text{Vor}_i = \{q \mid d_{\mathcal{N}}(q, q_i) \leq d_{\mathcal{N}}(q, q_j) \text{ for all } j \neq i\} \tag{3.3}$$

The network $\mathcal{L} = \{l_1, \ldots, l_{m_L}\}$ is associated with at least one of the m Voronoi subnetworks, $\text{Vor}_1, \ldots, \text{Vor}_m$. Consequently, the n Voronoi subnetworks collectively cover the entire network $\mathcal{L}$. However, they are not necessarily mutually exclusive. This characteristic distinguishes them from the ordinary network Voronoi diagram, where the subnetworks are mutually exclusive except at boundary points. An example of the nonexclusive case is illustrated in Figure 3.21, where the generator points are q_1 and q_2, and the lengths of the links l_2 and l_3 are equal. In this case, Voronoi subnetwork Vor_1 consists of l_1 and l_2, while Vor_2 consists of l_1 and l_3. Consequently, points on l_1 are shared between Voronoi subnetworks Vor_1 and Vor_2 [86].

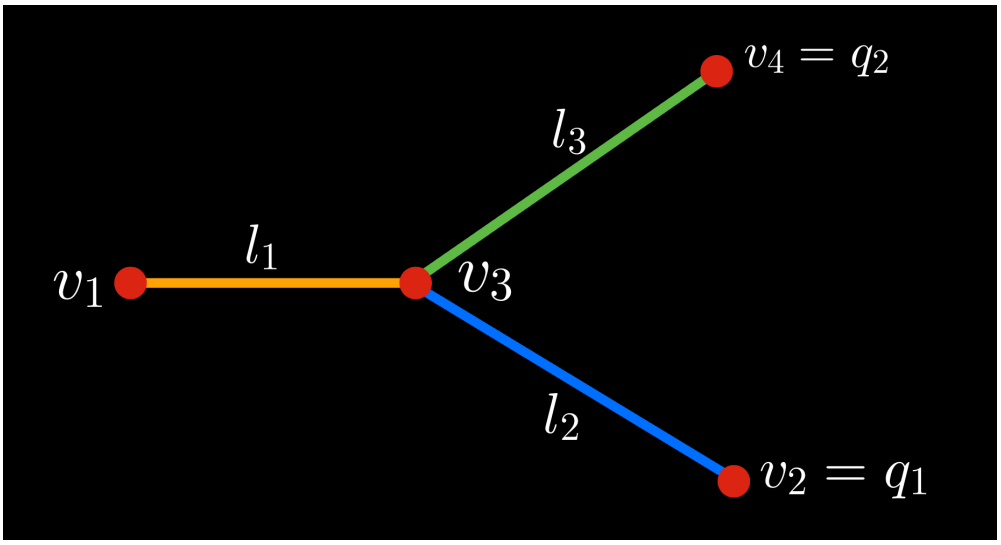


Figure 3.21: Illustration of overlapping Voronoi subnetworks $\text{Vor}_1 = \{q \mid q \in l_1 \cup l_2\}$ and $\text{Vor}_2 = \{q \mid q \in l_1 \cup l_3\}$, where v_3 represents a critical node.

To address this special case, we define a node v that meets the following criteria: k links converge at v, where $k \geq 3$; there exist k' shortest paths originating from v, each connecting v to a generator point in Q; and no other generator points are located along these paths 3.21. Additionally, k' satisfies $2 \leq k' < k$, and the lengths of these shortest paths are equal. In the example shown in Figure 3.21, $k = 3$, $k' = 2$, and the shortest paths between v_3 and q_1 and between v_3 and q_2 (i.e., l_2 and l_3) have the same length. Such a node is referred to as a critical node [86]. For instance, in Figure 3.21, node v_3 is a critical node. Essential geometric properties of the traditional network Voronoi diagram, as detailed in [86], are summarised below for subsequent use in our network Voronoi diagram construction.

Characteristic N 1. The network $\mathcal{N} = (\mathcal{V}, \mathcal{L})$ without critical nodes has Voronoi polygons $V_{\mathcal{N}}(q_1), \ldots, V_{\mathcal{N}}(q_m)$ generated by $Q = \{q_1, \ldots, q_m\}$ that are mutually exclusive, except at boundary points between Voronoi polygons, and collectively exhaustive for $\bar{\mathcal{L}} = \bigcup_{i=1}^{m_L} l_i$ with $\mathcal{L} = \{l_1, \ldots, l_{m_L}\}$.

Networks satisfying Characteristic N1 can use standard algorithms for constructing N-VD, without requiring specialised adaptations. In the following theoretical discussion, we assume that this characteristic holds. Based on this condition, two network Voronoi polygons can only interact by sharing boundary points or not sharing any points at all. If they share a common boundary, they are defined as adjacent cells.

Characteristic N 2. The nearest neighbour of a generator q_i in terms of shortest path distance is always a generator whose Voronoi region $V_{\mathcal{N}}(q_i)$ is adjacent to that of $V_{\mathcal{N}}(q_j)$.

Characteristic N 3. The Voronoi cell of generator q_i contains a point q on $\mathcal{L}$ if and only if q_i is the nearest generator to q.

3.2.1 Computational methods for the network Voronoi diagram

Following the conceptualisation of Network Voronoi Diagrams, actual applications require computational methods for their construction. N-VDs are network tessellations in which each network point is allocated to its nearest generator according to the shortest-path distances. This section presents a commonly used algorithm for the construction of N-VDs. Consequently, calculating shortest-path distances from points to generators is important for constructing N-VDs [56]. We present a method for computing multiple-source shortest paths, the multisource Dijkstra method [109], which manages multiple sources simultaneously.

Consider a network $\mathcal{N} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ represents the set of nodes, $\mathcal{L}$ denotes the set of links, and $Q \subset \mathcal{V}$ is the set of generators. Let $\ell_c(v, v')$ represent the length or cost of the link between nodes v and v'. In the following algorithm, we extend the single-source Dijkstra method to multiple sources: $\text{Dist}(v)$ indicates the shortest-path distance from node v to the nearest source (generator point), $\text{prev}(v)$ denotes the immediate predecessor on the path from the nearest source to v, and S_A is a set of nodes whose neighbours have yet to be processed. Additionally, $\text{N}_\text{S}(v)$ represents the nearest source node to v.

Algorithm 5 Multisource Dijkstra method.

Input: Set of generator nodes Q, graph $\mathcal{N} = (\mathcal{V}, \mathcal{L})$ with nodes $\mathcal{V}$ and edges with costs $\ell_c(v, v')$
Output: Shortest-path distances $\text{Dist}(v)$ and nearest sources $N_S(v)$ for all nodes $v \in \mathcal{V}$
Initialisation for generators:
for each node $v \in Q$ **do**
    $\text{Dist}(v) = 0$
    $N_S(v) = v$
end for
Initialisation for other nodes:
for each node $v \in \mathcal{V} \setminus Q$ **do**
    $\text{Dist}(v) = \infty$
end for
Initialise the Active set:
$S_A = Q$
Processing nodes:
while $S_A \neq \emptyset$ **do**
    Select and remove node v with the smallest $\text{Dist}(v)$ from S_A
    for each adjacent node v' to v **do**
        if $\text{Dist}(v') > \text{Dist}(v) + \ell_c(v, v')$ **then**
            $\text{Dist}(v') = \text{Dist}(v) + \ell_c(v, v')$
            $\text{prev}(v') = v$
            $N_S(v') = N_S(v)$
            Add v' to S_A
        end if
    end for
end while
Termination:

When Algorithm 5 completes, for each node $v \in \mathcal{V}$, $N_S(v)$ represents the nearest generator, and $\text{Dist}(v)$ denotes the shortest-path distance to $N_S(v)$. Thus, each node v is assigned to the Voronoi cell of the generator indicated by $N_S(v)$. For every link $\ell = (v, v') \in \mathcal{L}$ that connects nodes v and v', the membership

of the link in a Voronoi cell is then determined. If both nodes v and v' share the same generator, i.e., $N_S(v) = N_S(v')$, the entire link is assigned to the Voronoi cell of $N_S(v)$ [81]. However, if $N_S(v) \neq N_S(v')$, the link ℓ is divided at the point $p \in \ell$ where the equation $\text{Dist}(v) + d_S(v, p) = \text{Dist}(v') + d_S(v', p)$ holds. Here, $d_S(p, q)$ represents the distance from q to p measured along the link ℓ. A point q satisfying this condition always exists because v and v' belong to different Voronoi polygons, ensuring that $|\text{Dist}(v) - \text{Dist}(v')| \leq d(v, v')$ [82]. At point q, a new node is inserted to divide the link into two shorter links: one from v to q assigned to the Voronoi cell of $N_S(v)$, and another from v' to q assigned to the Voronoi cell of $N_S(v')$. The new node q thus marks the boundary between the two Voronoi polygons. This procedure is repeated for every link $\ell = (v, v')$ where $N_S(v) \neq N_S(v')$. The final result is a network Voronoi diagram induced by Q [83]. This procedure is summarised in Algorithm 6.

Algorithm 6 Network Voronoi diagram.

Input: Network $\mathcal{N}$ with nodes $\mathcal{V}$ and links $\mathcal{L}$, and generator assignment $N_S(v)$ for each node $v \in \mathcal{V}$ from Algorithm 5
Output: N-VD induced by Q
Step 1: Assign nodes to Voronoi polygons
for each node $v \in \mathcal{V}$ **do**
    Assign v to the Voronoi cell of $N_S(v)$
end for
Step 2: Process each link
for each link $\ell = (v, v') \in \mathcal{L}$ **do**
    Check Voronoi cell membership of v and v'
    if $N_S(v) = N_S(v')$ **then**
        The entire link ℓ belongs to the Voronoi cell of $N_S(v)$
    else
        Divide the link:
        Find point $q \in \ell$ such that: $\text{Dist}(v) + d_S(v, q) = \text{Dist}(v') + d_S(v', q)$
        Insert a new node at q
        Assign the sub-link (v, q) to the Voronoi cell of $N_S(v)$
        Assign the sub-link (q, v') to the Voronoi cell of $N_S(v')$
    end if
end for
Step 3: Repeat for remaining links
for each link $\ell = (v, v') \in \mathcal{L}$ where $N_S(v) \neq N_S(v')$ **do**
    Repeat step 2
end for
Output: Network Voronoi diagram of Q

3.3 Network Voronoi Polygons

The techniques in Section 3.2 produce a tessellation of the network space into tiles. This partitioned the network into uniquely labeled non-overlapping subsets. For additional analysis, we represent these tiles (Voronoi subnetworks) as polygons, akin to conventional planar Voronoi diagrams (like in Figure 3.17). This section introduces a method to generate these polygons, filling a gap in the literature where most studies focus on tessellating the network space but do not explicitly convert it into polygonal representations. The need for this polygonal representation is important because these polygons will be used to quantitatively

compare against administrative map boundaries, and the boundaries we created using Euclidean distance-based Voronoi diagrams. When superimposed on maps, these polygons are an effective visual and analytical instrument to assess similarities between the network-derived borders and the administrative boundaries.

As we shall demonstrate in subsequent spatial similarity tests, these polygons will be instrumental in extracting specific networks within a given police precinct. This will allow for a more precise quantitative comparison between the network-based tessellation and traditional Voronoi diagrams constructed using Euclidean distances. Polygonal boundaries make it possible to calculate metrics like boundary overlap, deviation, and area differences.

A systematic approach for extracting individual subnetworks from a larger linear tessellation is introduced in Section 3.3. The tessellation consists of several components. The components include the network itself which is composed of vertices where edges meet, the edges that connect these vertices, and a connectivity matrix that describes these connections. Distinct subnetwork region are each represented by one or more segments of the network edges. Each segment has an attribute indicating the subnetwork region to which it belongs.

To extract the subnetwork linked to any region, one must consider the network as a graph composed of interconnected vertices and edges. The tessellation segments this graph into subnetworks; yet, vertex information alone fails to provide sufficient details about subnetwork structure. Additional information from network segments and subnetwork identities are necessary to delineate these regions.

Initially, we filter the tessellation data to extract the particular subnetwork associated with a specified point. Each subnetwork is delineated by the set of its line segments, identified by start and end points along the edges. Data is filtered to gather all line segments constituting a certain subnetwork. To extract several subnetworks, the procedure is reiterated for each distinct identifier, resulting in a compilation of subnetworks.

The boundaries of each subnetwork are identified by marking the start and end of each of the road line segments. The extracted end points can be used in the creation of the polygon boundaries of the subnetworks. However, the use of these boundary points does not yield an accurate tessellation of the network. Since it produces gaps and overlapping polygonal regions. An example is shown in Figure 3.22

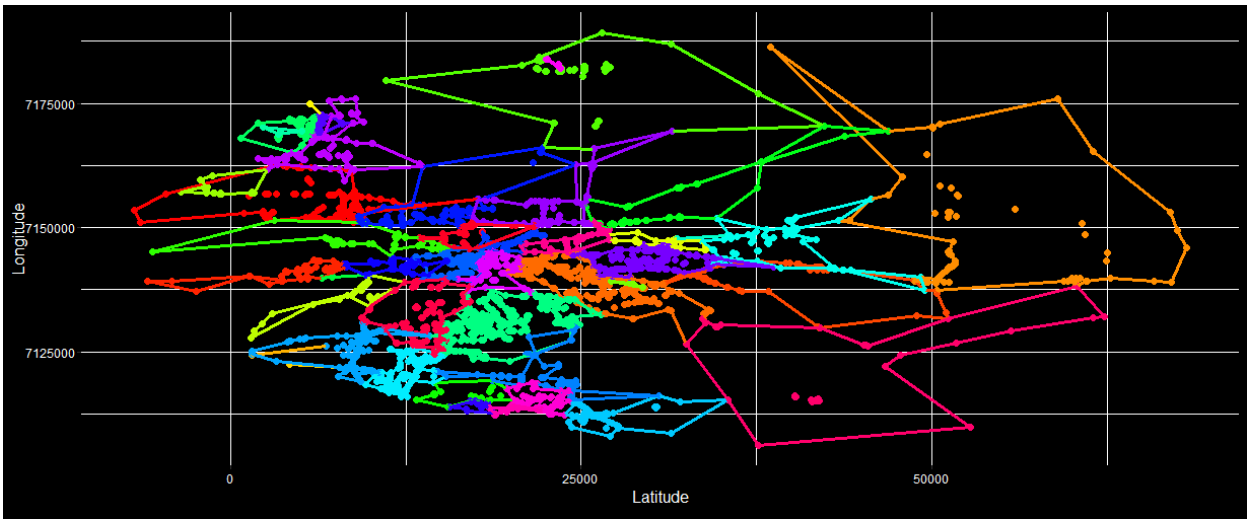


Figure 3.22: Illustration of boundary points marking the edges of line segments in Gauteng, highlighting gaps and overlaps before the refinement process. Note, each colour represent a different subnetwork region.

To generate more accurate polygons, the process is systematically broken down into the four following steps.

3.3.1 Grid Partitioning and Centroid Projection

First, define the spatial domain $X \subseteq \mathbb{R}^2$, which encompasses the entire road network. This domain is partitioned into a grid of size $N \times M$, where N and M are natural numbers representing the number of grid cells along the horizontal and vertical axes, respectively. Formally, the grid partition of X is expressed as

$$X = \{(i, j) \mid i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\}, \text{ for some } N, M \in \mathbb{N}\},$$

where each cell (i, j) corresponds to the i-th column and j-th row of the grid, starting from the bottom-left corner of the domain. Let c_{ij} represent the centroid of the grid cell located at row i and column j, which will serve as the representative location for that cell.

Next, consider the graph representation of the road network $\mathcal{N} = (V^*, L^*)$, where L^* is the set of line segments comprising the network. The network is defined as the union of individual line segments $l_1^*, \ldots, l_{m_L*}^*$, such that $\tilde{L^*} = \bigcup_{i=1}^{m_L*} l_i^*$.

To relate the grid centroids to the road network, we introduce a nearest projection function, f_{proj}, which projects each grid centroid c_{ij} onto the closest road segment in the network. The function $f_{\text{proj}} : X \to \tilde{L^*}$ computes the shortest Euclidean distance from the centroid c_{ij} to the set of road segments, projecting the

centroid onto the nearest segment. Formally, the function f_{proj} is defined as

$$f_{\text{proj}}(c_{ij}, \tilde{L}^*) = \arg\min_{l^* \in \tilde{L}^*} d(c_{ij}, l^*)$$
$$= p_{ij} \tag{3.4}$$

where

- $d(c_{ij}, l^*)$ is the distance between the grid centroid c_{ij} and the road segment l^*,
- p_{ij} is the projection of c_{ij} onto the nearest point on the road segment l^*.

Figure 3.23 illustrates an example of a network, which is used in Figure 3.24 to demonstrate how centroid points are projected onto their nearest road or network line segment. In Figure 3.24, the grid centroids, represented as blue circles, are snapped to the network (indicated by red arrows showing the direction and final destination). The network segments to which they are snapped are highlighted in orange. Segments that do not have any centroids snapped to them remain black, preserving the appearance of the original network.

Figure 3.23: The original road network, showing nodes and edges used in the analysis.

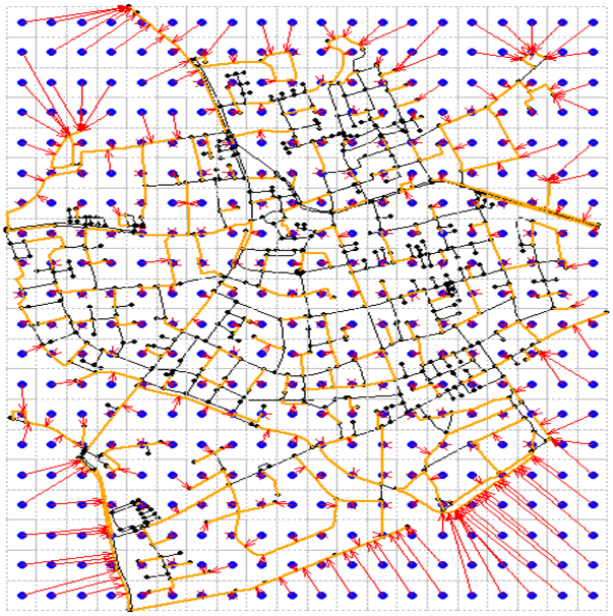


Figure 3.24: The blue centroids are snapped to their nearest network segment.

In this application, we opt to use the Euclidean distance d in Equation (3.4) for the projection, as it is computationally simpler and faster compared to more complex distance metrics. This choice is particularly effective as we are dealing with large data sets and fine grids (at least 1500×1500). When the grid is sufficiently fine, the difference between Euclidean distance and alternative metrics (such as geodesic or network distance) becomes negligible because the centroids c_{ij} are close to the road network [118, 90]. Therefore, using Euclidean distance is both justified and practical for this application.

3.3.2 Network-based Generator Projection and Centroid labeling

To continue building upon the previous step, we need to project (snap) generator points to the nearest segments of the network. This is because we are assuming everything happens on the the network [86]. The network serves as the foundational framework for spatial analysis on a network, much like the Cartesian plane underpins spatial analysis in two dimensions. Note that a network space is not analogous to spaces composed of networks, such as a function space in mathematics [86, 104, 95]. For each generator point $q_i \in Q$, we use the nearest projection function f_{proj} to snap these points onto the road network $\mathcal{N}$. We define the projection of each generator point q_i as follows

$$f_{\text{proj}}(q_i, \tilde{L}^*) = q_{l_i},$$

where q_{l_i} is the generator point q_i snapped to the nearest point on the network $\mathcal{N}$. Thus, for each generator point q_i, the function $f_{\text{proj}} : Q \to \{q_{l_1}, q_{l_2}, \ldots, q_{l_m}\} \subseteq \mathcal{N}$ snaps the generator points to the nearest location on the road network, yielding the set of projected generator points $\{q_{l_1}, q_{l_2}, \ldots, q_{l_m}\}$.

Now, we aim to assign each grid cell centroid c_{ij} (from the set X defined earlier) to the nearest generator point that has been projected onto the road network. This is done by computing the shortest path distance along the network between the snapped centroid $f_{\text{proj}}(c_{ij})$ and the snapped generator points $Q_l = \{q_{l_1}, q_{l_2}, \ldots, q_{l_m}\}$. To achieve this, we define a function h that assigns each centroid c_{ij} to the nearest generator point based on the shortest path distance on the network. The function $h : X \to \{1, \ldots, m\}$ is defined as follows

$$\begin{aligned} h(c_{ij}) &= \left\{ \gamma \mid \gamma = \arg\min_{q_l \in Q_l} \left\{ d_S(f_{\text{proj}}(c_{ij}), q_l) \right\} \right\} \\ &= \left\{ \gamma \mid \gamma = \arg\min_{q_l \in Q_l} \left\{ d_S(p_{ij}, q_l) \right\} \right\}, \end{aligned} \tag{3.5}$$

where

- $d_S(f_{\text{proj}}(c_{ij}), q_l)$ represents the shortest path distance between the snapped centroid $f_{\text{proj}}(c_{ij})$ and the generator point q_l on the network,
- γ is the index of the nearest generator point to the centroid c_{ij},
- If multiple generator points are equidistant (i.e., $|h(c_{ij})| > 1$), we select the first encountered index [81].

Once the nearest generator point q_{l_γ} is determined for each centroid c_{ij}, we assign an index $\gamma \in \{1, \ldots, m\}$ to each snapped point $f_{\text{proj}}(c_{ij})$ on the network. These indices correspond to the nearest generator point and

can be mapped back to the domain $X \subseteq \mathbb{R}^2$, filling the grid cells in X with the index of the closest generator point. This process effectively partitions the grid space based on proximity to the generator points. As a result of this procedure:

- Each grid cell in the domain X is labeled with the index of the nearest generator point.
- The assignment reflects the shortest path distance along the network, ensuring that centroids are aligned with the road network's geometry and structure.
- This partitioning can be visualised as a Voronoi-based network partition, where grid regions are formed around each generator point based on network distances.

The size of the grid into which X is divided significantly influences the shapes of the resulting polygons. For instance, Figures 3.25 and 3.26 demonstrate this effect: while both grids cover the same area of X, the grid in Figure 3.26 contains a greater number of smaller cells. These smaller cells allow for a more precise delineation of grid centroids and improve the representation of boundary cases, leading to a more accurate mapping of the Voronoi polygons. To enhance precision, in our application we use a grid of 1500×1500 cells. This high-resolution grid allows for a detailed partitioning of X. The smaller cell size provides a refined approximation of the tessellation boundaries, ensuring that even small differences in distance are accounted for. This level of granularity is important as capturing finer details will improve the resolution of the mapped regions (polygons) and accuracy of spatial analysis that follow.

1	1	1	2	2
1	3	3	3	2
4	3	3	3	2
4	4	5	5	5
4	4	5	5	5

Figure 3.25: A 5×5 grid. The coarser grid results in less accurate shapes, as the larger cells limit boundary detail and centroid precision, providing a rough spatial representation.

1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	2	2	2	2	2
1	1	1	3	3	3	3	2	2	2
1	1	3	3	3	3	3	2	2	2
4	4	3	3	3	3	3	5	5	5
4	4	3	3	3	3	3	5	5	5
4	4	4	4	4	5	5	5	5	5
4	4	4	4	4	5	5	5	5	5
4	4	4	4	4	5	5	5	5	5

Figure 3.26: A 10×10 grid. Increasing the grid size allows for more representative shapes, as smaller cells improve boundary accuracy and centroid mapping, creating a more detailed spatial representation.

3.3.3 Polygon representation of points and construction of concave hulls

Next we create a function that groups centroid coordinates with similar indices, resulting in a set of submatrices $\{M_1, \ldots, M_m\}$, where each submatrix corresponds to a set of points belonging or nearest or assigned to one of the m generator points. Having the points, a polygon representation of these points is needed. We use the idea of convex hulls which are iteratively refined to get concave hulls [91]. A concave hull is like a more tailored boundary around a set of points, adapting closely to their shape, compared to the convex hull, which forms a broader, outermost "rubber band" enclosure that ignores indentations in the data's distribution. So we opt for concave hulls as they provide a more natural and precise boundary . For each $M_i \in \{M_1, \ldots, M_m\}$, the convex hull H is the smallest convex polygon enclosing all points in M_i [13]. We implemented the algorithm in [91] in the construction of the concave hulls based on M_i. **Construction of Convex and Concave hulls:**

To demonstrate the process, consider the set of coordinates M^*, comprising eight randomly generated points, as defined in Equation 3.6 and visualised in Figure 3.27. The objective is to construct the convex hull using this data set. The goal is to construct the convex hull using this set. The complete set M^* is given by

$$M^* = \left\{ \begin{bmatrix} 0.288 \\ 0.551 \end{bmatrix}, \begin{bmatrix} 0.788 \\ 0.457 \end{bmatrix}, \begin{bmatrix} 0.409 \\ 0.957 \end{bmatrix}, \begin{bmatrix} 0.883 \\ 0.453 \end{bmatrix}, \begin{bmatrix} 0.940 \\ 0.678 \end{bmatrix}, \begin{bmatrix} 0.046 \\ 0.573 \end{bmatrix}, \begin{bmatrix} 0.528 \\ 0.103 \end{bmatrix}, \begin{bmatrix} 0.892 \\ 0.900 \end{bmatrix} \right\} \tag{3.6}$$

The algorithm begins by constructing the convex hull by selecting points from M^* to form the edges. In this case example, two inner points (shown as the two white points in Figure 3.27) are excluded. The remaining points, used to form the convex hull, are represented by the set M^{**}, which is

$$M^{**} = M^* \setminus \left\{ \begin{bmatrix} 0.288 \\ 0.551 \end{bmatrix}, \begin{bmatrix} 0.883 \\ 0.453 \end{bmatrix} \right\} \tag{3.7}$$

This selection results in a convex hull composed of 6 edges, as shown in Figure 3.27 (left). The points from M^* are used to update the edges of the hull, expanding the number of edges from 6 (convex hull) to 8, as shwon in Figure 3.27 (right).

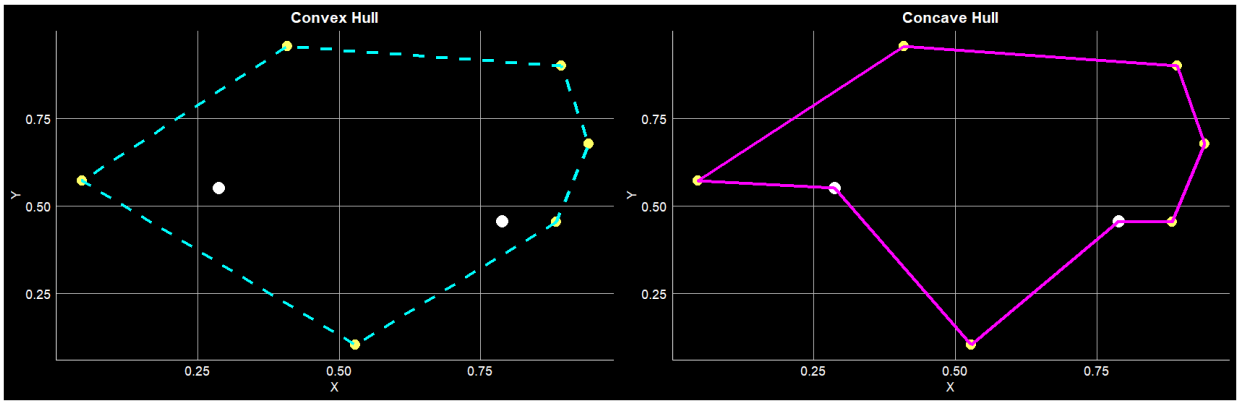


Figure 3.27: Figure showing left: the initial convex boundary prior to digging and right: the refined concave boundary iterations.

The updated edges form a concave hull, which allows for indentations that more accurately reflect the shape of the point set. The algorithm operates in two main phases. Initially, it processes the input data set M_i by generating its convex hull, and then it iteratively refines it to a concave hull [91]. A threshold value is chosen to guide the concavity refinement. This process continues until no major changes in the shapes are occuring.

Figure 3.28 demonstrates the intermediate stages, showing the evolving hull shape as the algorithm iteratively adjusts the boundary to better encapsulate the points, ultimately converging to the final concave hull form.

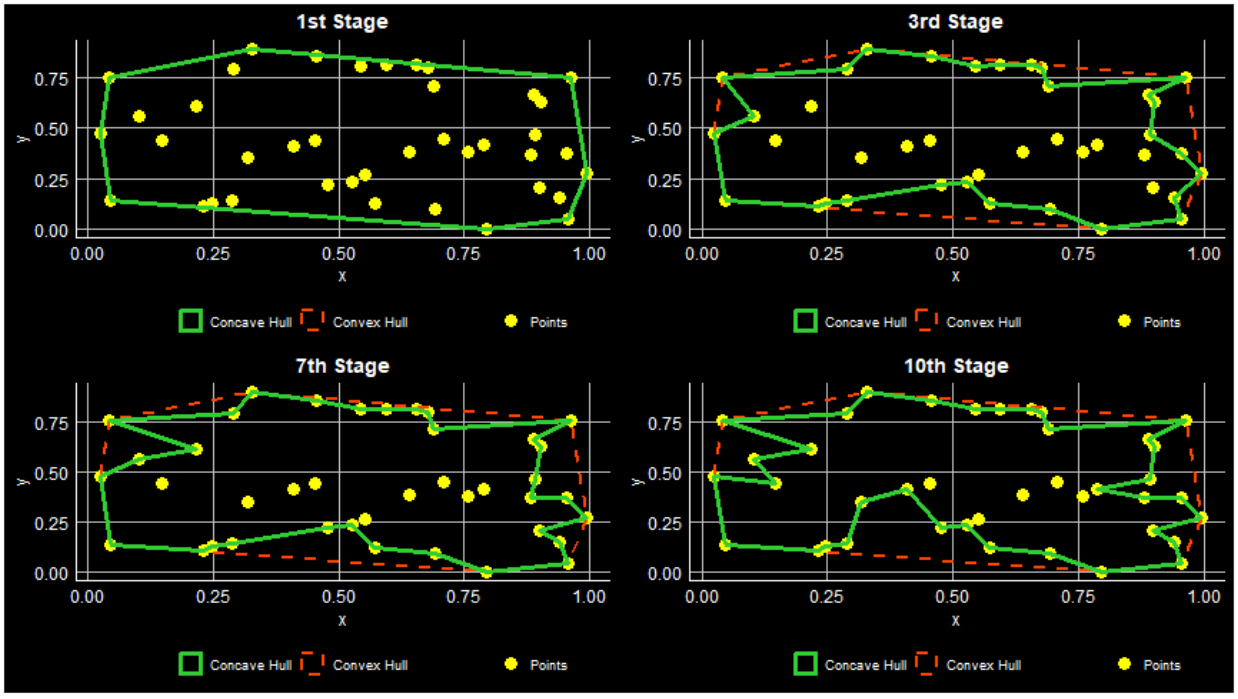


Figure 3.28: This plot illustrates the progression of concave hull shapes at four randomly choosen stages namely on the first, third, seventh and lastly the tenth stage.

3.3.4 Tessellation of the Domain and Conversion to Simple Features

In this final step, we require a standard object to represent the polygons or concave hulls that were obtained in section 3.3.3. We verify the validity of each polygon in the list, ensuring it has at least four vertices to form a closed, non-degenerate shape. We exclude invalid polygons that pose problems in the spatial analysis. After validation, we group the remaining polygons. This group of valid polygons is then converted into a standardised format, the simple features (sf) object, for compatibility with GIS tools and visualisation platforms [92, 93]. The result is a set of valid spatial features representing the tessellation, ensuring spatial integrity and usability.

The problem of invalid polygons becomes more significant in grid structures with fewer cells, as highlighted in Figures 3.29 and 3.30, which show 10×10 and 20×20 grid structures, respectively. Note in both figures the centroids are labelled. In grids with fewer cells, the concave hulls formed around the centroids tend to become degenerate due to a lack of sufficient points to properly define the boundaries of a polygon. This leads to a higher occurrence of invalid polygons (such as lines only and labelled centroids which are just assigned to the nearest generator point but do not form a polygon) that cannot be used in spatial tessellations. As the grid becomes finer, such as in the 20×20 grid, the issue of invalid polygons decreases because there are more points available to define each polygon boundary more accurately, leading to fewer degenerate shapes.

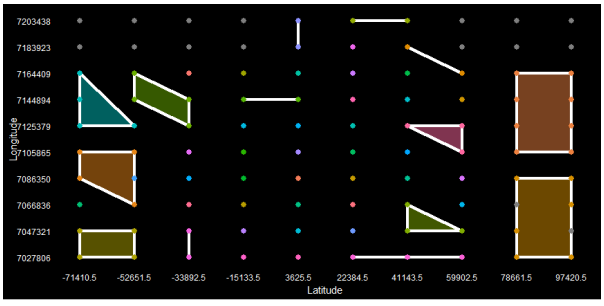


Figure 3.29: Grid structure with fewer cells (10×10), highlighting invalid polygons in the tessellation.

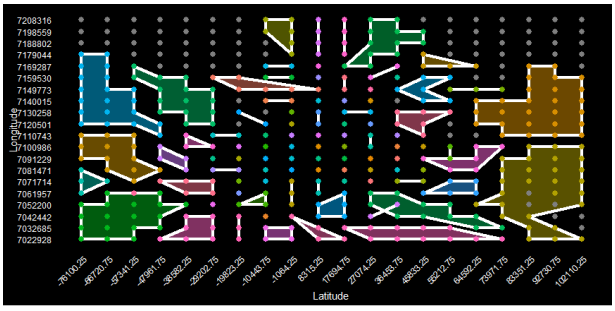


Figure 3.30: Grid structure with more cells (20×20), reducing invalid polygons in the tessellation.

In this process, each grid cell (or tile) represents a region of the plane, and the concave hull of the centroids within each tile is used to define the polygon for that tile. Mathematically, for each tile $i \in 1, 2, \ldots m$, the set of centroids assigned to it is denoted as M_i, and the concave hull of those centroids is represented as $\text{CH}(M_i)$. These concave hulls represent the boundaries of the tessellation tiles.

When a sufficiently fine grid is used, such as a 1500×1500 grid, the union of the concave hulls provides a highly accurate polygonal representation of the network Voronoi diagram. The overall tessellation can be

expressed as the union of these concave hulls:

$$\mathrm{V}(Q) = \bigcup_{i=1}^{m} \mathrm{CH}(M_i)$$

This means that by computing the concave hull for M_i, we are effectively defining the spatial boundaries of a Voronoi diagram that approximates the network structure, based on the centroids assigned to each grid cell. The finer the grid, the more precise and valid the tessellation becomes, reducing the occurrence of invalid polygons.

By following through this process, we obtain a set of polygons that make up the network Voronoi diagram. However, particularly in coarser grids, invalid polygons may arise. To address this, we filter out shapes that do not meet the geometric validity criteria discussed earlier in this subsection. Finer grids tend to result in better-defined polygons, thereby improving the overall accuracy of the tessellation and minimising the requirement for manual correction or filtering.

The described procedure results in an unbounded Voronoi diagram within a bounded region B, where the generator points are located. Typically, this region B represents the boundary of a map or another defined area of interest. Therefore, we focus on the bounded Voronoi diagram, denoted by $\mathcal{V}_{\cap B}$, defined as

$$\begin{aligned} \mathscr{V}_{\cap B} &= \{V(q_1), \ldots, V(q_m)\} \cap B \\ &= \{V(q_1) \cap B, \ldots, V(q_m) \cap B\} \end{aligned} \tag{3.8}$$

This set represents the Voronoi diagram constrained by the boundary points of B.

3.4 Conclusion

This very important chapter, introduced a novel approach for creating network Voronoi Diagram polygons. In this chapter were able to complete with discussions on the theoretical framework, as well as the computational methods and algorithms for Euclidean Voronoi diagrams and network Voronoi diagrams.

Chapter 4

Similarity Tests

This mini-dissertation aims to quantify how the structure of road networks within a precinct affects how accessible that police station is. Consider two road networks in Figure 4.1 within two different areas. The left example has a more regular road structure, as opposed to the irregular road layout structure on the right. The design of these road networks has an effect on both the distribution of crime occurences and the effectiveness of police response. This section introduces spatial similarity tests that will be used in the application section to evaluate how variations in road network layouts affect the accessibility of police stations.

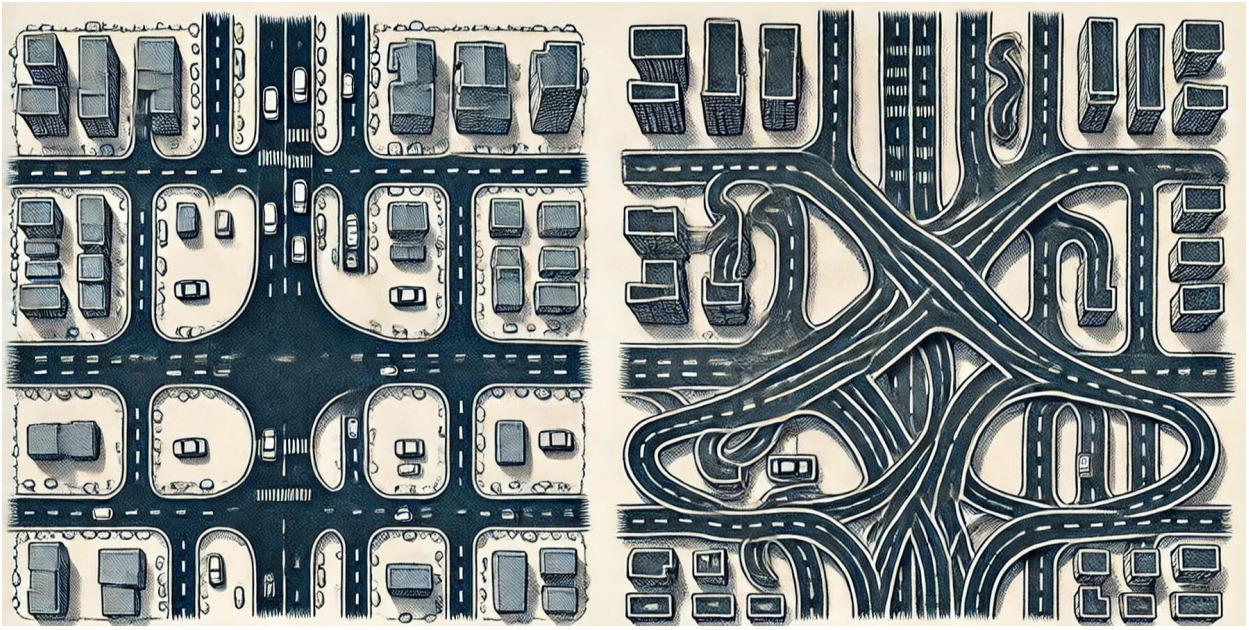


Figure 4.1: An example image created by DALL-E 3 [88], showing two types of road network designs.

4.1 Polygon Similarity

The polygons considered in Chapter 3, namely the network Voronoi polygons, Euclidean Voronoi polygons, and the administrative precinct polygons, are classified as lattice data [106]. Figure 4.2 illustrates the basic concept of polygon overlap between two polygons, A and B. We are interested in the proportion of overlap between the two polygons, which measures their similarity in terms of coverage.

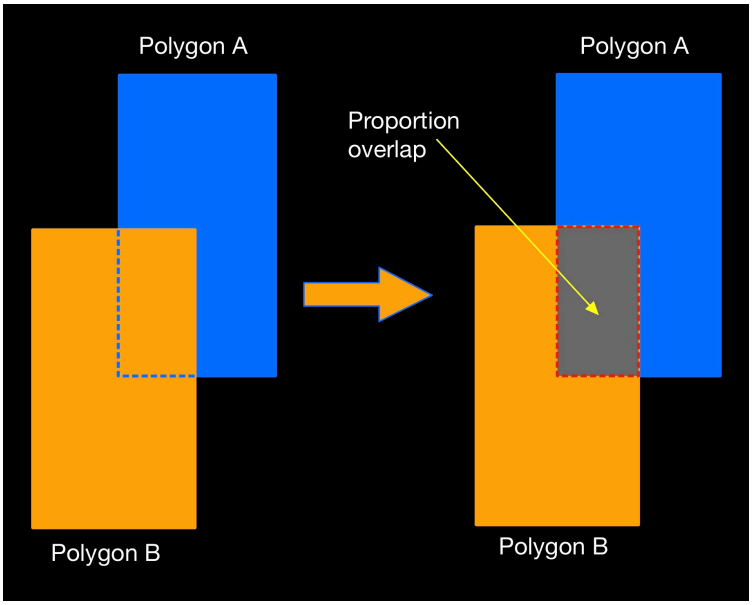


Figure 4.2: Illustration of two overlapping polygon layers.

Let r_i denote a spatial region contained within the spatial domain W, which is a subset of a q-dimensional space. We introduce a discrete spatial process $S(r)$ [27] defined as

$$S(r) = \{S(r_1), S(r_2), \ldots, S(r_n)\} \tag{4.1}$$

where r_i represents the set of coordinates $(r_{i1}, r_{i2}, \ldots, r_{iq})$ that span the spatial region i, and $S(r_i)$ denotes the values observed at spatial region r_i. We apply a specific instance of the generalised spatial similarity test in [61], tailored for irregular lattice data. This involves conducting similarity tests among the three distinct polygon structure data sets on the pairs A_{1ik} and A_{2ik} for $i = \{1, 2, 3\}$ and k is the total number of polygons being compared. The initial step involves generating pixel image representations B_{1ik} and B_{2ik} respectively. To create these representations, the spatial domain W is partitioned into an $m \times m$ grid, where each grid cell corresponds to a pixel, resulting in $M = m^2$ pixels.

We denote spatial locations by the vector $\mathbf{s} = (s_1, s_2, \ldots, s_M)$, with each spatial location element in $\mathbf{s}$ assigned the corresponding pixel value. For illustrative purposes, consider Figure 4.3 where W is enclosed within the smallest rectangular window and divided into pixels. Each pixel that falls within a given polygon is given a value of 1. Specifically, if a centroid s_j is contained in r_i, it takes the value $S(r_i)$. Note $S(r_i)$

has discrete values from 1 to m. Pixels falling outside the domain W are assigned an arbitrary value of NA. Note that, even though the polygons in Figure 4.3 are rectangular, we do not necessarily deal with rectangular polygons. The domain is the union of the two polygons being compared. Subsequently, a local similarity map is constructed to indicate local similarity for each pixel based on the pairwise comparison of B_{1ik} and B_{2ik}. Since the pixel values are categorical and discrete, comparison is straightforward: if a pixel at a particular position in B_{1ik} matches the corresponding pixel in B_{2ik}, the similarity map assigns a value of 1 to that position; otherwise, it assigns a value of 0. These local similarity values are stored, from these local similarity values, the global similarity is then computed as the proportion of pixels that match between the two images [61]. Figure 4.3 schematically illustrates the algorithm's process. It compares two different polygon types from the maps by analysing corresponding pixel positions in the two images. A similarity map (binary mask) is generated with values based on whether the pixels match. In the final step, the similarity values for all pixels are stored for the two polygons being compared. Note, Figure 4.3 highlights two examples where different boundary types are extracted from the maps with all the polygons.

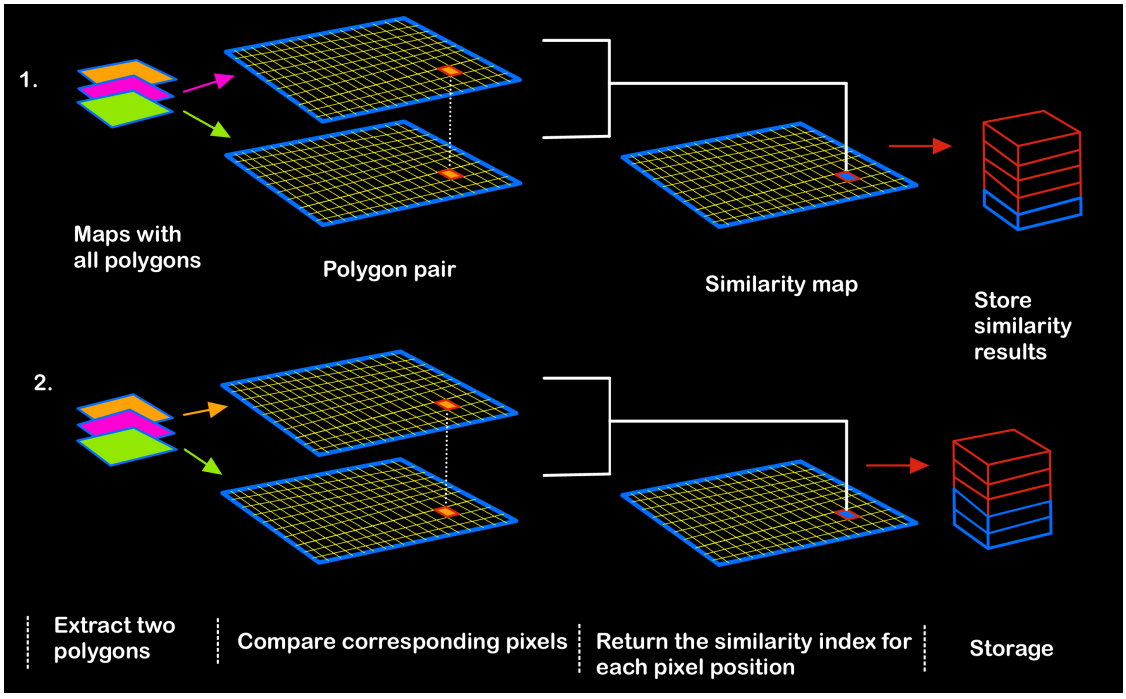


Figure 4.3: Illustration of spatial overlay analysis showing two examples of polygon boundary extraction and pixel-based similarity mapping with grid overlays.

4.2 Network Similarity Test

In this section, we aim to further evaluate the spatial similarity of road networks within two distinct polygon types to assess the road network coverage. The objective of the test is to determine the degree of similarity between the two pairs of linear networks, marked as N_{1ik} and N_{2ik}, for network types $i = \{1, 2, 3\}$ and k is the total number of networks being compared. This similarity will be represented through a local similarity

map that illustrates both the spatial distribution and the level of similarity inside a specific polygon.

When deriving or extracting these networks from three polygon types, the following relations (as shown in Figure 4.4) between the boundaries and the linear networks exists [85],

- The network entirely lies inside the polygon (scenario A in Figure 4.4).
- The network entirely lies outside the polygon (scenario B in Figure 4.4).
- The network lies inside the polygon but it intersects with the polygon boundaries (scenario C in Figure 4.4).

In our methodology, scenario A and C are more appropriate, in the case of scenario C, we only retain parts of the network that completely lie inside the polygon. Scenario B is excluded because our analysis involves pairwise comparisons of networks with their respective precincts or polygons.

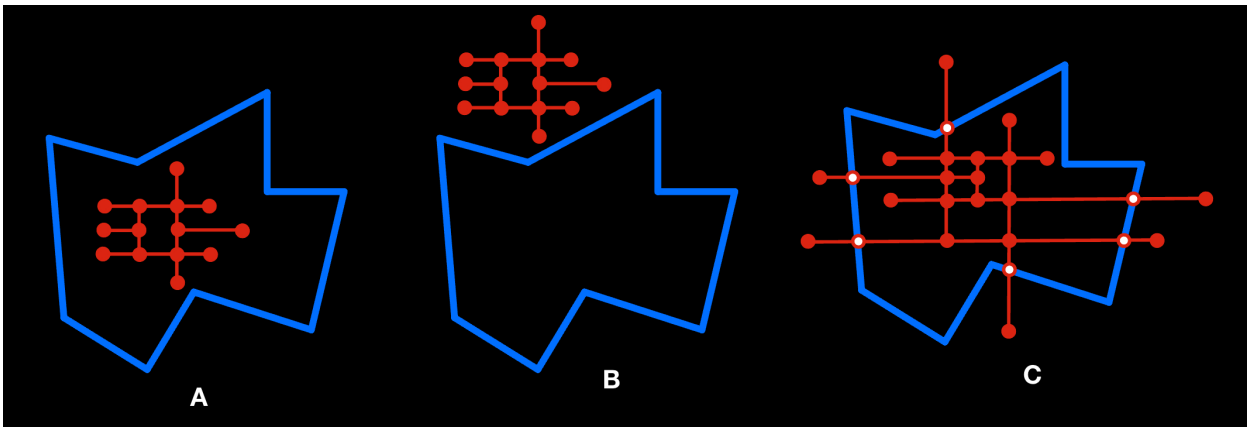


Figure 4.4: An illustration of the relations between polygons and the network. In A, the network is entirely contained within the polygon. In B, the network is entirely outside the polygon. In C, the network is intersecting the boundary segment of the polygon. The red circles filled with white indicate points of intersection.

The methodology for the similarity is adapted from the generic spatial similarity test outlined in [61], which has been further refined for application to linear networks in [26]. The test is conducted on the three different network structures derived from three distinct polygon configurations,

- The administrative precinct boundaries,
- The Euclidean Voronoi diagram boundaries,
- The network Voronoi diagram boundaries.

We conduct three pairwise comparisons. For clarity, we will describe the methodology for only one pairwise comparison, as the process is analogous for the other two. Sections 4.2.1 to 4.2.4 provide an overview of each step involved in the similarity test.

4.2.1 Spatial Transformation: Graph-Embedded Linear Network to Discrete Point Pattern

The generic spatial similarity test in [61] mandates a standardised pixel image format for all spatial data sets. Consequently, linear networks must be converted to a suitable pixelated data type prior to the application of the linear network spatial similarity test. Consider a linear network represented by a graph existing in the two-dimensional Euclidean space, $\mathbb{R}^2$. Such a graph, denoted as $G = (V, E)$, consists of a set V of n vertices and a set E of m edges, where each edge corresponds to a line segment connecting two vertices [5]. We assume that $n \in \mathbb{N}$ and $m \in \mathbb{N}$, where both $n > 1$ and $m > 1$. The linear network L is geometrically described as the set of all its line segments/ edges

$$L = \bigcup_{i=1}^{m} e_i$$

For any two vertices $v_i, v_j \in V$, the line segment e_i connecting them can be parameterised by

$$e_i(\alpha) = \alpha v_i + (1 - \alpha)v_j, \quad \alpha \in [0, 1]. \tag{4.2}$$

This parameterisation enables the representation of any points of the line segment as the convex combination of its end points. This gives a continuous representation of the entire connection of the result network. For the spatial analysis, a continuous linear network L must be transformed into a point pattern. This is a change in which the line segments e_i are discretised into a finite number of points. One known way is equal space sampling of points within each line segment in the given network. Taking into account a desired discretisation density $\delta > 0$, then the set of the sampled points on each line segment e_i can be represented as

$$P_j(e_i) = \{e_i(\alpha_t) \mid \alpha_t = t \cdot \delta,\ t = 0, 1, 2, \ldots, \lfloor 1/\delta \rfloor\}. \tag{4.3}$$

The union of sampled points over all line segments forms the discrete point pattern P_j, which corresponds to the original network from one of N_{1ik} or N_{2ik} mentioned earlier

$$P_j = \bigcup_{i=1}^{m} P_j(e_i). \tag{4.4}$$

The parameter δ plays a critical role when it comes to the level of precision of this discretisation. The parameter controls the balance between the computational costs and the level of details in the spatial structure. Converting the linear network into a point pattern facilitates the use of similarity tests in Sections 4.2.2 to 4.2.4. This approach is important as it captures the spatial patterns and relationships in the analysis [5].

4.2.2 Rasterisation: Point Pattern to Pixel Mapping

After the process of converting the continuous linear network into a discrete set of sampled points s_i, the next step is converting this point pattern into a pixel-based representation for the spatial similarity test in 4.2.3. Rasterisation is the process that transforms the finite set of points, into a grid of pixels. An image is formed when these pixels are displayed together [36, 16]. We approach this step with caution as there are potential limitations of the pixel-based representation such as the loss of spatial precision. Rasterisation introduces discretisation errors, as continuous point patterns are forced into a grid structure, potentially leading to loss of fine spatial detail, especially at lower resolutions [8]. Another limitation is that of boundary effects. Even with edge correction techniques, rasterisation can introduce biases near spatial boundaries, affecting density estimation. This issue is adressed by introducing the edge corrections [33, 9].

Consider two point patterns S_1 and S_2. These are converted to two pixel images, A and B, respectively. Both point patterns are enclosed within the same spatial domain or window. This is the smallest bounding box containing both patterns to ensure no edges are truncated and we compare images in the same region. The window is divided into $m \times m$ grid, where m is the chosen resolution [8]. The choice of grid resolution m determines the quality of the resulting image. A higher value of m leads to a higher resolution, preserving more spatial detail from the original point pattern but at the cost of increased computational complexity. Conversely, a smaller m simplifies the representation but may result in the loss of finer spatial details from the network [77, 122]. Note, we take the intersection of the grid with the union of the two point patterns, discarding any pixels outside the domain. This increases accuracy as pixels outside the window are not included in the similarity tests.

Each sampled point in the point pattern S_k is assigned to the appropriate pixel within the grid. A sampled point $s_i \in S_k$ with coordinates (x_i, y_i) is mapped to the pixel containing those coordinates. The value of each pixel represents the intensity of the underlying spatial point process. It measures the expected number of points in a given area and thus serves as a density estimate [59]. For a small area d_s, where $s \in d_s$, $|d_s|$ is the area of d_s, and $N(d_s)$ is the number of points within d_s. The intensity function at any centroid S is expressed as [8, 77]:

$$\lambda(s) = \lim_{|d_s| \to 0} \frac{\mathbb{E}[N(d_s)]}{|d_s|}. \tag{4.5}$$

To estimate the intensity function, we use Diggle's non-parametric kernel estimator, which applies kernel smoothing across the point pattern [33, 122, 16]:

$$\hat{\lambda}_D(s_i) = \sum_{j=1}^{n} \frac{\kappa(s_i - y_j)}{\int_{\mathbb{R}^2} \kappa(y_j - p_k)\, dp_k}. \tag{4.6}$$

where $\kappa(\cdot)$ is the bivariate Gaussian kernel function:

$$\kappa(d) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left(-\frac{1}{2} d\Sigma^{-1} d'\right),$$

and the denominator of the equation is termed the edge correction factor. The kernel smoothing process is applied to all points y_j, and Diggle's edge correction ensures accurate intensity estimation near the window boundaries [33].

4.2.3 Spatial analysis: Local Similarity Mapping

The important step after converting point patterns to densities (pixel images) is local similarity mapping. This approach compares the two pixel images locally to find similarities and differences and return similarity maps. In this context, the Structural Similarity Index (SSIM) is used for measuring the similarity between two images [61]. In contrast to basic intensity-based metrics (such as MSE, Peak Signal-to-Noise Ratio [62], Sum of Absolute Differences [73], and Normalised Cross-Correlation [131]), SSIM incorporates structural information. SSIM also considers how pixel intensities are structured. This makes it more suitable for spatial patterns where texture, contrast, and luminance are important. SSIM evaluates these properties across small, local windows in the rasterised images, providing a detailed map of similarity scores [123]. Figure 4.5 shows how SSIM combines all the three different measures, to come up with the final similarity measure (score).

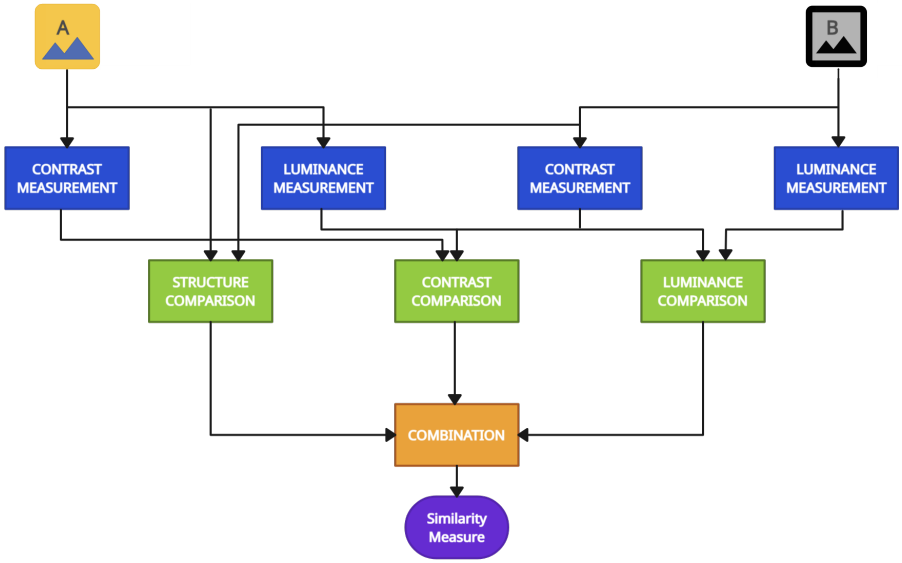


Figure 4.5: Illustrating the SSIM architecture, the system demonstrates the process of evaluating image quality by comparing luminance, contrast, and structural components. It highlights how these factors are integrated to compute the similarity score.

Figure 4.6 is showing the sliding window mechanism used to compute the SSIM. SSIM is a localised metric

designed to assess the similarity between corresponding small regions (windows) of pixels extracted from Image A and Image B. The index is a value ranging from -1, for perfect dissimilarity, to 1, for identical windows. To generate an SSIM quality map, the sliding window is systematically moved across all pixels of both image maps. The amount of detail that the SSIM index captures depends on the sliding window's dimensions. Smaller windows yield fine-grained details compared to larger windows [123].

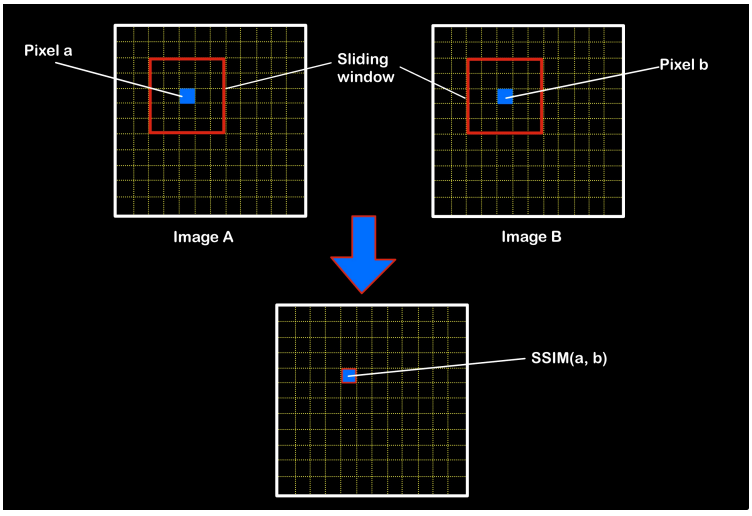


Figure 4.6: SSIM sliding window mechanism. It illustrates how a sliding window is used to compare small regions of two images, generating a similarity score for each position.

SSIM values range from -1 to 1, where 1 indicates perfect similarity, 0 indicates no similarity, and negative values indicate dissimilarity. For local similarity mapping, the SSIM score is computed for each small local window in the images A and B, over the sliding window. The result is a matrix of SSIM values, which can be visualised as a heatmap, as shown in Figure 4.7. In this heatmap, regions with high SSIM scores are highlighted (by yellow), indicating areas where the two patterns are similar. Regions with low SSIM scores are identified by blue.

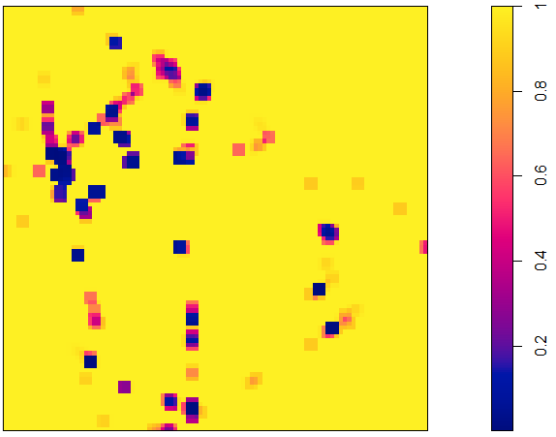


Figure 4.7: A similarity map example, showing regions of high scores as yellow, and regions of low scores as blue.

The metrics for comparing two images, A and B, are outlined in Table 4.1. Two means, two standard deviations, and one covariance value are calculated from the pixels in the sliding windows of the images, based on the formulas in Table 4.1. Using these statistical values, the luminance, contrast, and structural comparisons are calculated, resulting in the SSIM between images A and B in Equation (4.7). The SSIM between two pixel images A and B at location (a, b) in Figure 4.6 is defined as

$$SSIM(A,B) = \frac{(2\mu_A\mu_B + C_A)(2\sigma_{AB} + C_B)}{(\mu_A^2 + \mu_B^2 + C_A)(\sigma_A^2 + \sigma_B^2 + C_B)} \tag{4.7}$$

where C_A and C_B are constants used to stabilise the division, especially when the denominator is close to zero [2].

Table 4.1: Statistical measures and formulae used for SSIM.

Statistical Measure	Description	Formula
μ_A	Mean pixel intensity of image A	$\mu_A = \frac{1}{M}\sum_{i=1}^{M} A_i$
μ_B	Mean pixel intensity of image B	$\mu_B = \frac{1}{M}\sum_{i=1}^{M} B_i$
σ_A^2	Variance of image A pixel intensity	$\sigma_A^2 = \frac{1}{M-1}\sum_{i=1}^{M}(A_i - \mu_A)^2$
σ_B^2	Variance of image B pixel intensity	$\sigma_B^2 = \frac{1}{M-1}\sum_{i=1}^{M}(B_i - \mu_B)^2$
σ_{AB}	Covariance between images A and B local windows	$\sigma_{AB} = \frac{1}{M-1}\sum_{i=1}^{M}(A_i - \mu_A)(B_i - \mu_B)$

4.2.4 Spatial Analysis: Computing Global Similarity Metrics

After conducting local similarity mapping, the next step is to compute global similarity metrics that summarise the overall similarity between two spatial point patterns. We obtain the global SSIM score, by averaging the local SSIM values across the entire similarity map [123]. Assuming there are M local windows in the image and the local SSIM value computed at each window i is $SSIM(A(i), B(i))$. The global similarity score is expressed as

$$G = \frac{1}{M}\sum_{i=1}^{M} SSIM(A(i), B(i)) \tag{4.8}$$

4.3 Conclusion

In this chapter, we described the similarity tests used to assess the amount of overlap between different types of polygon boundaries. The statistic we obtain from these tests is the proportion of overlap. We also provided a detailed explanation of the metrics used to quantify the differences in the structures of the networks or subnetworks extracted from the different types of polygonal boundaries.

Chapter 5

Application

In this section, we apply the methodology outlined in Chapters 3 and 4 to selected regions in South Africa. We begin by creating two alternative boundary types to replace the official administrative boundaries: boundaries based on the Euclidean and network Voronoi diagram polygons. Subsequently, we compute spatial similarities between these three types of polygon boundaries and the networks derived from them.

The data set used in this section is based on those described in Chapter 2. The first data set is used for illustrative purposes, while the second, focusing on Khayelitsha, is used for validation. This validation data set is used to examine the relationship between accessibility metrics and crime rates. Specifically, it evaluates how spatial accessibility to police stations affects the distribution of crime and explores whether under-reporting of crimes is linked to limited accessibility to police stations.

For precincts with low similarity values (indicating low accessibility), we analyse the proportional change in the number of crimes reported within those precincts following boundary modifications. If there is a decrease in reported crimes (a negative proportional change) within the new boundaries generated by Voronoi models, it would support our hypothesis. Such a decrease would suggest that a significant portion of crimes is now being reported to other, nearer, and more accessible police stations.

5.1 Polygon Boundaries

Recall the methodology discussed in Section 3.3 to fully understand how this refinement was implemented. As shown in Figures 5.2 and 5.1, the diagrams were created using the `spatstat` package, following the algorithm outlined in the Chapter 3. Since we have access to police station data for the entire of South Africa, a Voronoi diagram was constructed based on the positions of all the police stations, which can be seen in Figure 5.2. This diagram represents a tessellation of the South African map, with each region determined

by the Euclidean distance to the nearest police station. Figure 5.1 presents the natural geographic boundaries of South Africa.

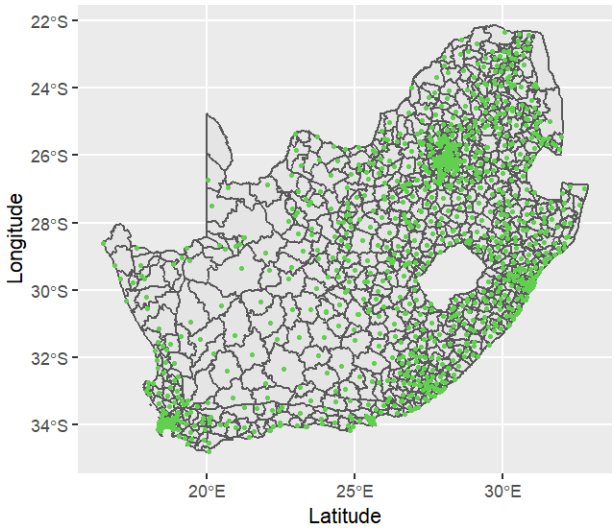


Figure 5.1: Map of South Africa showing natural geographic boundaries for comparison with the Voronoi diagram.

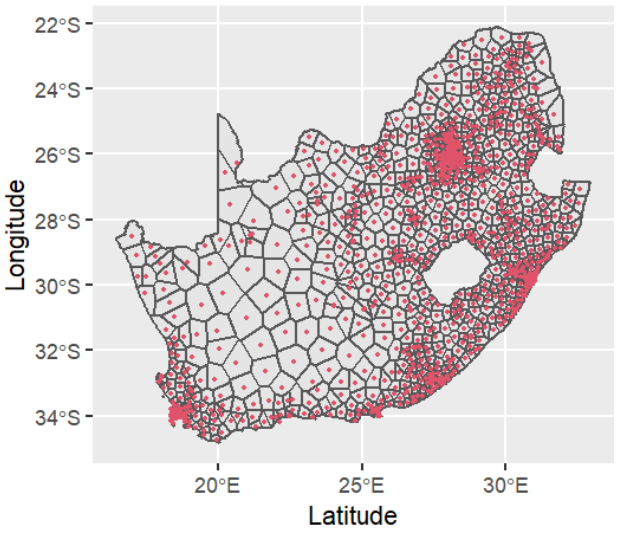


Figure 5.2: Euclidean Voronoi diagram based on police station locations, covering the entire map of South Africa.

Figures 5.3 and 5.4 present data specific to the Gauteng district, highlighting the police precincts in this area. Figure 5.3 shows the administrative boundaries of Gauteng. In contrast, Figure 5.4 illustrates the Euclidean Voronoi diagram, where each region is determined by its nearest police precinct based on Euclidean distance.

Figure 5.3: Map of Gauteng showing administrative boundaries of the districts.

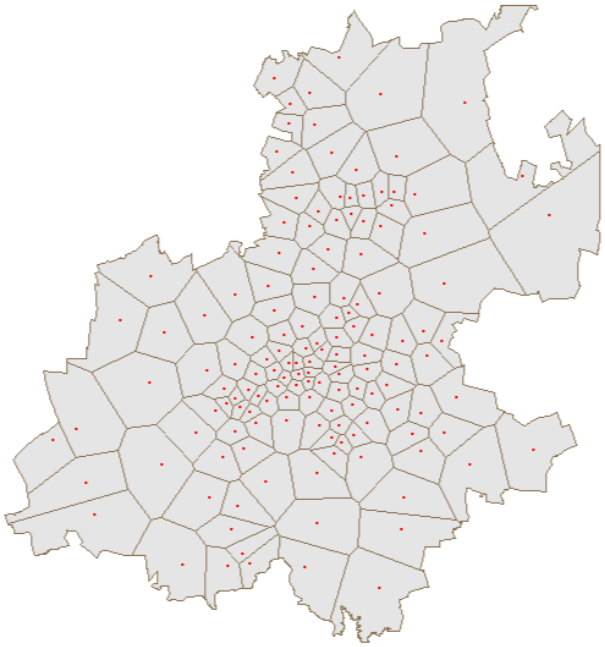


Figure 5.4: Voronoi diagram based on police precinct locations in Gauteng, with each region reflecting the nearest police precinct by Euclidean distance.

Figure 5.5 shows the tessellated road network of Gauteng. This tessellation only includes specific road types,

such as "motorway", "primary", "secondary" and their links. Using this subset of road as a start, ensures a balance between computational performance and a meaningful road network representation. Figure 5.5 illustrates the resulting road network tessellation. The corresponding polygons formed from this network tessellation is shown in Figure 5.6. The polygons can be thought of as areas that are closest to generator points (police stations) within the road network, based on network distance. The omission of some road types, however, limits the tessellation's overall accuracy and may mean that it fails to correctly represent spatial structure in the area.

Figure 5.5: Network-based tessellation of Gauteng, created using a subset of the road types.

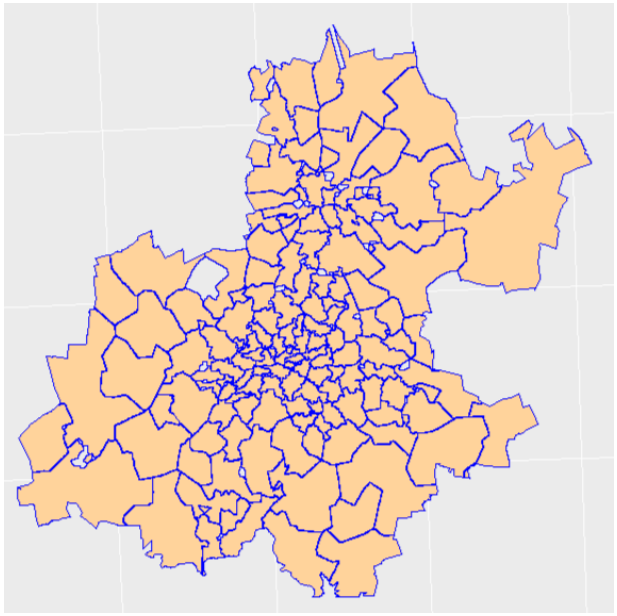


Figure 5.6: Polygons generated from the network tessellation, showing the areas closest to each point via the road network.

Including smaller roads adds complexity and increases computational costs but improves analysis. However, we limit our analysis to five police station locations and their respective precincts, while using a denser road network, that is smaller subset of the area. Although we work with fewer points, the reduced data set will still provide ample opportunity for an interesting discussion, given the current computational limitations.

Figure 5.7 illustrates the administrative boundaries of the five chosen police stations. In contrast, Figure 5.8 presents the Euclidean-based Voronoi diagram for the same five police stations. This diagram divides the region based on proximity to each station, using Euclidean distances.

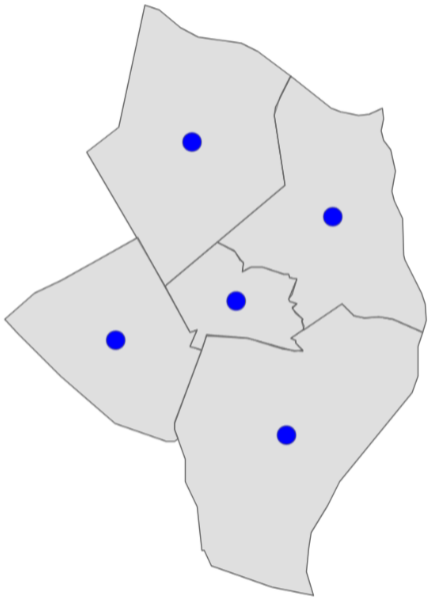


Figure 5.7: Administrative boundaries of the five selected police precincts.

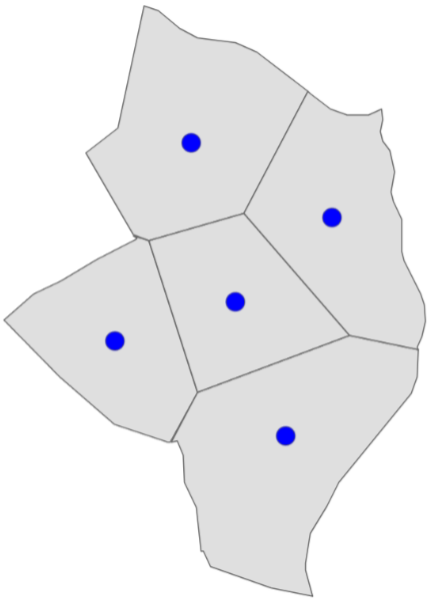


Figure 5.8: Euclidean-based Voronoi diagram for the five selected police station points.

Figure 5.9 shows the network tessellation for the five contiguous police stations, using a portion of the road network that connects these stations. The tessellation is based on network distances, providing a more representative measure of proximity, when compared to Euclidean distance. Figure 5.10 presents the polygons generated from this network tessellation. These polygons define the areas closest to each police station based on the road network. Notably, the polygons have an outer boundary, which we defined as the union of the administrative boundaries' outermost edges. This outer boundary cuts out parts of the network that lie outside the precincts, ensuring the tessellation is focused on the relevant areas.

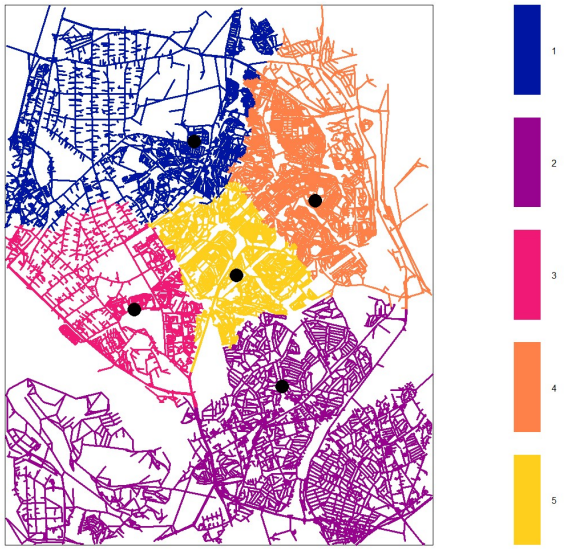


Figure 5.9: Network tessellation of the road network for the five contiguous police stations.

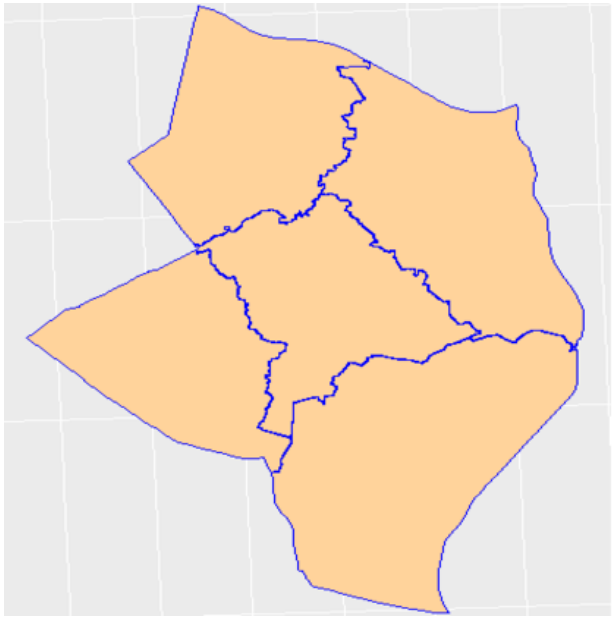


Figure 5.10: Network polygons based on the tessellation, with an outer boundary defined by the union of administrative boundaries.

The use of the communal outer boundary (on the area subset), while necessary for this analysis, compromises spatial similarity accuracy. It is defined by the outer edges of administrative boundaries. This limits the precision of the outer polygons. However, the inner polygons remain unaffected since their boundaries are formed within the network. When the analysis covers the entire area, this outer boundary will no longer be needed, and accuracy will be restored. So, when analysing a subset, extra caution needs to be taken into consideration when interpreting the similarity values.

5.2 Similarity Tests

In this section, we compare polygons and subnetworks derived from different polygon types using pixel-based image representations as discussed in Chapter 4. These representations allow us to generate binary masks that highlight areas of overlap between different polygon types and subnetworks. Pairwise comparisons are performed to assess the degree of similarity between polygons by evaluating the extent of their overlap. Figure 5.11 shows the five stations that share a common outer boundary, with all three different boundary types overlaid on top of each other. This image provides a visual impression of how the networks or polygons may differ.

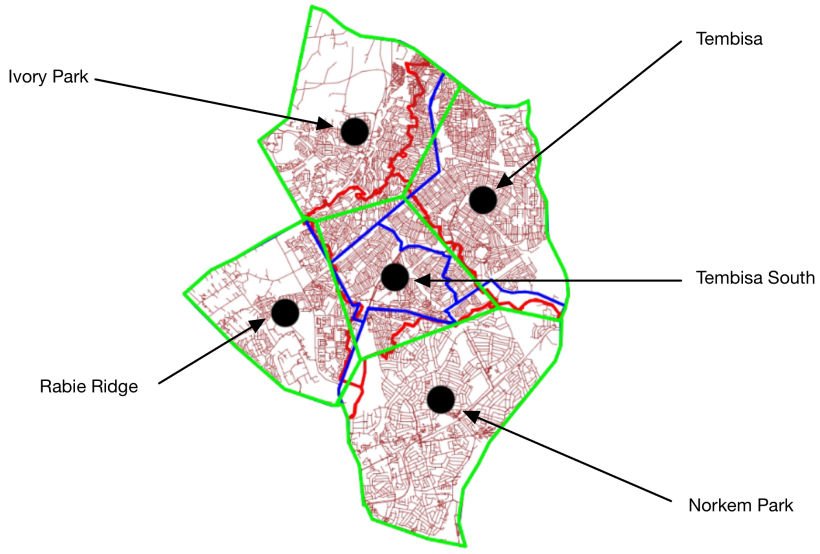


Figure 5.11: Overlaid boundaries of five contiguous police stations, enabling pairwise comparisons between different types of polygons.

Figures 5.12, 5.13, and 5.14 display the discrete pixel images used for this comparison. Figure 5.12 shows the administrative boundaries, while Figures 5.13 and 5.14 depict the Euclidean distance Voronoi diagram and network Voronoi diagram, respectively. These images represent the polygons with categorical pixel values corresponding to the five precincts, ranging from 1 to 5.

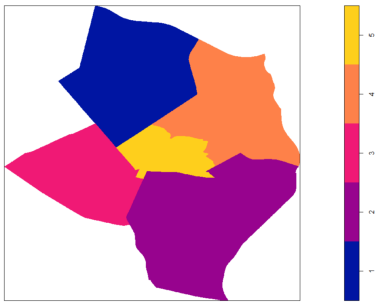


Figure 5.12: Discrete pixel image of administrative boundaries.

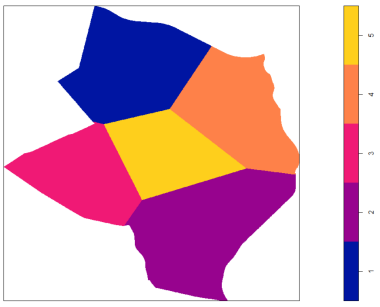


Figure 5.13: Discrete pixel image of the Euclidean VD.

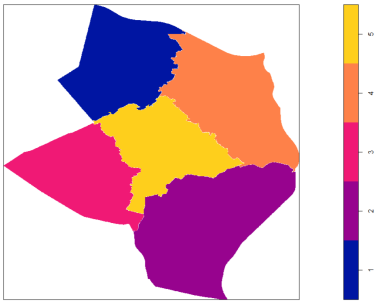


Figure 5.14: Discrete pixel image of network VD.

To quantify polygon similarity, we adopt the Jaccard Index (or Intersection over Union) [102], which is an appropriate metric for comparing the overlap between two sets. The Jaccard Index is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are the sets of pixels representing the two polygons being compared, $|A \cap B|$ denotes the number of pixels in the overlapping region, and $|A \cup B|$ represents the total number of unique pixels in both polygons. This index, ranging from 0 to 1, measures the degree of similarity, with 1 indicating perfect overlap and 0 indicating no overlap.

Figures 5.15 to 5.19 present the binary masks (similarity maps) resulting from pairwise comparisons of the polygons and subnetworks. Note these polygons are named A to E corresponding to the ID column in Table 5.1. These maps illustrate areas of overlap between the polygons. Yellow regions indicate areas of overlap, between the same pixels, and blue, the areas that do not.

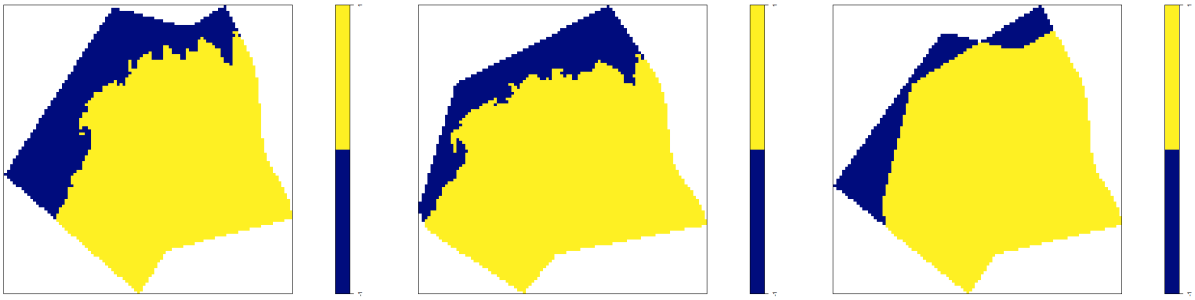


Figure 5.15: Polygon A binary masks, from left to right: Euclidean and administrative precincts similarity, network and administrative precincts similarity, and Euclidean and network similarity.

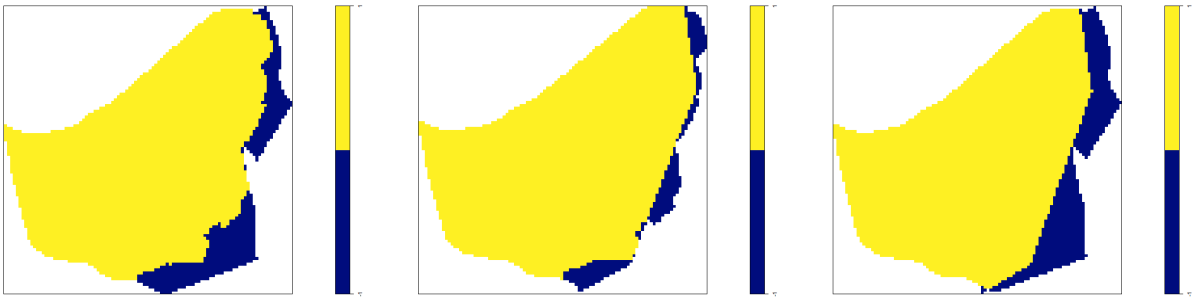


Figure 5.16: Polygon B binary masks, from left to right: Euclidean and administrative precincts similarity, network and administrative precincts similarity, and Euclidean and network similarity.

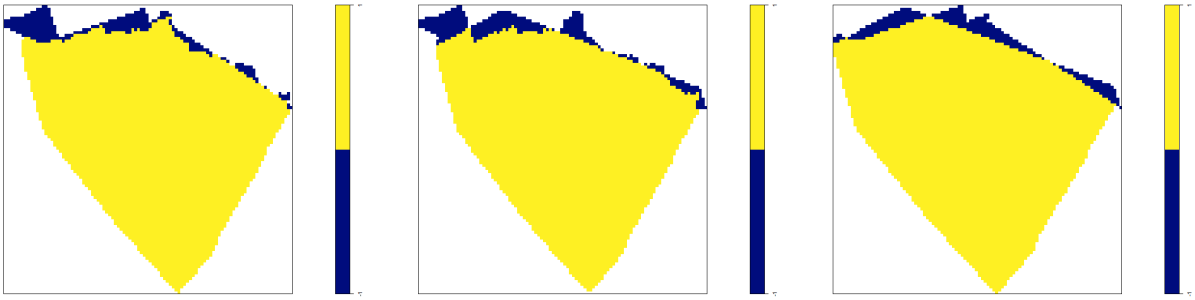


Figure 5.17: Polygon C binary masks, from left to right: Euclidean and administrative precincts similarity, network and administrative precincts similarity, and Euclidean and network similarity.

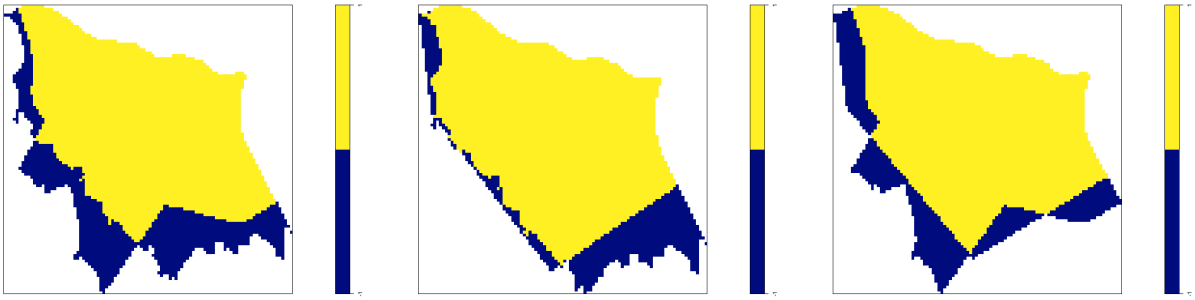


Figure 5.18: Polygon D binary masks, from left to right: Euclidean and administrative precincts similarity, network and administrative precincts similarity, and Euclidean and network similarity.

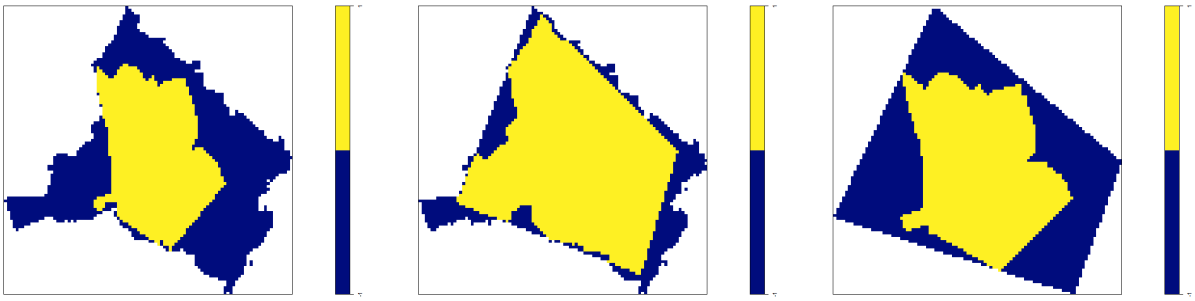


Figure 5.19: Polygon E binary masks, from left to right: Euclidean and administrative precincts similarity, network and administrative precincts similarity, and Euclidean and network similarity.

Table 5.1 shows the similarities scores across the polygons of different police stations. In the table, **NV** refers to Network Voronoi polygon, **EV** refers to Euclidean Voronoi polygon, and **Prec** represents administrative polygon. Based on Table 5.1 it can be observed that Rabie Ridge and Norkem Park have the highest spatial overlap among the three comparisons. Ivory Park and Tembisa have moderate similarity scores, suggesting a reasonable but not perfect polygon overlap. In contrast, Tembisa South displays significantly lower scores, highlighting a huge polygon non-overlap or challenges in defining its administrative and Voronoi boundaries. The lower similarity score or polygon overlap for Tembisa South is partly due to its location as a central precinct surrounded by the other four. Unlike the other precincts, it does not share the outer communal boundary with them, which further contributes to its lower similarity score.

ID	Police Station	NV vs Prec	NV vs EV	EV vs Prec
A	Ivory Park	0.7259	0.7921	0.8430
B	Norkem Park	0.8585	0.9159	0.8367
C	Rabie Ridge	0.9192	0.9089	0.9247
D	Tembisa	0.7173	0.7937	0.7599
E	Tembisa South	0.3957	0.7355	0.4501

Table 5.1: Comparison of the similarity among the three types of polygon boundaries per police staion.

Comparison	Similarity
Network Voronoi and administrative precincts	0.7233
Network and Euclidean Voronoi	0.8292
Euclidean Voronoi and administrative precincts	0.7629

Table 5.2: Comparison of the overall similarity among the three types of polygon boundaries.

To reduce redundancy in the analysis, we will only look into individual police stations in Section 5.3, but based on the similarity scores in Table 5.2 we can deduce the following

- The comparison between the network Voronoi and Euclidean Voronoi have a high similarity score. This indicates that the two border types are similarly aligned. Given that both network Voronoi and

Euclidean Voronoi models rely on spatial proximity, it is anticipated that they will provide similar borders. The similarity indicates that accessible areas defined by either model overlap significantly.

- Administrative precincts and Euclidean Voronoi have a moderate similarity score, which indicates some variance but also reasonable alignment. This is probably because precinct boundaries are influenced by administrative or jurisdictional divisions that do not soley rely on spatial proximity alone.
- The lowest similarity is among the pairs is between network Voronoi and the administrative precincts. This score reflects greater variation between network Voronoi boundaries and precinct boundaries. This discrepancy indicates that accessible areas defined by the precinct model might not align well with network-based areas, potentially affecting the consistency of police service coverage along road networks.

Recall from Section 4.2, different spatial relationships can exist between a network and a polygon, namely the network can intersect the polygon, be entirely contained within it, or lie completely outside of it. As we proceed, our aim will be to ensure that either the first, the second, or a combination of these first two conditions is met, depending on the case. This ensures the the network segments are contained in the respective polygons.

Figures 5.20 through 5.22 present the subnetworks for each precinct, depicting them against the three types of boundaries under consideration. To aid in visual clarity, the polygon boundaries are included alongside the subnetworks to illustrate how the network interacts with the polygon.

Figure 5.20: Subnetworks based on administrative boundaries for polygons A to E (from left to right).

Figure 5.21: Subnetworks based on Euclidean Voronoi boundaries for polygons A to E (from left to right).

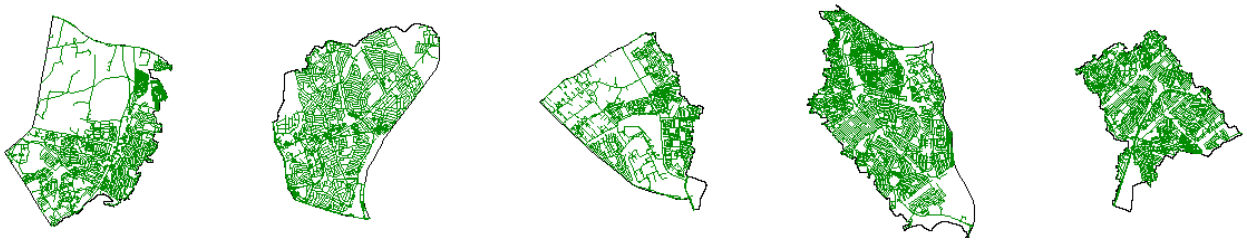


Figure 5.22: Subnetworks based on network voronoi for polygons A to E (from left to right).

In the next step (analogous to step 1 in Section 4.2.1), we transform the networks into point patterns, as the subnetworks have been extracted. This approach is exemplified using Polygon A (with a administrative boundary), and Polygon B (with a Euclidean Voronoi boundary). To compare the two linear networks from both polygons, we generate two regular, equidistant point patterns (P1 and P2, respectively). During the conversion, line segments in the network are replaced with equally spaced points. The number of points is fixed at $n = 10$ for each case. By keeping the value of n same for both point patterns, we guarantee that the comparison is unaffected by variations in point density. The equidistant point placement approach guarantees uniform distribution of points along the network's linear distribution of points along the network's line segments. An example of the resultant point patterns are shown in Figures 5.23 and 5.24.

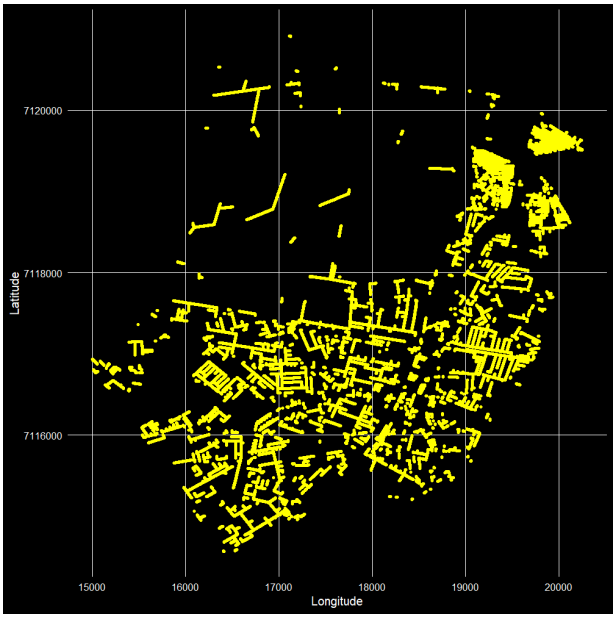


Figure 5.23: Point pattern for polygon A-administrative boundary subnetwork.

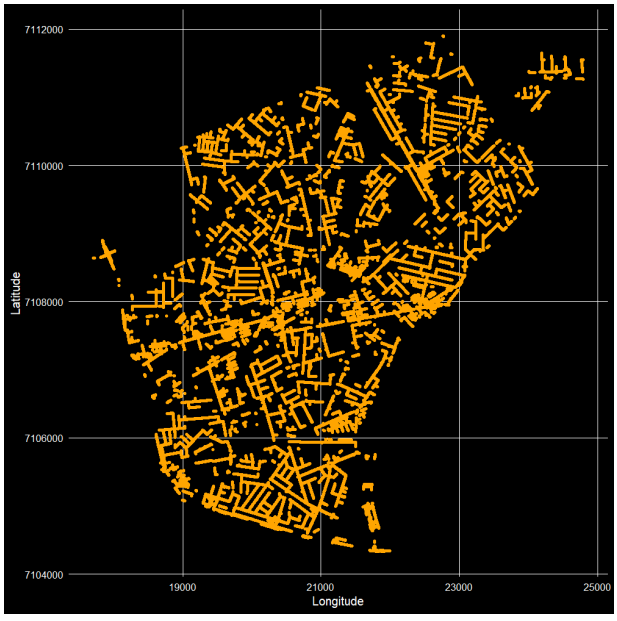


Figure 5.24: Point pattern for polygon B-Euclidean Voronoi boundary subnetwork.

After generating the point patterns for each polygon, the patterns are transformed into pixel-based images (density maps) by evaluating the intensity across the pixels (following the same procedure as step 2 in Section 4.2.2). Figures 5.25 and 5.26 provide examples of some of the resulting pixel images, each constructed at a resolution of $m = 100$. It is important to note that the boundaries shown represent the union of the two polygons being compared. This approach prevents zero densities outside the union, which would otherwise

inflate similarity values.

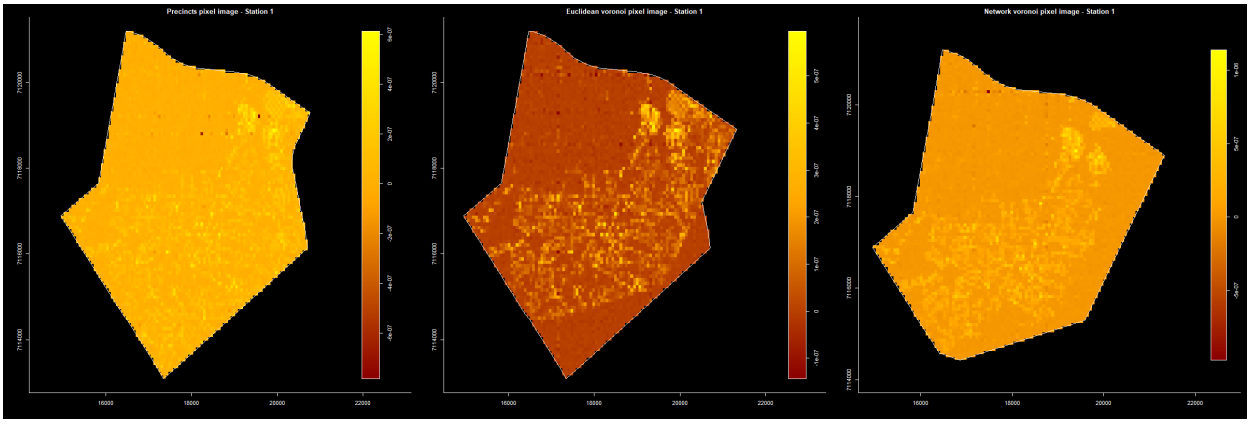


Figure 5.25: Density plots of polygon A for all three boundary types, used as pixel image representations in the similarity analysis.

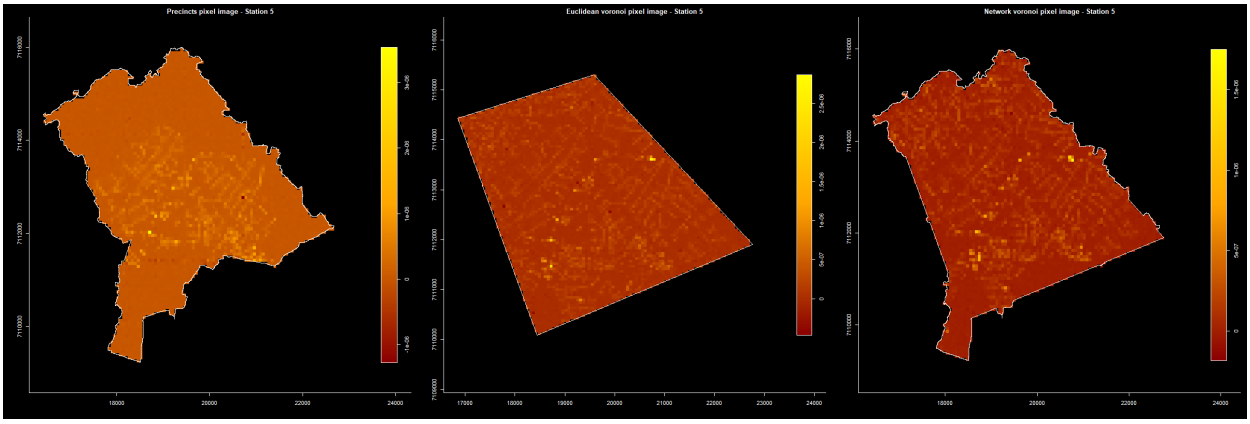


Figure 5.26: Density plots of polygon E for all three boundary types, used as pixel image representations in the similarity analysis.

The subsequent step (refer to step 3 in Section 4.2.3) involves calculating the local similarity [61] of the pixel images presented in Figures 5.25 to 5.26.

The local similarity maps, calculated to display pixel-level similarities across different boundary types for each polygon subnetwork, are shown in Figures 5.27 to 5.31. We now, analyse the local similarity maps with respect to the different boundaries for each polygon type. The yellow coloured parts of the similarity maps indicate regions where the two network structures, extracted from both polygon boundary types, are identical or in agreement. The red areas highlight locations where the road networks differ. Additionally, the similarity map returns a value of one (yellow) in areas where neither polygon contains a road network. In Figure 5.31 which corresponds to police station E, there is an example of very low correlation, as evidenced by the bluish pixels in the similarity map (first image).

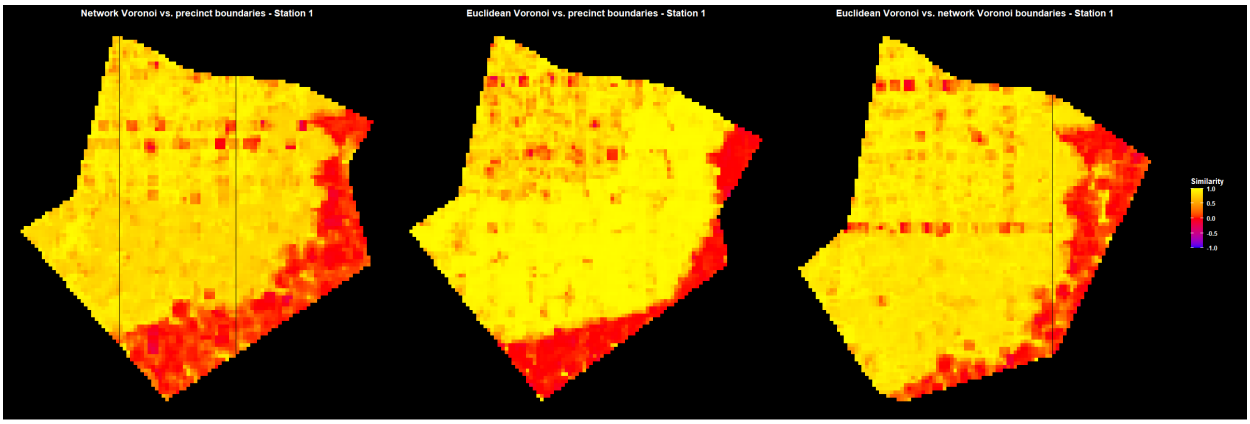


Figure 5.27: Police station A: Local similarity maps showing pixel-level agreement and differences between the networks of (i) Network Voronoi and administrative boundaries, (ii) Euclidean Voronoi and administrative boundaries, and (iii) Euclidean Voronoi and Network Voronoi boundaries.

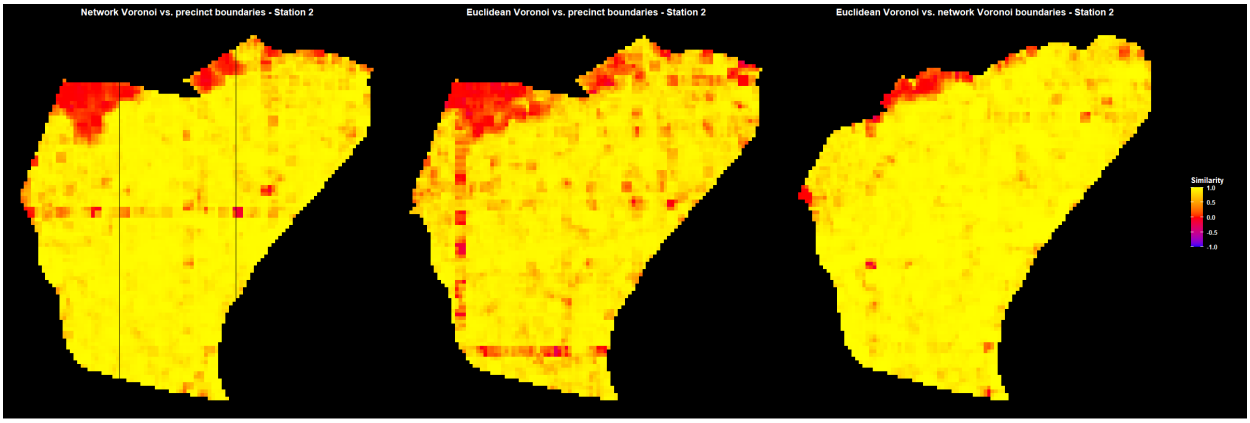


Figure 5.28: Police station B: Local similarity maps showing pixel-level agreement and differences between the networks of (i) Network Voronoi and administrative boundaries, (ii) Euclidean Voronoi and administrative boundaries, and (iii) Euclidean Voronoi and Network Voronoi boundaries.

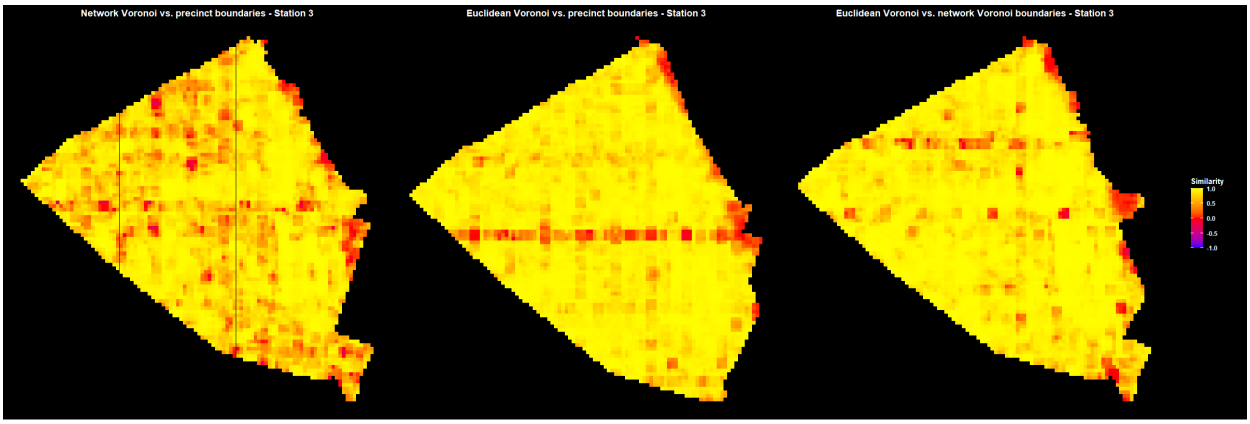


Figure 5.29: Police station C: Local similarity maps showing pixel-level agreement and differences between the networks of (i) Network Voronoi and administrative boundaries, (ii) Euclidean Voronoi and administrative boundaries, and (iii) Euclidean Voronoi and Network Voronoi boundaries.

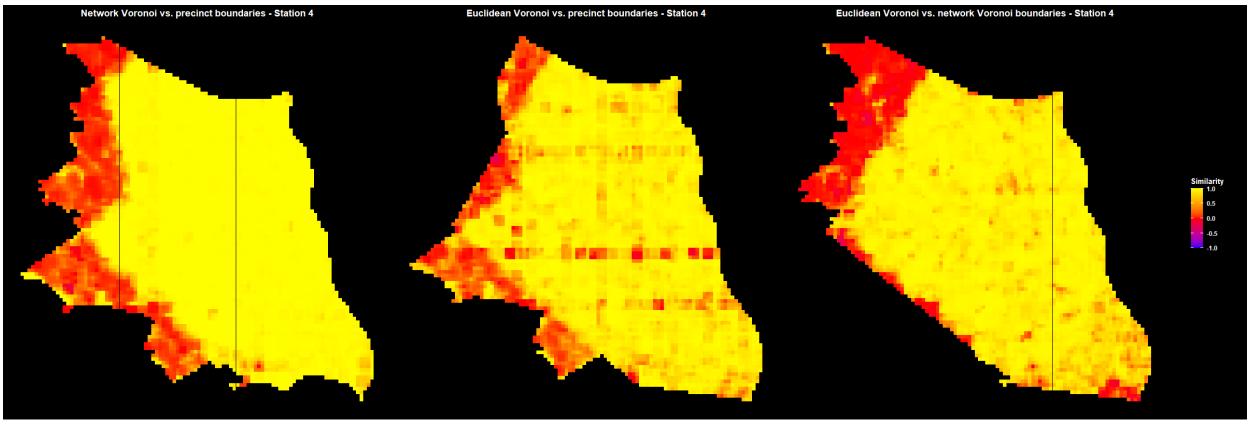


Figure 5.30: Police station D: Local similarity maps showing pixel-level agreement and differences between the networks of (i) Network Voronoi and administrative boundaries, (ii) Euclidean Voronoi and administrative boundaries, and (iii) Euclidean Voronoi and Network Voronoi boundaries.

Figure 5.31: Police station E: Local similarity maps showing pixel-level agreement and differences between the networks of (i) Network Voronoi and administrative boundaries, (ii) Euclidean Voronoi and administrative boundaries, and (iii) Euclidean Voronoi and Network Voronoi boundaries. Yellow regions indicate agreement, red regions indicate differences, and bluish regions highlight low correlation. The first similarity map shows an example of very low correlation, as shown by the bluish pixels.

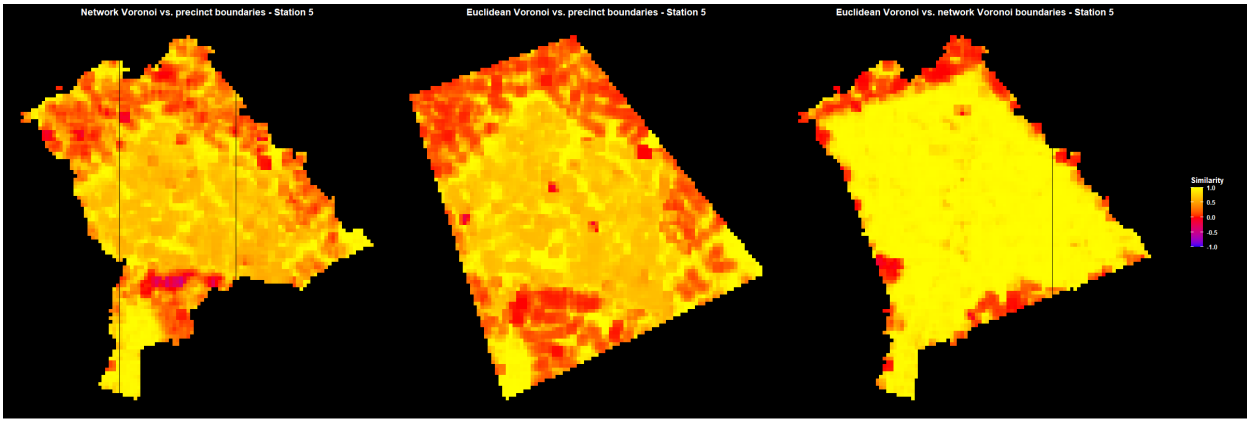


Table 5.3 summarises the global similarity values for all the local similarity maps. These values are also used in the creation of the dot plot in Figure 5.32 for the network comparison.

ID	Police Station	NV vs Prec	EV vs Prec	NV vs EV
A	Ivory Park	0.6200	0.6974	0.7450
B	Norkem Park	0.8522	0.9031	0.7729
C	Rabie Ridge	0.7604	0.8599	0.8541
D	Tembisa	0.7954	0.7506	0.7694
E	Tembisa South	0.5675	0.8260	0.5629

Table 5.3: Global similarity values for the five police stations and the three comparisons.

To quantify accessibility to police stations based on the similarities across the different boundary types, we use similarity metrics to measure how well networks extracted from each boundary type captures the

accessibility to police services. Accessibility here is defined as the extent to which local similarity metrics correlate with proximity to police stations. A higher similarity within a region indicates that the police station effectively serves that area, implying a high accessibility.

A visual illustration of how the two types of similarity are differing across the different police stations is shown in Figure 5.32

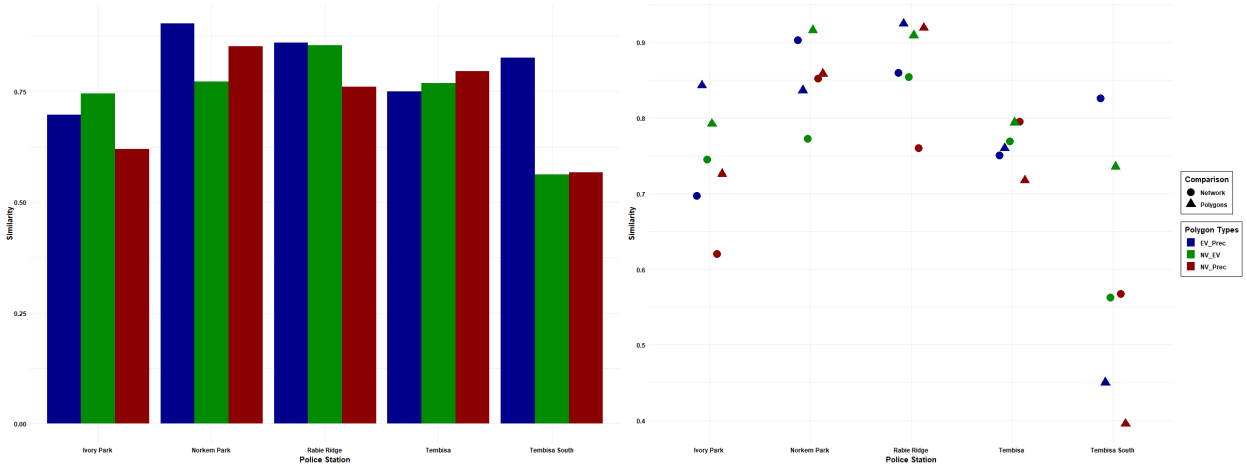


Figure 5.32: A bar (left) and dot (right)plot showing the variation in polygon and road network similarity across five contiguous police stations.

By quantifying accessibility through similarity, an approach is provided that measures how each boundary type aligns with local characteristics in terms of police station accessibility. The outcomes inform decisions on which boundary types best reflect accessibility needs, helping optimise precinct planning and service allocation.

5.3 Validation Data

We consider a different location in Western Cape, South Africa. In particular three police precincts in Khayelitsha, Harare and Lingelethu are selected. Note these precincts are contiguous. Point pattern data on these three precincts, with the exact location of the crime incidences are recorded the location of where the crime was reported. The data set spans 2006 to 2016, for each of the three precincts. Table 2.2 shows the head and tail of the data set for Khayelitsha.

Herein we consider the year 2016. Our analysis spans eight police precincts, comprising the three primary precincts of Khayelitsha and five adjacent precincts. This expanded scope enables us to investigate how the configuration of the road network and the diverse shapes of precinct polygons affect their similarity, furthering the discussion introduced in Section 5.2. Figure 5.33 shows contiguous plots including the three police precincts as lattice data, with the location of police stations as centers. We also include the Euclidean

Voronoi boundaries induced by these police stations, as shown in Figure 5.34.

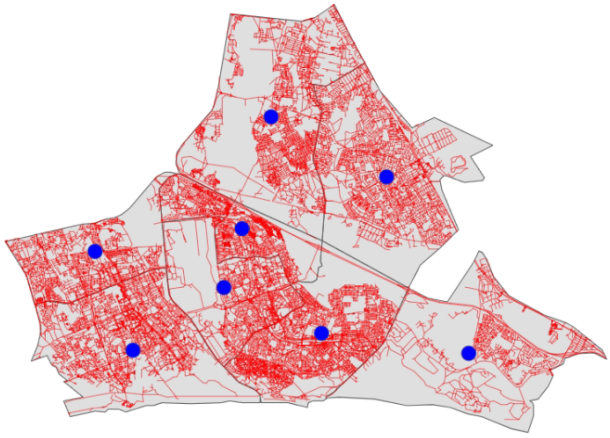


Figure 5.33: A subset of Western Cape, showing 8 police stations, with the corresponding administrative precincts and the road network.

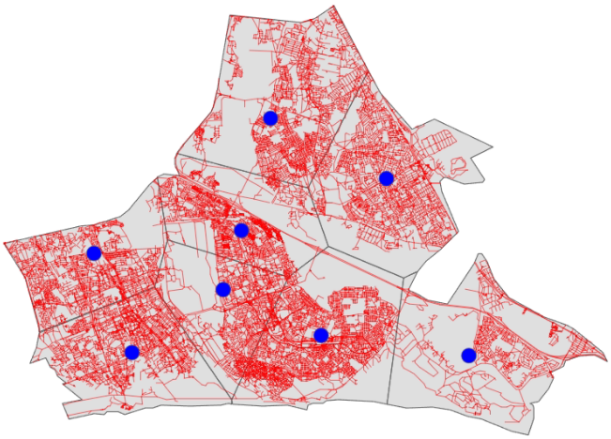


Figure 5.34: A subset of Western Cape, showing 8 police stations, with the corresponding Euclidean Voronoi boundaries and the road network.

Figure 5.35 shows the tessellated road network, with the tessellation induced by the police station locations.

The resulting network Voronoi polygons are shown in Figure 5.36.

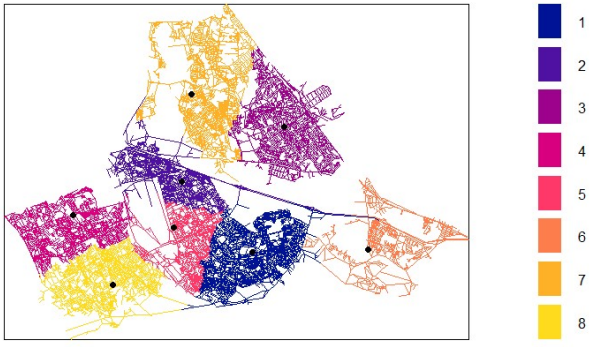


Figure 5.35: Tessellated road network generated by the locations of the police stations, with each tessellation corresponding to a Voronoi polygon.

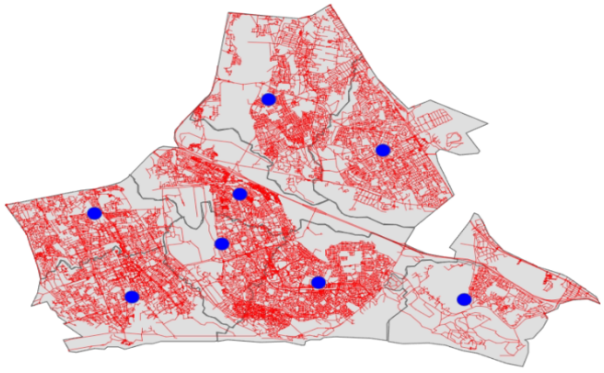


Figure 5.36: A subset of Western Cape, showing 8 police stations, with the corresponding Euclidean voronoi boundaries and the road network.

Similar to Figure 5.11, we present the eight police precincts with various boundary types overlaid, providing a visual comparison of the differences between the boundary types.

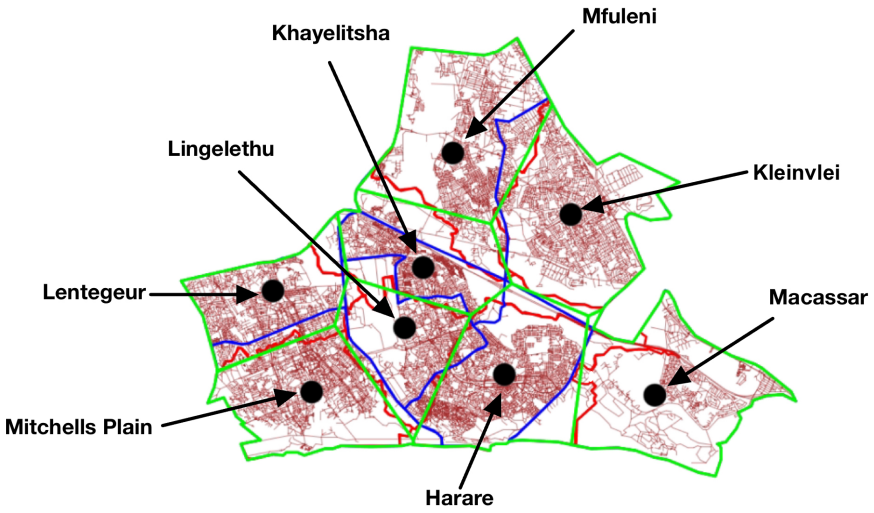


Figure 5.37: Overlaid boundaries of eight contiguous police stations, enabling pairwise comparisons between two types of polygons.

Table 5.4 and Figure 5.38 show the similarity of polygons and network representations for various police stations across three pairwise comparisons. Kleinvleiand Macassar have the highest similarities across the two methods. Khayelitsha, shows a relatively lower polygon similarity, between the network Voronoi and administrative precinct comparison and Euclidean Voronoi boundary comparison, which indicates weaker alignment in polygonal structures. However, network similarity values are higher. Lingeletu-West, shows moderate similarity in both polygonal and network comparisons. Harare and Mfuleni, both have very close similarity values across all metrics, suggesting well-aligned polygonal boundaries and road network structures.

Table 5.4: A summary of polygons and network similarity results of the police stations.

ID	Police Station	Polygons			Network		
		NV Prec	NV vs EV	EV vs Prec	NV vs Prec	EV vs Prec	NV vs EV
1	Harare	0.62341	0.79498	0.72557	0.84097	0.87215	0.87750
2	Khayelitsha	0.37282	0.56552	0.44403	0.90446	0.91283	0.83834
3	Kleinvlei	0.82868	0.92324	0.84807	0.77526	0.95590	0.67687
4	Lentegeur	0.68365	0.78984	0.79310	0.69123	0.68767	0.70838
5	Lingelethu	0.57476	0.79170	0.62141	0.62164	0.70488	0.68639
6	Macassar	0.82996	0.90097	0.88941	0.85257	0.96787	0.85191
7	Mfuleni	0.74220	0.81781	0.69862	0.84049	0.86881	0.79663
8	Mitchells Plain	0.77805	0.91006	0.78564	0.74578	0.91279	0.74994

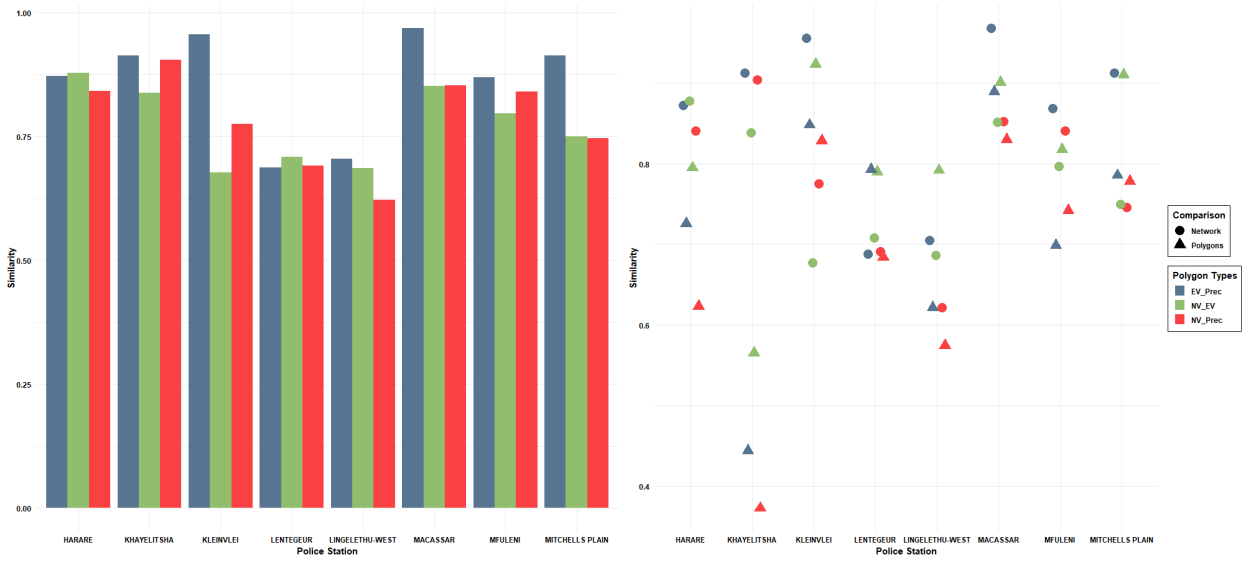


Figure 5.38: An illustration of how the similarities vary across the different police station precincts, bar plot on the left and a point plot on the right. The different types of comparisons are colour coded on both charts, and similarities tests based on the polygons are presented using a triangle and network using circles.

Table 5.5 presents an overview of the average similarities between polygon and network metrics across three pairwise comparisons. Table 5.5 indicates that the average similarity of the road network is consistently higher than that of the polygons across all three comparisons. The similarity in road networks across various polygons is evident, irrespective of the manner in which the polygons delineate the network. This is due to the presence of extensive areas within the precincts that have sparse or nonexistent road networks. The comparison of network Voronoi boundaries and administrative precincts reveals a moderate similarity in polygon shapes, whereas the similarity in network structures is even greater. This indicates that the network structure encompasses more information than the polygons alone. The comparison of network Voronoi and Euclidean Voronoi reveals a high degree of similarity in both polygonal and network analyses. This indicates that both Voronoi polygonal regions (determined along the network and in Euclidean space) are effective representations of the boundaries in terms of accessibility. Comparative analysis of Euclidean Voronoi polygons and administrative precincts reveals a moderate degree of similarity in terms of polygonal overlap and network characteristics.

Table 5.5: Comparison of the averages for the polygonal similarity test and the network similarities.

Comparison	**Average Polygons Similarity**	**Average Network Similarity**
Network Voronoi and administrative precincts	0.67933	0.78405
Network and Euclidean Voronoi	0.82410	0.84756
Euclidean Voronoi and administrative precincts	0.75061	0.77394

Figure 5.39 shows the three stations bounded by the administrative precinct boundaries. The points in the plot are extracted from the data set as the coordinates of where the crimes were reported. These points are superimposed onto the various boundary types shown in Figures 5.39 to 5.41 and are labeled based

on the polygon or precinct in which they fall. Figures 5.39 to 5.41 illustrate differences in the shapes of the polygons. Consequently, certain crime points that appear, for example, in Khayelitsha in Figure 5.39 might now fall under Lingelethu in Figure 5.41 due to changes in the boundary types. This is particularly evident in cases where crime incidents occur in areas that are geographically closer to one police station but fall under the administrative jurisdiction of another. A prime example of this is the top left corner of the Lingelethu precinct in Figure 5.39, where crime points are closer to the Khayelitsha police station but are currently reported to Lingelethu.

Another illustrative example are those few crimes located at the top left of Lingelethu precinct which are obviously far from the Lingelethu station and nearer to the Khayelitsha police station. These are supposed to be reported at the Khayelitsha precinct not Lingelethu as the administrative boundaries suggest.

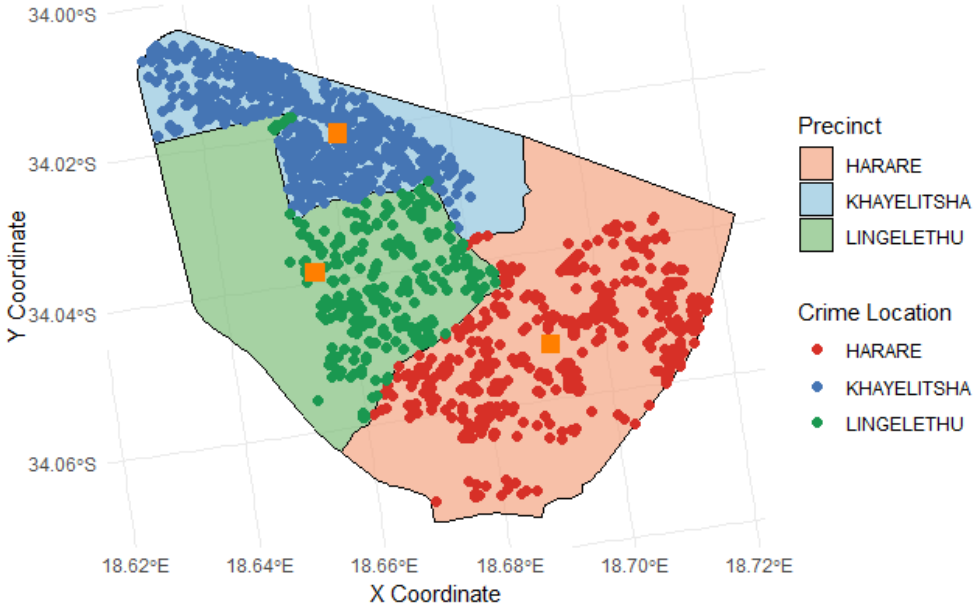


Figure 5.39: The three precincts in the Western Cape, delineated by adminstrative boundary lines, with reported crime points displayed within each precinct. Police station locations are represented by the orange squares.

An interesting observation in Figures 5.40 and 5.41 is the slight difference in scatter of the crime points between the network and Euclidean Voronoi polygon boundaries. This indicates the distribution or number of crimes falling within these two polygon types is nearly identical. Based on this observation, the administrative boundaries are not truly representative of access to police station. Table 2.3 summarises the types and total number of crimes recorded in 2016 across the three precincts.

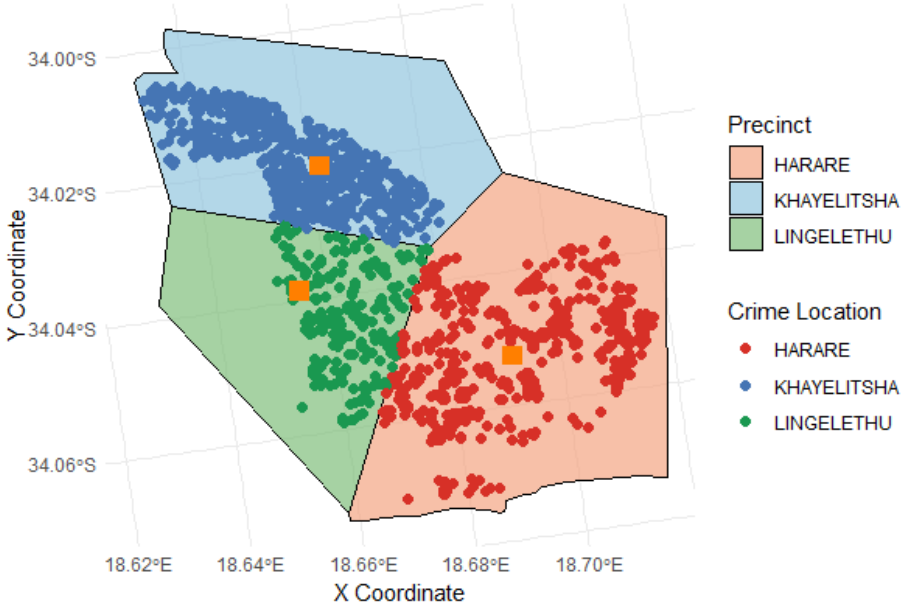


Figure 5.40: The three precincts in the Western Cape, delineated by Euclidean Voronoi boundaries, with reported crime points shown in each precinct.

Figure 5.41 shows the precincts resulting from the network structure, with boundary shapes that are more detailed than those of the first two boundary types.

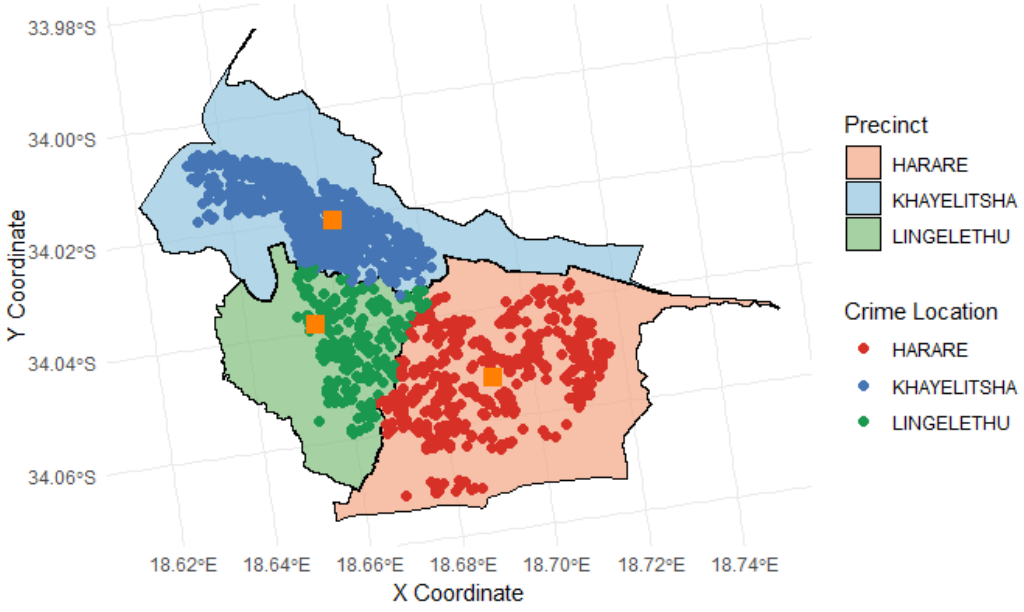


Figure 5.41: The three precincts in the Western Cape, separated by network Voronoi boundaries, with reported crime points shown in each precinct.

Table 5.6 provides a brief summary of how the locations of reported crimes change as the polygon boundary types vary.

Table 5.6: Crime counts for administrative, Euclidean Voronoi, and network Voronoi boundaries.

Place	**Administrative Boundaries**	**Euclidean Voronoi**	**Network Voronoi**
Harare	723	753	751
Khayelitsha	780	817	817
Lingelethu	327	260	255

To examine and evaluate proportional differences in the number of crime points across the three types of boundaries, we estimated the fraction change relative to the administrative police precinct boundaries (which served as the baseline). Equation (5.1) was used to determine proportional changes with respect to the administrative boundaries, that are shown in Table 5.7.

$$\text{Proportional Change} = \frac{\text{Voronoi Boundary Count} - \text{Administrative Boundary Count}}{\text{Administrative Boundary Count}} \times 100 \tag{5.1}$$

Table 5.7: Proportional changes in crime point metrics for Network and Euclidean Voronoi boundaries compared to administrative police precinct boundaries.

Precinct	**Network Voronoi % change**	**Euclidean Voronoi % change**
Harare	+3.87%	+4.15%
Khayelitsha	+4.74%	+4.74%
Lingelethu	-22.02%	-20.49%

According to the results in Tables 5.6 and 5.7, the number of crime incidents in Harare precinct increases when network and Euclidean Voronoi borders are compared to administrative boundaries. According to this, the police station's accessibility increased. That is, the accessibility improved as it is measured by the spatial proximity using Voronoi polygons, as opposed to administrative boundaries. This reflects a reasonable distribution of police station service areas when spatial proximity is factored in. Similar patterns are seen in Khayelitsha, where both Voronoi boundary types consistently increased by 4.74% when compared to administrative boundaries. Proposing improved coverage and accessibility as assessed through Voronoi polygons. In Lingelethu, the Voronoi boundaries lead to a noteworthy decrease in the number of points. This suggests limited accessibility to this police station under administrative precinct boundaries, indicating that Voronoi-based coverage models may more accurately represent the actual service area.

To link these findings to the similarity values obtained in Table 5.4, as illustrated graphically in Figure 5.38, the dot plot shows that Lingelethu precinct has low similarity scores for both polygon and network comparisons. This points to lower accessibility, as evidenced by the decrease in crimes reported to Lingelethu under the Voronoi models. Instead, crime points are being reassigned to more accessible precincts, as shown by the increased number of crime reports in the other two precincts in Table 5.6.

Under the Voronoi models, fewer crimes are reported in Lingelethu, indicating potential under-reporting.

While this analysis was limited to three police stations due to data constraints, with a larger data set, the relationship between accessibility and under-reporting could be further established.

The use of administrative boundaries for socio-cultural or legal divisions are important (for example in Lingelethu). In places like Harare and Khayelitsha, network-based boundaries are more appropriate since spatial proximity significantly improves accessibility.

Chapter 6

Conclusion

The goal of this mini-dissertation was to assess police station accessibility through applications of novel Voronoi diagrams and spatial similarity tests. The study was motivated by the need to look into police station accessibility, especially in areas where spatial factors contribute to under reporting of crimes to police stations within a precinct. The study involved a multi-step process. Euclidean Voronoi diagrams were used to delineate police station service areas based on spatial proximity. These were improved by network-based Voronoi diagrams that incorporated road networks. Police service areas were analysed to identify gaps and overlaps in police station coverage. Finally, the approach was validated using a data set comprising exact crime location points and precinct boundaries.

Key findings from this research include:

1. Voronoi diagrams, when augmented with road networks and police station data, provided an effective means of identifying service areas, gaps in coverage, and areas of overlap in accessibility.
2. Accessibility varied across regions, with under-served areas often linked to inadequate road network.
3. The validation data showed the potential of the proposed spatial similarity tests to detect misalignments between administrative precinct boundaries and spatial proximity-based boundaries.
4. The results highlight the potential of evidence-based adjustments to police station placement and precinct boundaries to improve service delivery.

This mini-dissertation provides a tool that allows policymakers, law enforcement, and urban planners to use spatial data to make decisions. The study offers a methodology for evaluating police station accessibility by combining Voronoi diagrams with information on administrative boundaries and the road network. In the end, these insights can improve public safety results by precinct design, and police service resource allocation.

The study extends the use of Voronoi diagrams to practical accessibility assessments, which theoretically advances spatial analysis. Approaches created here can be applied to several areas of public service, offering a reproducible and scalable tool for resource allocation spatial planning.

Despite these contributions, the study faced some limitations:

1. The study was limited by the high density of the road network for the whole of South Africa. This restricted the analysis to smaller areas due to computational complexity.
2. The analysis was constrained by validation data availability. It relied on a limited validation data set of crime points of Khayelitsha.
3. The accessibility analysis made simplifying assumptions. These assumptions included uniform response capacity across police stations and consistent road conditions. These assumptions may not fully represent real-world dynamics.
4. The analysis does not account for temporal variations in crime trends, traffic conditions, or population movements.

Building on this research, future studies could:

1. Develop a similarity testing measure that address the limitations of communal outer boundaries in subsets. This ensures that inflated similarity values that might arise from shared outer polygons boundaries are appropriately handled.
2. Investigate changes in accessibility over time, and consider factors like traffic patterns, urban growth, and crime trends.
3. Integrate real-time road network and crime data to improve the relevance and applicability of the accessibility models.
4. Expand validation scope to include more precincts and crime point data sets. This will in-turn improve the generalisability of the methodology.
5. Adapt the methods to quantify accessibility for other essential public services, such as fire stations, emergency medical services, and disaster response units.

In conclusion, this mini-dissertation:

1. Introduced a novel methodology for creating network Voronoi diagram polygons.
2. Developed a novel, spatially-informed methodology for quantifying police station accessibility using linear network Voronoi diagrams and administrative boundaries data.

3. Demonstrated the use of the approach in identifying gaps and overlaps in service coverage, using a recently introduced similarity test for linear networks.

4. Validated the methodology on real crime data, to yield insights into the relationship between administrative boundaries and accessibility.

5. Highlighted actionable recommendations for improving police service resource allocation and precinct boundary design.

Bibliography

[1] Tinghua Ai, Wenhao Yu, and Yakun He. Generation of constrained network Voronoi diagram using linear tessellation and expansion method. *Computers, Environment and Urban Systems*, 51:83–96, 2015.

[2] Yusra A. Y. Al-Najjar and Der Chen Soong. Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI. *International Journal of Scientific & Engineering Research*, 3(8), August 2012.

[3] Harith Alani, Christopher B Jones, and Douglas Tudhope. Voronoi-based region approximation for geographical information retrieval with gazetteers. *International Journal of Geographical Information Science*, 15(4):287–306, 2001.

[4] Martin A Andresen. Testing for similarity in area-based spatial patterns: A nonparametric Monte Carlo approach. *Applied Geography*, 29(3):333–345, 2009.

[5] Qi Wei Ang, Adrian Baddeley, and Gopalan Nair. Geometrically corrected second order analysis of events on a linear network, with applications to ecology and criminology. *Scandinavian Journal of Statistics*, 39(4):591–617, 2012.

[6] Franz Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[7] Franz Aurenhammer and Rolf Klein. Voronoi diagrams. *Handbook of Computational Geometry*, 5(10):201–290, 2000.

[8] Adrian Baddeley, Ege Rubak, and Rolf Turner. *Spatial Point Patterns: Methodology and Applications with R*. CRC Press, 2015.

[9] Adrian Baddeley, Ege Rubak, and Rolf Turner. *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC Press, London, 2015.

[10] Adrian Baddeley, Ege Rubak, and Rolf Turner. *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC Press, London, 2018.

[11] Adrian Baddeley and Rolf Turner. Spatstat: An R package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42, 2005.

[12] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[13] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[14] Marc Barthelemy. Spatial networks. *Physics Reports*, 499(1-3):1–101, 2011.

[15] Priyadarshi Bhattacharya and Marina L Gavrilova. Voronoi diagram in optimal path planning. In *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*, pages 38–47. IEEE, 2007.

[16] Roger S Bivand, Edzer J Pebesma, and Virgilio Gomez-Rubio. *Applied Spatial Data Analysis with R*. Springer, 2008.

[17] B Boots, A Okabe, and K Sugihara. Spatial Tessellations. *Geographical Information Systems*, 1:503–526, 1999.

[18] B. Boots, A. Okabe, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, volume 1 of *Geographical Information Systems*, pages 40–60. John Wiley & Sons, New York, NY, 2nd edition, 1999.

[19] B. Boots, A. Okabe, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, volume 1 of *Geographical Information Systems*, pages 49–72. John Wiley & Sons, New York, NY, 2nd edition, 1999.

[20] B. Boots, A. Okabe, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, volume 1 of *Geographical Information Systems*, pages 229–270. John Wiley & Sons, New York, NY, 2nd edition, 1999.

[21] M. I. Borrajo, W. Gonzalez-Manteiga, and M. D. Martinez-Miranda. Testing for significant differences between two spatial patterns using covariates. *Spatial Statistics*, 2019.

[22] David A. Brannan, Matthew F. Esplen, and Jeremy J. Gray. *Geometry*. Cambridge University Press, 1998.

[23] Peter A Burrough, Rachael A McDonnell, and Christopher D Lloyd. *Principles of Geographical Information Systems*. Oxford University Press, USA, 2015.

[24] Long Chen and Jin-chao Xu. Optimal Delaunay triangulations. *Journal of Computational Mathematics*, pages 299–308, 2004.

[25] Emilio Chuvieco. *Fundamentals of Satellite Remote Sensing: An Environmental Approach*. CRC press, 2020.

[26] Mila Coetzee. A new similarity measure for spatial linear networks. Master's thesis, University of Pretoria, South Africa, 2023.

[27] Noel Cressie. Statistics for Spatial Data. *Probability and Mathematical Statistics*, 1993.

[28] Mark RT Dale and Marie-Josée Fortin. *Spatial Analysis: A Guide for Ecologists*. Cambridge University Press, 2014.

[29] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, 2nd edition, 2000.

[30] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry*. Springer-Verlag, 2nd revised edition, 2008. Section 7.2: Computing the Voronoi Diagram.

[31] B. Delaunay. Sur la sphère vide. *Bulletin of the Academy of Sciences of the USSR, VII: Classe des Sciences Mathématiques et Naturelles*, pages 793–800, 1934.

[32] SciPy Developers. scipy.spatial.Voronoi – SciPy v1.7.1 Manual, 2021. Available at: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Voronoi.html`. Accessed: 2023-09-23.

[33] Peter J. Diggle. A kernel method for smoothing point process data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 34(2):138–147, 1985.

[34] Peter J. Diggle. *Statistical Analysis of Spatial Point Patterns*. Hodder Arnold, London, 2nd edition, 2013.

[35] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[36] Matt Duckham, Qian Chayn Sun, and Michael F Worboys. *GIS: A Computing Perspective*. CRC press, 2023.

[37] T. Duong, B. Goud, and Kristine K. Schauer. Closed-form density-based framework for automatic detection of cellular morphology changes. *Proceedings of the National Academy of Sciences*, 109(22):8382–8387, 2012.

[38] Rex A Dwyer. A faster divide-and-conquer algorithm for constructing Delaunay triangulations. *Algorithmica*, 2(1):137–151, 1987.

[39] John Eck, Spencer Chainey, James Cameron, and Ronald Wilson. *Mapping crime: Understanding hotspots*. National Institute of Justice, 2005.

[40] Chenglin Fan, Jun Luo, Jinfei Liu, and Yinfeng Xu. Half-plane Voronoi diagram. In *2011 Eighth International Symposium on Voronoi Diagrams in Science and Engineering*, pages 127–133. IEEE, 2011.

[41] Hongchao Fan, Bisheng Yang, Alexander Zipf, and Adam Rousell. A polygon-based approach for matching OpenStreetMap road networks with regional transit authority data. *International Journal of Geographical Information Science*, 30(4):748–764, 2016.

[42] Sam Fletcher and Md Zahidul Islam. Comparing sets of patterns with the Jaccard index. *Australasian Journal of Information Systems*, 22, 2018.

[43] M. J. Fortin and M. R. T. Dale. *Spatial Analysis: A Guide for Ecologists.* Cambridge University Press, 2005.

[44] Steven Fortune. A sweepline algorithm for Voronoi diagrams. In *Proceedings of the Second Annual Symposium on Computational Geometry*, pages 313–322, 1986.

[45] Steven Fortune. Voronoi diagrams and delaunay triangulations. In *Handbook of Discrete and Computational Geometry, 2nd Ed.*, 2004.

[46] I. Fuentes-Santos, W. Gonzalez-Manteiga, and J. Mateu. A nonparametric test for the comparison of first-order structures of spatial point processes. *Spatial Statistics*, 22:240–260, 2017.

[47] Thierry Garaix, Maxence Delorme, Olivier Peton, and Roberto W Calvo. A robust capacitated facility location problem with uncertain demand within neighborhoods. *European Journal of Operational Research*, 206(2):489–497, 2010.

[48] Andrea Gilardi and Robin Lovelace. *osmextract: Download and Import Open Street Map Data Extracts*, 2024. R package version 0.5.1.

[49] Christopher M Gold, Peter R Remmele, and Thomas Roos. Voronoi methods in GIS. *Advanced School on the Algorithmic Foundations of Geographic Information Systems*, pages 21–35, 1996.

[50] Michael F Goodchild. Geographical data modeling. *Computers & Geosciences*, 18(4):401–408, 1992.

[51] Peter J Green and Robin Sibson. Computing Dirichlet tessellations in the plane. *The Computer Journal*, 21(2):168–173, 1978.

[52] Leonidas Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi. *ACM Transactions on Graphics (TOG)*, 4(2):74–123, 1985.

[53] Timothy C. Hart and Callie Marie Rennison. Reporting crime to the police, 1992-2000. *Bureau of Justice Statistics, U.S. Department of Justice*, 2003.

[54] Jared Hewko, Karen E Smoyer-Tomic, and M John Hodgson. Measuring neighbourhood spatial accessibility to urban amenities: does aggregation error matter? *Environment and Planning A*, 34(7):1185–1206, 2002.

[55] Robert J. Hijmans. *terra: Spatial Data Analysis*, 2023. R package version 1.7-29.

[56] Shinichi Honiden, Michael E Houle, Christian Sommer, and Martin Wolff. Approximate shortest path queries using Voronoi duals. *Transactions on Computational Science IX: Special Issue on Voronoi diagrams in Science and Engineering*, pages 28–53, 2010.

[57] SR Jones. Accessibility measures: A literature review. *Transport and Road Research Laboratory (TRRL)*, 1(TRRL LR 967 Monograph), 1981.

[58] Deepti Joshi, Leen-Kiat Soh, Ashok Samal, and Jing Zhang. A dissimilarity function for geospatial polygons. *Knowledge and Information Systems*, 41(1):153–188, 2014. Received: 12 December 2011 / Revised: 27 May 2013 / Accepted: 28 May 2013 / Published online: 5 July 2013.

[59] David H. Kaplan. Spatial transformations in geographic analysis. *Geographical Analysis*, 53(2):123–137, 2021.

[60] Michael Karasick, Derek Lieber, and Lee R Nackman. Efficient Delaunay triangulation using rational arithmetic. *ACM Transactions on Graphics (TOG)*, 10(1):71–91, 1991.

[61] Rene Kirsten and Inger Nicolette Fabris-Rotelli. A generic test for the similarity of spatial data. *South African Statistical Journal*, 55(1):55–71, 2021.

[62] Jari Korhonen and Junyong You. Peak signal-to-noise ratio revisited: Is simple beautiful? In *2012 Fourth International Workshop on Quality of Multimedia Experience*, pages 37–38. IEEE, 2012.

[63] Charles L Lawson. Generation of a triangular grid with applications to contour plotting. *Jet Propulsion Laboratory Technical Memorandum*, 299(2), 1972.

[64] Charles L Lawson. Transforming triangulations. *Discrete Mathematics*, 3(4):365–372, 1972.

[65] D. T. Lee. On finding k-Nearest neighbors in the plane. Technical Report Eng. 76-2216, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, 1976.

[66] D. T. Lee. *Proximity and Reachability in the Plane*. Ph.d. thesis, University of Illinois, Urbana, Illinois, 1978. Coordinated Science Laboratory Report ACT-12.

[67] D. T. Lee and B. J. Schacter. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer and Information Sciences*, 9(3):219–242, 1980.

[68] Der-Tsai Lee. Two-dimensional Voronoi diagrams in the Lp metric. *Journal of the ACM (JACM)*, 27(4):604–618, 1980.

[69] Mokwon Lee, Kokichi Sugihara, and Deok-Soo Kim. Robust construction of Voronoi Diagrams of spherical balls in three-dimensional space. *Computer-Aided Design*, 152:103374, 2022.

[70] Lixuan Liu and Zijian Liu. Delineation of traditional village boundaries: The case of Haishangqiao village in the Yiluo River Basin, China. *Plos One*, 17(12):e0279042, 2022.

[71] Errol Lynn Lloyd. On triangulations of a set of points in the plane. In *18th Annual Symposium on Foundations of Computer Science (FOCS 1977)*, pages 228–240. IEEE, 1977.

[72] Paul A. Longley, Michael F. Goodchild, David J. Maguire, and David W. Rhind. *Geographic Information Science and Systems*. John Wiley & Sons, Hoboken, NJ, 4th edition, 2015.

[73] Ma Cruz López-De-Los-Mozos and Juan A Mesa. The sum of absolute differences on a network: Algorithm and comparison with other equality measures. *INFOR: Information Systems and Operational Research*, 41(2):195–210, 2003.

[74] Xiaomin Lu and Haowen Yan. An algorithm to generate a weighted network Voronoi diagram based on improved PCNN. *Applied Sciences*, 12(12), 2022.

[75] Dennis Luxen and Christian Vetter. Fastest path computation with on-the-fly edge-flag shortcuts. *Proceedings of the 9th International Conference on Experimental Algorithms (SEA'11)*, pages 260–270, 2011.

[76] Ithandile Mbewu, Emeka E Obioha, and Ishmael Mugari. Encouraging and discouraging factors to crime reporting in South Africa: A survey of residents' crime reporting behaviour in Mthatha South Africa police service precinct. *Cogent Social Sciences*, 7(1):2002542, 2021.

[77] Lionel Moisan and Adrien Stival. Optimal methods for image comparison in spatial analysis. *IEEE Transactions on Image Processing*, 28(5):2435–2448, 2019.

[78] JM Nelson. A triangulation algorithm for arbitrary planar domains. *Applied Mathematical Modelling*, 2(3):151–159, 1978.

[79] Anna Nowak. Application of Voronoi diagrams in contemporary architecture and town planning. *Challenges of Modern Technology*, 6(2):30–34, 2015.

[80] Takao Ohya, Masao Iri, and Kazuo Murota. A fast Voronoi-diagram algorithm with quaternary tree bucketing. *Information Processing Letters*, 18(4):227–231, 1984.

[81] Atsuyuki Okabe, Kenji Okunuki, and Shinya Shiode. SANET: A toolbox for spatial analysis on a network. *Geographical Analysis*, 38(1):57–66, 2006.

[82] Atsuyuki Okabe, Kenji Okunuki, and Shinya Shiode. The SANET toolbox: New methods for network spatial analysis. *Transactions in GIS*, 10(4):535–550, 2006.

[83] Atsuyuki Okabe and Takashi Satoh. Spatial analysis on a network. In A. Stewart Fotheringham and Peter A. Rogerson, editors, *SAGE Handbook of Spatial Analysis*, pages 443–464. SAGE Publications, London, 2009.

[84] Atsuyuki Okabe, Toshiaki Satoh, Takehiro Furuta, Atsuo Suzuki, and Kyoko Okano. Generalized network Voronoi diagrams: Concepts, computational methods, and applications. *International Journal of Geographical Information Science*, 22(9):965–994, 2008.

[85] Atsuyuki Okabe and Kokichi Sugihara. *Spatial Analysis Along Networks: Statistical and Computational Methods*, volume 1, pages 63–68. John Wiley & Sons, New York, NY, 1st edition, 2012.

[86] Atsuyuki Okabe and Kokichi Sugihara. *Spatial Analysis Along Networks: Statistical and Computational Methods*, volume 1, pages 81–83. John Wiley & Sons, New York, NY, 1st edition, 2012.

[87] Atsuyuki Okabe and Atsuo Suzuki. Locational optimization problems solved through Voronoi diagrams. *European journal of operational research*, 98(3):445–456, 1997.

[88] OpenAI. DALL-E 3, 2024. Text-to-Image Generation Model, `https://openai.com/research/dall-e`.

[89] Joseph O'Rourke, Chi-Bin Chien, Thomas Olson, and David Naddor. A new linear algorithm for intersecting convex polygons. *Computer Graphics and Image Processing*, 19(4):384–391, 1982.

[90] Dimitrios Papadias, Nikos Mamoulis, and B Delis. Algorithms for querying by spatial structure. In *Proceedings of Very Large Data Bases Conference (VLDB), New York*, 1998.

[91] Jin-Seo Park and Se-Jong Oh. A new concave hull algorithm and concaveness measure for n-dimensional datasets. *Journal of Information Science and Engineering*, 28(3):587–600, 2012.

[92] Edzer Pebesma. Simple features for R: Standardized support for spatial vector data. *The R Journal*, 10(1):439–446, 2018.

[93] Edzer Pebesma and Roger Bivand. *Spatial Data Science: With Applications in R*. Chapman and Hall/CRC, 2023.

[94] Thomas Lin Pedersen. *ggforce: Accelerating 'ggplot2'*, 2023. R package version 0.4.1.

[95] W. J. Pervin. *Foundation of General Topology*. Academic Press, New York, 1964.

[96] GR Pontius Jr. Metrics that make a difference. *Switzerland. ISBN-10: 3030707644*, 2022.

[97] R Gil Pontius Jr and Laura C Schneider. Land-cover change model validation by an ROC method for the Ipswich watershed, Massachusetts, USA. *Agriculture, Ecosystems & Environment*, 85(1-3):239–248, 2001.

[98] Robert Gilmore Pontius Jr and Marco Millones. Death to Kappa: Birth of quantity disagreement and allocation disagreement for accuracy assessment. *International Journal of Remote Sensing*, 32(15):4407–4429, 2011.

[99] Franco P Preparata and Michael I Shamos. *Computational Geometry: An Introduction*. Springer Science & Business Media, 2012.

[100] QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation, 2009.

[101] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023.

[102] Raimundo Real and Juan M Vargas. The probabilistic basis of Jaccard's index of similarity. *Systematic Biology*, 45(3):380–385, 1996.

[103] Dierk Rhynsburger. Analytic delineation of Thiessen polygons. *Geographical Analysis*, 5(2):133–144, 1973.

[104] S. Riviere and D. Schmitt. Two-dimensional line space Voronoi diagram. In *International Symposium on Voronoi Diagrams in Science and Engineering*, pages 168–175, 2007.

[105] Marjorie R. Sable, Fran Danis, Denise L. Mauzy, and Sarah K. Gallagher. Barriers to reporting sexual assault for women and men: Perspectives of college students. *Journal of American College Health*, 55(3):157–162, 2006.

[106] Stephan R Sain and Noel Cressie. A spatial model for multivariate lattice data. *Journal of Econometrics*, 140(1):226–259, 2007.

[107] Mehul P Sampat, Zhou Wang, Shalini Gupta, Alan Conrad Bovik, and Mia K Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing*, 18(11):2385–2401, 2009.

[108] Michael Ian Shamos and Dan Hoey. Closest-point problems. In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, pages 151–162. IEEE, 1975.

[109] Wang Shu-Xi. The improved Dijkstra's shortest path algorithm and its application. *Procedia Engineering*, 29:1186–1190, 2012.

[110] de Melo Silas Nogueira, Frank Richard, and Brantingham Patricia. Voronoi diagrams and spatial analysis of crime. *The Professional Geographer*, 69(4):579–590, 2017.

[111] Steven S Skiena. Probing convex polygons with half-planes. *Journal of Algorithms*, 12(3):359–374, 1991.

[112] Laurynas Speicys and Christian S. Jensen. Road network data model. In Shashi Shekhar and Hui Xiong, editors, *Encyclopedia of GIS*, pages 964–971. Springer, Boston, MA, 2008.

[113] Richard Stassen and Vania Ceccato. Police accessibility in Sweden: An analysis of the spatial arrangement of police services. *Policing: A Journal of Policy and Practice*, 15(2):896–911, 12 2019.

[114] Alfred Stein, Freek D van der Meer, and Ben Gorte. *Spatial Statistics for Remote Sensing*, volume 1. Springer Science & Business Media, 1999.

[115] Kokichi Sugihara. Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams. *CVGIP: Graphical Models and Image Processing*, 55(6):522–531, 1993.

[116] Yi-li Tan, Ye Zhao, and Yourong Wang. Power Network Voronoi Diagram and Dynamic Construction. *Journal of Networks*, 7(4):675–682, 2012.

[117] T. Tango and K. Takahashi. A flexible spatial scan statistic with a restricted likelihood ratio for detecting disease clusters. *Statistics in Medicine*, 29(2):207–218, 2010.

[118] Pornrawee Tatit, Kiki Adhinugraha, and David Taniar. Navigating the Maps: Euclidean vs. Road Network Distances in Spatial Queries. *Algorithms*, 17(1):29, 2024.

[119] Rolf Turner. *deldir: Delaunay triangulation and Dirichlet (Voronoi) tessellation*, 2023. R package version 1.3-10.

[120] Jeroen van Hoof, Volker Jaenisch, and Yiming Huang. Foronoi: A Python library for Voronoi Diagrams using Fortune's Algorithm, 2019. Available at: `https://github.com/Yatoom/foronoi` and `https://bit.ly/Foronoi`, Accessed: 2023-09-23.

[121] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stefan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, Charles J. Carey, Ä˚lhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antonio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[122] Y. Wang. A comprehensive study on rasterization techniques. *Journal of Computer Graphics*, 19(4):512–526, 2007.

[123] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[124] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multi-scale structural similarity for image quality assessment. In *IEEE Asilomar Conference on Signals, Systems, and Computers*, pages 9–13, 2003.

[125] Karen G. Weiss. Too ashamed to report: Deconstructing the shame of sexual victimization. *Feminist Criminology*, 5(3):286–310, 2010.

[126] Beccy Wilebore and David Coomes. Combining spatial data with survey data improves predictions of boundaries between settlements. *Applied Geography*, 77:1–7, 2016.

[127] Kenny Wong and Hausi A Müller. *An efficient implementation of Fortune's plane-sweep algorithm for Voronoi diagrams*. University of Victoria, Department of Computer Science, 1991.

[128] Chang Xia, Anqi Zhang, Haijun Wang, and Bin Zhang. Modeling urban growth in a metropolitan area based on bidirectional flows, an improved gravitational field model, and partitioned cellular automata. *International Journal of Geographical Information Science*, 33(5):877–899, 2019.

[129] Kefeng Xuan, Geng Zhao, David Taniar, Bala Srinivasan, Maytham Safar, and Marina Gavrilova. Network voronoi diagram based range search. In *2009 International Conference on Advanced Information Networking and Applications*, pages 741–748. IEEE, 2009.

[130] Kai Zeng and Zhou Wang. 3D-SSIM for video quality assessment. In *2012 19th IEEE International Conference on Image Processing*, pages 621–624. IEEE, 2012.

[131] Feng Zhao, Qingming Huang, and Wen Gao. Image matching by normalized cross-correlation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 2, pages II–II. IEEE, 2006.

[132] Chen Zhong, Stefan Müller Arisona, Xianfeng Huang, Michael Batty, and Gerhard Schmitt. Detecting the dynamics of urban structure through spatial network analysis. *International Journal of Geographical Information Science*, 28(11):2178–2199, 2014.

[133] Jun Zhu, Hsin-Cheng Huang, and Perla E Reyes. On selection of spatial linear models for lattice data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(3):389–402, 2010.