Ternary spatial relations for error detection in map databases

Ivan Majic

https://orcid.org/0000-0002-0834-3791

A thesis submitted in total fulfillment for the degree of Doctor of Philosophy

in the
Department of Infrastructure Engineering
Melbourne School of Engineering
The University of Melbourne

May 2021

Supervisors:
Dr Martin Tomko
Prof Stephan Winter
Dr Elham Naghizadeh

Čovječe pazi da ne ideš malen ispod zvijezda!

Pusti da cijelog tebe prođe blaga svjetlost zvijezda!

Pusti Da ni za čim ne žališ kad se budeš zadnjim pogledima rastajo od zvijezda!

Pusti Na svom koncu mjesto u prah prijeđi sav u zvijezde!

Antun Branko Šimić, Opomena

Abstract

The quality of data in spatial databases greatly affects the performance of location-based applications that rely on maps such as emergency dispatch, land and property ownership registration, and delivery services. The negative effects of such dirty map data may range from minor inconveniences to life-threatening events. Data cleaning usually consists of two steps - error detection and error rectification. Data cleaning is a demanding and lengthy process that requires manual interventions of data experts, in particular where for complex situations involving the consistency of relationships between multiple objects. This thesis presents computational methods developed to automate the detection of errors in map databases and ease the demand for human resources in error detection. These methods are intrinsic, ie., depend only on data being analysed, without the need for a reference dataset. Two models for ternary spatial relations were developed to enable the analyses not possible with existing binary spatial relations. First, the Refined Topological relations model for Line objects (RTL) examines whether the core line object is connected to its surrounding objects on both or only one of its ends. This distinction is particularly important in networks where connectedness determines the function of the object. Second, the Ray Intersection Model (RIM) casts rays between two peripheral objects and uses the intersection sets between these rays and the core object to model its relation to peripheral objects. This provides a basis for reasoning about the core object being between peripheral objects. Both models have been computationally implemented and demonstrated on error detection tasks in OpenStreetMap. The case studies on data for the State of Victoria, Australia demonstrate that the methods developed in this research are effectively detecting errors that could so far not be automatically identified. This research contributes to automated spatial data cleaning and quality assurance, including reducing experts' workload by effectively identifying potential errors.

Declaration of Authorship

I, Ivan Majic, declare that this thesis titled *"Ternary spatial relations for error detection in map databases"* and the work presented in it are my own. I confirm that:

- The thesis comprises only of my original work towards the degree of Doctor of Philosophy except where indicated in the preface;
- due acknowledgement has been made in the text to all other material used; and
- the thesis is fewer than $100,000$ words in length, exclusive of tables, maps, bibliographies and appendices as approved by the Research Higher Degrees Committee.

Signed: ____________________________________

Date: 24.12.2020.

Acknowledgements

When I first landed in Melbourne in January 2017, I knew that the next several years of my life are going to be special! I was excited to have the opportunity to research the topic of my choosing while living in one of the most liveable cities in the world (Melbourne) and in the country that is the second furthest from my home in Croatia (first is New Zealand). Now I can confirm that the past four years have been special in every way. They were filled with challenging problems, stimulating conversations, and smart and interesting people. I can say with confidence, and I am my own worst critic, that I am proud of everything I have achieved in this time as a researcher and as a person. However, none of this would be possible without some special people who have helped me along the way and to whom I will forever be grateful.

First of all, I wish to thank my principal supervisor Dr Martin Tomko. He has supported me from the very beginning and has many times gone above and beyond what is expected from a supervisor. Martin, I have tried to learn from you as much as I could during this time and you will always be my role model. Your passion and love for science are inspiring and I will always admire the creativity and genius with which you approach research problems!

I also wish to thank my co-supervisors Prof Stephan Winter and Dr Elham Naghizadeh without whom I would never to be able to solve all of my research challenges. Stephan, I am amazed by your ability to give attention to the smallest details and maintain the vision of the big picture at the same time. It is something that can only be achieved with a lifetime of dedication and knowledge, and I can only hope to one day attain that ability. Elham, I cannot express how essential you were to my PhD journey. You have helped me understand the data mining methods, you gave me excellent feedback and help with my writing, but most importantly, you have been an example of a young and passionate researcher in whose footsteps I can follow.

I have to mention my PhD colleagues with whom I have developed friendships that I hope will last a lifetime. In the beginning, there were my PhD elders who have come before me and were generous enough to show me the ways of a PhD student, both by sharing their writings with me and by the spoken words of advice: Salil, Zahra, Yaoli, Hao, Surabhi, Debaditya, Ha, Santa, and Hanxian. Then, the other PhD students came after me and I have strived to be an equally good colleague to them: David, Ehsan, Rajesh, Debjit, Subhra, Masoud, Kimia, Hang, Kamal, Marko, Youjin, and Wenyan. All of you have helped me on my PhD journey in one way or another and for that, I am forever thankful.

I would not be here without my parents Zdenka and Bozo who have sacrificed so much to enable me to follow my dreams. They have taught me that there does not exist a problem that cannot be solved, and I have remembered this many times during my PhD. Mama i tata volim vas i nadam se da cu jednog dana biti makar upola dobar roditelj svojoj djeci.

Also, I am the youngest of three brothers and my older brothers Petar and Marko have made me who I am today. They have been my role models growing up and they are my friends growing old. My brothers, I love you more than words can say, but I will still want to beat you in every game we play for there is no sweeter victory.

Lastly, to my partner, my soulmate, and my love. It seems like only yesterday we were a pair of foolish undergrads who dreamt of doing a PhD somewhere far away. Now that we have done it, I am starting to believe that we can do anything we set our minds to. I cannot even imagine doing any of this without you and it is still hard for me to believe that we have achieved all of this. Jelena, I am immensely proud of you and there is no doubt in my mind that you will become the best professor the world has ever seen!

List of Publications

Refereed journal papers

1. Ivan Majic, Elham Naghizade, Stephan Winter, and Martin Tomko. Discovery of topological constraints on spatial object classes using a refined topological model. *Journal of Spatial Information Science*, 18(18):1–30, jun 2019. ISSN 1948660X. doi: 10.5311/JOSIS.2019.18.459. URL http://josis.org/index.php/josis/article/view/459

2. Ivan Majic and Elek Pafka. AwaP-IC—An Open-Source GIS Tool for Measuring Walkable Access. *Urban Science*, 3(2):48, apr 2019. ISSN 2413-8851. doi: 10.3390/urbansci3020048. URL https://www.mdpi.com/2413-8851/3/2/48

3. Ivan Majic, Elham Naghizade, Stephan Winter, and Martin Tomko. RIM: a ray intersection model for the analysis of the between relationship of spatial objects in a 2D plane. *International Journal of Geographical Information Science*, pages 1–26, 2020. ISSN 1365-8816. doi: 10.1080/13658816.2020.1778002. URL https://www.tandfonline.com/doi/full/10.1080/13658816.2020.1778002

4. Ivan Majic, Elham Naghizade, Stephan Winter, and Martin Tomko. There is no way! Ternary qualitative spatial reasoning for error detection in map data. *Transactions in GIS (accepted for publication)*, pages 1–20, 2021. doi: 10.1111/tgis.12765

Refereed conference papers

1. Ivan Majic, Stephan Winter, and Martin Tomko. Finding Equivalent Keys in Openstreetmap: Semantic Similarity Computation Based on Extensional Definitions. In *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, GeoAI '17, pages 24–32, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5498-1. doi: 10.1145/3149808.3149813. URL http://doi.acm.org/10.1145/3149808.3149813

2. Jelena Gabela, Ivan Majic, Allison Kealy, Mark Hedley, and Shenghong Li. Robust vehicle localization and integrity monitoring based on spatial feature constrained PF. In *Proceedings of IEEE/ION PLANS 2020*, St. Louis, Missouri, sep 2020. URL https://www.ion.org/plans/abstracts.cfm?paperID=8276

Other

1. Ivan Majic. Learning rules and detecting errors in OpenStreetMap data. Conference presentation at FOSS4G SOTM 2019, Wellington, NZ, 2019. URL https://youtu.be/SIH2_ZfQrX8

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

Nicaraguan troops have accidentally invaded Costa Rica [14], motorists got lost in a forest without food or water [15], and an Uber driver drove an elderly couple from the airport straight to the cemetery instead of their daughter's house [16]! The one thing that all three of these events have in common is that the people involved were just following the instructions on a map. In all three cases, the maps have misinformed the users because they contained erroneous data. This goes to show how we trust maps, probably because we often believe they are always correct. But, as these examples illustrate, maps can contain incorrect information and this can cause various problems that range from simple inconveniences (e.g., being driven to a wrong location by Uber) to life-threatening events (e.g., being stranded without food or water).

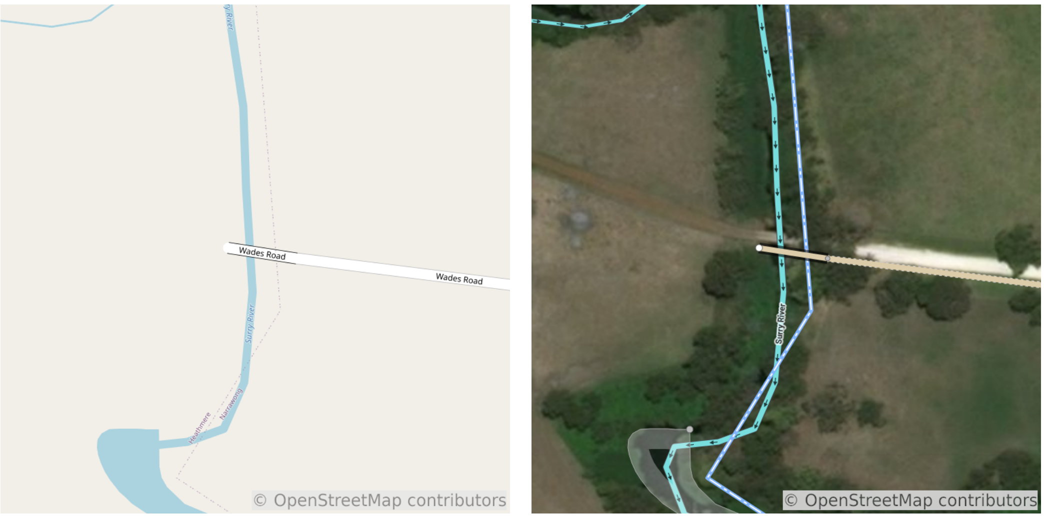


FIGURE 1.1: An example of a map, showing a dead-end bridge that is disconnected on one side (left) and the satellite imagery of the same area that shows that there exists a road or a path that is connected to the bridge, but is not mapped (right).

Most maps we use today come in digital form, and the spatial data that these maps are based on are stored in map databases either locally on our computers and mobile devices or accessed online through the internet. Map databases provide information essential for orientation in space and navigation. They are also used for other important functions and services such as emergency dispatch, land and property ownership registration, and delivery services.

The quality of map data pertains to how well these data reflect the physical world, and it is often assessed across five standard elements of geographical information quality - *completeness, logical consistency, positional accuracy, temporal accuracy, and thematic accuracy* [11]. The data of low quality that contain errors and inconsistencies are often called *dirty data* [17]. The quality of data in map databases directly affects the performance of location-based applications that rely on them. Thus, dirty data can have negative effects on these applications and cause them to provide incorrect results or no results at all.

For example, the map in Figure 1.1 shows a road that is leading to a bridge, but does not continue on the other side of the bridge. In other words, the bridge is a dead-end. However, the satellite imagery in Figure 1.1 clearly shows that there is a path that leads on from the bridge, but this path is missing from the map. This is problematic because there are several dwellings further down that path that have no alternative access. If there was an emergency and this map was used to plan the ambulance dispatch route, the correct path would not be found.

FIGURE 1.2: An example of an incorrect proposed route to the destination building (left) and the satellite imagery of the same area that show there is a driveway that leads directly to the destination building, but is not mapped (right).

Figure 1.2 shows a different example where the map contains the route to the marked destination building that is proposed by the routing algorithm. The problem with the proposed route is that the user will not be able to access the destination building from the destination of the route because other private properties between them are in the way. Again, the satellite imagery of the area (Figure 1.2) shows that there is a driveway access from the main street to the destination building, but it is missing from the map. If there was an emergency and the ambulance used this route, it would have to back-track from the route destination and find the right access road on its own. This would result in delays and increased danger for the patient.

The problems presented in these two examples could be solved by amending the dirty data in map databases that these maps are based on. This process is called data cleaning and usually consists of two components - detection of errors in data and rectification of errors in data [8]. Data cleaning can be a demanding and lengthy process that conventionally requires manual interventions of data experts, because errors and their corrections are often not trivial. In the case of many map databases that are widely used, such as OpenStreetMap (Figures 1.1 and 1.2), Bing Maps, or Google Maps, human resources available for their maintenance are limited and expensive. Thus, if the errors were detected in an automated process, this would enable human experts to focus their attention on error rectification. This would also enable more data to be cleaned with the same resources and would provide a systematic error detection mechanism that is immune to potential human errors such as error oversight.

For the example in Figure 1.1, we could check if there are bridges that are not connected on both sides. For the example in Figure 1.2, we could automatically check if there are obstacles that block access from the route end point and the destination building. However, to be able to perform these checks, we need to be able to capture and describe these relationships between spatial objects (e.g., bridges and roads, and roads and buildings). This is not a trivial task, because these relations usually involve three spatial objects and most current spatial relationship models support only binary relationships of two objects. Because of this, there is a need to develop new spatial relationship models for effective capturing and expression of ternary spatial relationships that involve three objects. Thus, the problem addressed in this thesis is the automated detection of errors in map data with ternary spatial relationships.

1.2 Thesis scope

For this thesis, map databases may be defined as spatial databases containing vector features for general-purpose mapping. Furthermore, general-purpose mapping aims to model the physical world as completely as possible to assist users in tasks such as orientation and navigation (pedestrian or road). Such databases usually contain information about transportation infrastructure, street networks, buildings, and public green spaces. Domain specific data such as meteorological data or raw measurements such as point clouds are not considered here. The focus of this thesis is also set on 2D map data since these are utilised as the basis of most applications such as navigation, routing, and land ownership register.

However, all errors are not alike and they can be differentiated in terms of their *type*, *scope*, *cause*, and *detection strategy*. At the topmost level, map data can be divided into a spatial component that consists of objects' geometries (i.e., points, lines and polygons defined with coordinates), and a semantic component that consists of objects' attributes (i.e., type, name, address, and other attributes are usually defined with text). Spatial relations such as topological (e.g., intersections), metric (e.g., distance), or order (e.g., in front of) relations are used to describe the relationships that objects' geometries are forming in space. This research considers *error type* to define whether the error occurs in the spatial (i.e., spatial errors), the semantic (i.e., semantic errors), or both spatial and semantic components of map data (i.e., spatio-semantic errors). *Error scope* describes how many objects in map data are affected by the error (e.g., errors affecting single, two, or multiple objects). *Error cause* refers to the reason why the error has occurred. In this research, we distinguish three error causes - missing data (i.e., an entire object or parts of its geometry and/or attributes are not present in map database), incorrect data (i.e., geometry and/or attribute data are present in the database but are incorrect), and redundant data (i.e., same geometry or attribute is entered into the database more than once). *Error detection strategy* defines how the error is being detected. The first differentiation of error detection strategies is whether they are manual or automatic and automatic error detection can be achieved by many different approaches such as outlier detection, rule or pattern violation, topological analysis, spatial analysis, and semantic analysis.

Given the above classification, this thesis presents automated error detection strategies. The approaches in this thesis are based on the analysis of ternary spatial relationships between objects, so the error scope involves multiple objects. Regarding their type, these errors are spatio-semantic because they affect both geometric and semantic components of map data. Finally, the cause of the error is often not determined by the automated error detection method and needs to be assessed by a human expert. For instance, the examples in Figures 1.1 and 1.2 both show cases of missing data. These errors can be discovered by analysing the topological relations between objects, but the error cause can only be determined by comparing the map data to the satellite imagery of the area.

1.3 Research challenges

There are several research problems in developing automated error detection approaches that need to be addressed. Some are common to all types of data, and some are specific to map data.

Reference data or ground truth

Current prominent map databases such as Google Maps, OSM, and Bing Maps cover the entire globe in fine detail. It is not uncommon even for trees, trash cans or park benches to be individually mapped and mapping is often performed remotely by volunteers using satellite imagery as a source of information. Due to such large amounts of geographical information, it is becoming increasingly harder to maintain map databases and assure their quality. Traditionally, a high-quality reference dataset produced by expert land surveyors would be acquired and used as ground truth for assessing the accuracy of data. However, it is not feasible to expect such a process to be repeated for all the detailed data present in current maps. Consequently, ground truth data are often not available, and checking each object in a database manually would be too demanding in time and human resources. This leaves an open problem of quality assurance and in particular detection of errors in databases without the ground truth.

Thus, there is an emerging trend of developing intrinsic data quality metrics that rely solely on the data being analysed and do not need ground truth. This is common in machine learning approaches as well, where data are often partitioned into training and test datasets.

This research is based on the assumption that any practically usable dataset must, at least in the area where it is usable, have the majority of the data of satisfactory quality (i.e., fit for use). OpenStreetMap, for example, has a strong community of contributors and a long lineage of edits where contributors are encouraged to correct errors and update the database. On the other hand, many proprietary datasets have a strict quality control process which assures high quality of data. In this thesis, datasets for which the majority of data are assumed to be of acceptable quality are called *mature spatial datasets*. If the majority of the data is considered a norm, then the deviations from that norm can be seen as anomalies or outliers [18]. Thus, outlier detection is applied to detect probable errors in data [19]. This assumption will be referred to as the *assumption of mature spatial dataset* in the remainder of this thesis.

Certainty of errors

Errors are usually defined as data that are different from the reference dataset. However, as discussed in the previous paragraph, the reference dataset is often not available or even not defined. Other times it is not the reference dataset that is used to evaluate the correctness of data, but rules that the experts have agreed upon (e.g., buildings should not overlap each other). Thus, it is becoming more common in data cleaning to use the term *potential error*. This means that there is an indication that an error might exist, but it is not certain. In the context of automated error detection, the cases where the system has found potential errors where no errors exist in data are considered false-positive results. In alignment with this, all errors that are detected by the automated error detection methods in this thesis are potential errors and need to be checked by humans.

Complex relations in map data

When we think and communicate about space, we often refer to qualitative relationships that exist between spatial objects - the road is *connected* to the bridge, or the house is blocking the access *between* the route end point and the destination building. These spatial relations are also implicitly present in map data and can be used for spatial analysis and reasoning about both spatial and semantic components of data. However, to be able to make use of them, it is first necessary to have the means of modelling spatial relations. While models for binary topological spatial relations between two objects are well established and widely used, they do not enable us to utilise the relations that require simultaneous modelling of more than two objects. Thus, one focus of this thesis is to develop ternary spatial relation models that will enable these more complex spatial analyses to be used for error detection.

1.4 Hypothesis and research questions

This thesis aims at developing computational methods to automate the detection of errors in map databases and ease the demand for human resources in error detection. Given the assumption of mature spatial datasets, the aim is to perform the error detection intrinsically without the need for ground truth or reference data. Furthermore, error detection strategies in this research will be based on the analysis of ternary qualitative spatial relations between objects in map data.

The main hypothesis of this research is that *the analysis of ternary spatial relationships between map features enables the detection of otherwise undetectable errors in map data. This would result in an automated, intrinsic error detection method with formal grounding, eliminating the need for ground truth or reference data.* This hypothesis is approached by addressing three partial research questions, as laid out below.

1. *How can ternary spatial relationships be modelled?* In response to the challenge of *complex relations in map data*, two models for ternary spatial relationships are

proposed in this thesis - *Refined topological relations model for line objects (RTL)* and the *Ray Intersection Model (RIM)*. These models enable us to describe, capture, and analyse relationships between objects in map data that the previously existing models were not able to express. Consequently, they enable more sophisticated strategies for error detection to be developed.

2. *How can errors be intrinsically detected with the analysis of ternary spatial relations?* In response to the challenge of unavailable *reference data or ground truth*, this thesis proposes two approaches for intrinsic error detection based on the RTL and RIM models. One approach is based on the analysis of spatial relations between objects in a map database and detecting the patterns that are formed by the majority of objects. These patterns are then, according to the assumption of mature datasets, adopted as rules to which all objects must conform. The other approach differs in that rules are defined by our heuristics and expectations of relationships between objects in a map database. The objects that violate the defined rules, in both these approaches, are labelled potentially erroneous. Thus, both approaches are intrinsic without the need for the reference or ground truth data.

3. *How can the potential errors be detected and evaluated in a prominent map database?* In response to the challenge of *certainty of errors*, the errors detected by the intrinsic approaches in this thesis are considered potential errors. To demonstrate the proposed approaches and their ability to detect potential errors, they have been applied to one of the most prominent map databases today - the OpenStreetMap. To do this, the ternary spatial relation models and the intrinsic error detection methods have been computationally implemented. This thesis also shows how potential errors can be evaluated by a community of mappers. Their feedback can give a precise understanding of the performance of the intrinsic error detection approaches.

1.5 Outcomes and contributions

The first major contribution of this thesis are the two novel models for ternary spatial relationships - RTL and RIM. The RTL model addresses the problem that the commonly used 9-intersection model [20] is limited to characterising binary topological relations and cannot describe scenarios where more than two objects are involved. This is particularly important in networks, where there may be the need to show that a certain segment is connected to the rest of the network on both sides. Thus, the refinement proposed in this model examines whether a core line object is connected to its surrounding objects on both or only one of its ends (i.e. start and end points).

RIM addresses the problem that there is still no widely accepted model for ternary spatial relations that provides an adequate platform for reasoning about one spatial object being *between* others, as the existing models are either not sufficiently specific regarding betweenness [10], or provide only reasoning about the relation *between*, without an adequate implementation strategy [9]. As a result, the question of whether one spatial object is located *between* two other spatial objects remains hard to answer, especially if we take into consideration that peoples' answers may be subjective, depending on the particular arrangements of these objects. RIM casts rays between two peripheral objects and uses the intersection sets between these rays and the core object to model its relation to peripheral objects. These intersection sets provide a basis for reasoning about the core object being between peripheral objects.

Both these models have been computationally implemented in tools that are commonly used in spatial information processing - Python programming language and PostgreSQL database enabled with PostGIS spatial extension. This makes them easy to apply and use in other spatial analyses such as error detection. The contributions these models make are:

- Refined topological relations for line objects refine existing models of topological relations to describe line-line relationships in more detail;
- Refined topological relations for line objects enables new insights into the connection between topology and function of line objects;
- RIM presents a novel approach to ternary spatial relations, where two objects are treated as peripheral objects and one is the core object whose relative position towards the peripheral objects is characterised;
- RIM provides a basis for reasoning about the betweenness of spatial objects.

The second major contribution of this thesis is its investigation of topology as the defining characteristic for certain classes of spatial objects. Traditionally, a spatial object in a map database would be annotated at its creation and its class would be determined in this way (e.g. a spatial object annotated as a bridge belongs to the bridge class). The problem is that such classification depends on the human judgement which is prone to errors and might be inconsistent. This thesis proposes a data mining approach based on frequent itemset mining to learn frequent topological relation patterns between objects of a given class and other spatial objects. Given a selection of spatial objects, the proposed method analyses topological relations between these objects and other spatial objects that are not disjoint from them. These data are then analysed with frequent itemset mining to determine which topological relationships are defining for objects of the given class. A set of defining topological relationships is adopted as a topological constraint for objects of the given class and objects that are annotated as belonging to that class but do not satisfy the topological constraint (i.e. they do not have all of the necessary topological relationships with surrounding objects) are labelled as potentially erroneous. The approach here implements the RTL model. This thesis makes the following contributions regarding automated error detection by topological data mining:

- It shows that topology strongly contributes to the definition of spatial object classes;
- It provides an automated approach for learning topological constraints from data; and
- It shows that topology can contribute to the detection of spatial object misclassifications and missing-data errors.

The third major contribution of this thesis presents a novel approach that detects errors in map databases by analysing the accessibility of buildings. It is based on the assumption that is commonly made by routing algorithms where the start and destination points of a route are often snapped to the closest road segment. In doing so, they assume that, when the destination is a building the road that is closest to it provides access. However, it may happen that some buildings in map data are not accessible from the closest roads and this indicates potential errors such as missing data (e.g. access road missing from the map), spatial errors (e.g. building inside another building) or even semantic errors (e.g. another object incorrectly annotated as a building). The accessibility analysis here is based on the RIM model and it tests whether the surrounding buildings completely occlude the core building from the closest road, thus preventing access. This thesis makes the following contributions regarding building accessibility:

- it demonstrates a novel problem of building accessibility from the road network where sophisticated reasoning is needed and provided by a computational approach;
- it presents a computational refinement of the RIM for the determination of occlusion between two spatial objects in a 2D plane and a set of potentially occluding objects; and
- it demonstrates an approach for a large scale analysis of building accessibility in a way that enables automated detection of potential errors such as missing road data.

1.6 Thesis structure

This thesis is organised as follows. Chapter 2 reviews the literature in the fields of data quality, data cleaning, and error detection. Special consideration is given to methods and approaches that are aimed at spatial data. This chapter lays the foundations for the entire thesis in that it explains the importance and benefits of automated and intrinsic approaches for detection errors and improving the quality of map databases.

Chapter 3 showcases the shortcomings of binary spatial relation models and proposes the approach to modelling ternary spatial relations. It also sets the base assumptions for modelling ternary spatial relations in this thesis as well as the data that the methods in this thesis apply to.

Chapter 4 presents the RTL model for ternary spatial relations. This chapter also shows the computational implementation of this model and its comparison to an existing spatial relation model.

Chapter 5 introduces the RIM model for ternary spatial relations. Apart from the model definition, this chapter also demonstrates its expressiveness across twenty-three different configurations of spatial objects. The computational implementation of RIM is also presented.

These methods enable more complex analyses of map data that can provide better insight into patterns and regularities that occur.

Chapter 6 proposes automated error detection approaches that are intrinsic and do not require any reference data. These approaches make use of the proposed RTL and RIM models to enable more complex analyses of map data that can provide better insight into patterns that occur. Because of the mature dataset assumption, the objects that violate the patterns of the majority are seen as potentially erroneous in these methods.

Chapter 7 shows three case studies that demonstrate the capabilities of the proposed ternary spatial relation models and error detection approaches to analyse data and detect errors. Each case study shows a set of quantitative results and their evaluation.

This is followed by Chapter 8 that discusses the major findings of this thesis presented in the previous five chapters. The discussion also analyses the limitations and possible improvements to the proposed models and error detection approaches. The conclusions and final remarks of this thesis, together with the directions for future work, are given in Chapter 9.

2

Background

Chapter 2 reviews the relevant literature and sets the background for the research presented in this thesis. It starts with an overview of work on spatial data quality and elements of spatial data quality is presented in Section 2.1. Section 2.2 focuses on the part of data quality literature regarding data cleaning and error detection that comes mostly from the field of computer science. This section then overviews the work on error detection in spatial datasets and classifies the errors that are detected in this thesis. Given that this thesis proposes novel ternary spatial relation models for error detection, Section 2.3 gives an overview of spatial relations. The particular topics that are reviewed in this section are topological relations, spatial relation "between", visibility and accessibility relations. These topics were chosen because of their relevance to the spatial relation models proposed in this thesis. Finally, Section 2.4 introduces the volunteered geographical information and OpenStreetMap – a free and open-source map database that is used in the case studies in Chapter 7. It reviews previous work on assessing and improving the quality of OpenStreetMap, as well as intrinsic quality assurance methods.

2.1 Spatial data quality

There is a long lineage of research in the field of spatial data quality, with multiple books published on this topic (e.g., [21–23]). Land surveyors, the main collectors of spatial data, have always cared about the quality of the data they collect. Traditionally, the goal was to measure and represent the "real world" as closely as possible. In surveying, data quality is related to the quality of measurements and is usually expressed through accuracy and precision. Accuracy represents how close the measured values (e.g., distances and angles) to the real value which is unknown and is to be estimated through repeated measurements. Precision is a measure of how similar are the repeated measurements of the same value. With the addition of attribute and temporal data, as well as division of the data into different themes, spatial data quality has since evolved to a much broader concept.

In their description of spatial data quality, the researchers in this field define different elements of data quality [24–26]. Veregin [24] defines accuracy, precision, consistency and completeness as elements of quality over three dimensions: space, time and theme. According to Wan et al. [25], the number of spatial data quality elements in the literature ranges from five to eleven. They have also been subject to standardisation by international institutions - International Organization for Standardization (ISO) and European Committee for Standardization (CEN). Today, the standards for spatial data quality elements are unified in a single international standard which is most commonly used - ISO 19157:2013: Geographic information – Data quality [11]. This standard defines a data quality element as "a component describing a certain aspect of the quality of geographic data" [11]. Table 2.1 lists the elements of spatial data quality that are present in the ISO standard. ISO and CEN standards are especially important for assuring unified quality reporting of authoritative datasets (e.g., datasets produced by national or international institutions). Van Oort [26] gives a detailed overview of research on spatial data quality, including the elements of spatial data quality as defined by different authors/organisations.

Parts of this chapter are based on published papers [1, 3–5].

TABLE 2.1: Elements of spatial data quality in ISO 19157:2013: Geographic information – Data quality [11]; descriptions taken from [12].

Completeness	Describes how complete a dataset is. A surplus of data is referred to as "Error of Commission", a lack of data in contrast as "Error of Omission".
Logical Consistency	Declares the accuracy of the relations manifested within a dataset. This element can be further subdivided into "intra-theme consistency" and "inter-theme consistency".
Positional Accuracy	Defines the relative and absolute accuracy of coordinate values.
Temporal Accuracy	The historical evolution of the dataset.
Thematic Accuracy	Describes the accuracy of the attributes assigned to a geometry.

The other approach to defining and expressing the quality of data is by determining it's "fitness for use" [24, 27–31]. This paradigm means that the quality of the data depends on the purpose for which it will be used [12] and hence is not absolute. This approach is becoming more common, as spatial data are finding their way towards new user-oriented applications such as navigation [6], location based services [32], and web-maps [33]. The assumption of mature spatial datasets upon which this thesis is based also reflects the "fitness for use" approach because it states that the majority of data in such datasets is of acceptable quality - for the users of the data.

2.2 Error detection in databases

The growing amounts of data that are collected by enterprises and used for improved decision making based on analytics and machine learning models are the main motivation for data cleaning research [13, 17]. Data cleaning, also known as data cleansing, can be defined as a process of detecting and correcting errors and replacing missing data in a database [34]. In [35], data cleaning/cleansing/scrubbing is defined as a process of maintaining the quality of data. In [8], where the focus is set on data consistency, it is defined as "an exercise of detecting errors, and possibly modifying the database, such that the data conforms to a set of data quality rules". In addition, Chu et al. [17] identify data cleaning as a key area in database research because every database analysis needs to consider the effects of dirty data. Regardless of differing definitions, [8, 34, 35] all recognise that data cleaning consists of two parts - detection of errors and rectification of errors. This thesis is focused solely on the detection of errors in map databases and does not attempt to recommend nor suggest how these errors should be corrected. Thus, the remainder of this section focuses solely on reviewing the work in error detection research.

2.2.1 Types of errors and detection techniques

To be able to discuss different approaches to error detection in data cleaning literature, the notion of error must first be defined. Abedjan et al. [13] have defined an error as a deviation from the ground truth value. They have followed up this definition with a more formal one - "given a data set, a data error is an atomic value (or a cell) that is different from its given ground truth" [13]. Most of the other authors first divide errors into different types. Chu et al. [17] have made a distinction between quantitative and qualitative error detection techniques (i.e., corresponding to quantitative and qualitative error types). Quantitative techniques are often used for outlier detection and employ statistical methods to identify abnormal behaviours and errors. Qualitative techniques rely on constraints, rules, and patterns for error detection [17]. A similar taxonomy of error types can be found in [13] where errors are seen as either quantitative (outliers and duplicates) or qualitative (rule violations and pattern violations). Definitions of these errors are listed in table 2.2.

TABLE 2.2: Categorisation of errors from [13].

Outliers	Data values that deviate from the distribution of values in a column of a table.
Duplicates	Distinct records that refer to the same real-world entity. If attribute values do not match, this could signify the error.
Rule violations	Values that violate any kind of integrity constraints, such as NOT NULL constraints and UNIQUENESS constraints.
Pattern violations	Values that violate syntactic and semantic constraints, such as alignment, formatting, misspelling, and semantic data types.

The most comprehensive classification of error detection techniques is given by Chu et al. [17] and Ilyas and Chu [8]. This particular classification is based on three questions - *what* to detect, *how* to detect and *where* to detect (Figure 2.1). Each of these questions corresponds to one classification dimension listed below [8, 17]:

- Error type (*What to detect?*) – Anomaly detection techniques can be classified according to which type of errors can be captured. Ilyas and Chu [8] discuss example violations of various integrity constraints (ICs) along with methods for their automatic discovery, and the main steps involved in detecting duplicate records.

- Automation (*How to detect?*) – Approaches can be classified according to whether and how humans are involved in the anomaly detection process. Most techniques are fully automatic, for example, detecting violations of functional dependencies, while other techniques involve human experts, for example, to identify duplicate records.

- business intelligence layer (*where to detect?*) – errors can occur at all stages of a business intelligence (BI) stack, for example, errors in the source database are often propagated through the data processing pipeline. while most anomaly detection techniques detect errors in the original database, some errors can only be discovered much later in the data processing pipeline, where more semantics and business logic becomes available, for example, constraints on a total budget can only be enforced after aggregating cost and expenses.

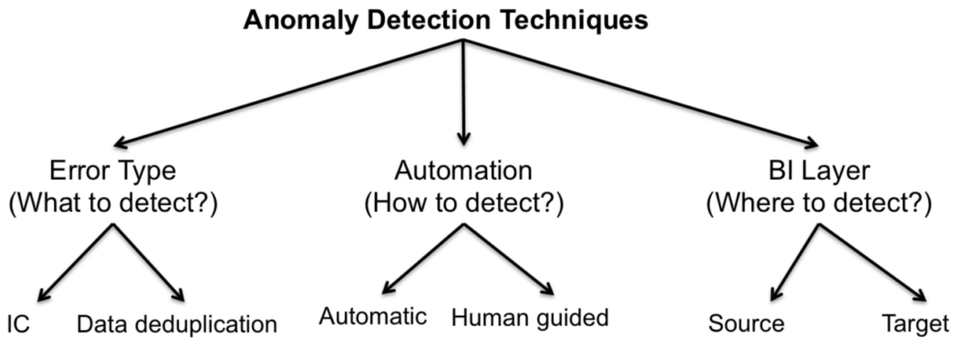


FIGURE 2.1: Classification of anomaly detection techniques by Ilyas and Chu [8].

Abedjan et al. [13] have conducted an overview of different error detection techniques and concluded that there is no single best tool for error detection, but the success of a certain method depends on the dataset. Furthermore, their analysis has shown that percentage of errors that can be detected by existing error detection tools is well under 100%. Similar reviews have been done by Chu et al. [17], by Chandola et al. [18], and by Ciszak [35] who claims that application of data mining methods to data cleaning can be useful in situations where reference data are not available. These reviews come from the area of the general database research and have thus not yet been applied to spatial data quality. However, the problems they have outlined may generalise to all applications of error detection. The error detection approaches proposed in this thesis apply qualitative techniques of detecting errors with rules and constraints, but the definition of these rules and constraints is informed by quantitative analysis of spatial relation patterns in data.

2.2.2 Error detection in spatial datasets

Ones of the first notions of automated error detection in spatial datasets date back to 2000 [36, 37], and were based on checking the consistency of data. Similar to this thesis, the assumption that these approaches were based on is that exceptions to the regularities in the data are probable data errors [19, 36, 37].

Gong and Mu [36] have proposed consistency checking of spatial relationships among neighbouring objects and attribute relationships among data from different sources to detect errors. In their work, they have considered three different types of errors: (1) *spatial inconsistency error* was described as a difference in the position of the same object on different sources (i.e., maps), (2) *logical errors* were described as cases where logical relationships which an object should have with other objects are missing (e.g., a parking lot that is not connected to the road via entrance or exit), and (3) *attribute errors* can be detected by looking at inconsistencies in the attribute data from different sources, similar to detection of spatial inconsistency errors. However, the authors did not go further than conceptualising different ways in which consistency checking can be used to detect errors. With no applicable methodology or framework, these concepts could not be applied to real spatial datasets.

Servigne et al. [37] have developed a framework for checking the consistency of spatial datasets and correction of detected errors. They have also introduced the concept of *topo-semantic consistency* which concerns the correctness of topological relationships between objects, based on their semantics (e.g., a building inside another building would be an error while a building inside a parcel would not be an error). They consider *structural errors* which come from data structures, *geometric errors* which come from the geometric part of objects and *topo-semantic errors* which come from spatial relations between objects. In fact, the topo-semantic errors described in this paper are equivalent to logical errors and geometric errors are equivalent to spatial inconsistency errors from [36]. The structural errors are no longer relevant because the majority of spatial datasets are being stored in spatial databases which deal with these errors efficiently. For detection of topo-semantic errors, they propose topological integrity constraints. They define these constraints using topological relations described by the 9-intersection model [20] (i.e., describes relations of interiors, boundaries and exteriors of two objects), based on the heuristics about certain objects (e.g., building cannot be contained by other building or land parcels must be adjacent). The extension of this model was proposed by Rodríguez et al. [38] in form of measures for evaluating the degree of violation of topological integrity constraints.

Vallires et al. [39] have made an overview of spatial integrity constraints, which they have described as “a desirable state of data”. They have divided spatial relations into metrical relations (i.e., deal with measurable characteristics of objects; e.g., distance and direction) and topological relations (i.e., concern geometric properties that are invariable to transformation; e.g., overlap and disjoint). In that sense, spatial integrity constraints are an extension of topological integrity constraints presented in [37]. Vallires et al. [39] describe inconsistencies in data as violations of spatial integrity constraints which can occur during the data acquisition process or following the integration of data from different sources or of different accuracy. Same as in [37], Vallires et al. [39] claim that the standard way of describing topological relations is the 9-intersection model. International standards concerning spatial integrity constraints were also reviewed in [39]. These mainly concern the use and documentation of the 9-intersection model for describing topological relations. Finally, the authors propose spatial integrity constraints to be manually defined in the form of *object class – spatial relation – object class* (e.g., road segment meets road segment). Validation against the defined constraints can then be used as a way of minimising the amount of errors in spatial databases. This thesis improves on this approach by learning these spatial integrity constraints automatically from spatial relationship patterns in data.

Mülligann et al. [40] have presented a methodology for analysing the spatial-semantic interaction of point features in Volunteered Geographic Information (VGI). Motivated by the lack of reference data for points of interest (POIs), they look at spatial-semantic patterns in the data with the aim of defining the likelihood of different objects to co-occur (e.g., a pub is more likely to occur close to a restaurant than a hospital). Among other uses, they propose this method to be used for attribute recommendation (i.e., advising volunteers which attributes to assign to objects), data cleaning (i.e., labelling objects that do not follow spatio-semantic patterns as possible errors) and coverage recommendation (i.e., highlighting areas where an object is expected to occur but does not). Although the terminology is different from [36] and [37], this can be seen as an extension of spatial consistency checking and/or outlier detection being used for error detection. However, these patterns are detected rather than defined by humans, which is contrasting the heuristic approach of [37]. This is important because it solves the problem where some rules may be left out due to human oversight. The difference with the methods proposed in this thesis is that instead of distance relations they utilise ternary topological and spatial relationships to characterise spatio-semantic patterns between objects. This thesis also supports objects with line and polygon geometries.

Most recent and comprehensive work on this topic was done by Maervoet et al. [19], who have used outlier detection to automatically detect errors in spatial data. They define outliers as individuals which are substantially different from the rest of the data. This work is motivated by the geographical data provider company Tele Atlas, whose infrastructure for quality maintenance uses manually built quality rules against which the data are verified. Similar to the creation of topological integrity constraints in [37], this is a heuristic approach based solely on the knowledge of human experts. To improve this quality maintenance process, the aim of [19] was to develop a method that will automatically detect such quality rules from the data. By doing so, considerable human effort would be saved and some rules that are overlooked by humans could be discovered. The first step of their approach is the use of a frequent pattern miner to discover association rules. After the rules are discovered, they are first examined by human experts in a sanity check, and then the outliers are detected by verification of the data against these rules. The tool they have developed can be used to discover *inter-feature* and *intra-feature* relations of objects in a database. In the case of *inter-feature* relations, they look at spatial relationships between different objects. These regularities correspond to the spatial-semantic interactions from [40], topo-semantic consistency from [37] and logical consistency from [36]. In the case of *intra-feature* relations, they look at relationships amongst attributes on the specified object class (e.g., if the *importance* of a road object is labelled as *high*, then its *condition* should be labelled as *paved*). The issue with the approach from [19] is that it has limited support for spatial computation and spatial relationships. Apart from the pre-processing step where it is determined which objects will be considered by the algorithm, there is no support for spatial relationships such as topological relationships or distance between objects and the patterns are mined in the relational tables that contain objects' identifiers and attributes. The approach in this thesis also mines for frequent patterns to discover rules, but the mining is performed on spatial relationships between objects.

2.2.3 Errors detected in this thesis

Section 2.2.2 shows continuous improvements in the field of error detection in spatial databases. What started as an intuitive idea in [36] has developed into a functioning tool [19] that is able to detect errors in spatial data. Over time, the approaches in this field have moved from the heuristic definition of constraints [37, 39] towards the use of data mining techniques for their automated discovery [19, 35, 40]. However, the *full automation* has not yet been achieved and considerable human guidance is needed in both error detection and error correction tasks. In this thesis, the automated approaches for detection of spatio-semantic errors continue the trend of using data mining techniques for automated error detection. The error detection approaches in Chapter 6 also aim to minimise the need for human guidance.

The classification of errors in this thesis is inspired by the classification of anomaly detection techniques from Figure 2.1. As described in Chapter 1, all errors are not alike and in this thesis they are differentiated by error *type*, *scope*, *cause*, and *detection strategy* (Table 2.3). However, error type described in Chapter 1 differs from the error type described in Figure 2.1 and refers to components of spatial data (i.e., spatial or semantic) that are affected by the error. Error scope refers to how many objects are affected by the error and this notion has not been considered in many of the previous error detection research. The cause of the error is the hardest characteristic of the error to automatically deduce. However, it is often not necessary to know the cause of the error in order to detect the error and even to correct it. For example, if someone has misplaced a house in the map it may be because of outdated aerial imagery or due to using the wrong cartographic projection. Either way, it is only needed to notice the mistake and correct the position of the house. Lastly, the error detection strategy in our classification refers to the method by which the errors are detected. This is similar to the automation and question "how to detect errors?" from Figure 2.1.

TABLE 2.3: Classification of errors in this thesis.

	Definition	Possible values (non-exhaustive list)	Approached in this thesis
Type	Which components of data are affected?	spatial, semantic, spatio-semantic	spatio-semantic
Scope	The number of objects that are affected	one, two, three or more	three or more
Cause	What has caused the error to occur?	reference data error, human error, systematic error	not addressed
Detection strategy	How can the error be detected?	manual, automatic, outlier detection, topological analysis, constraint violation	automated, topological constraints, spatial analysis

2.3 Spatial relations

Corcoran et al. [41] define a spatial relation as "a means of modelling a particular property of the spatial relationship which exists between two or more objects". Spatial relations can be quantitative or qualitative [41] and differ based on the particular property of the relationship they describe (Table 2.4), or by the number of objects involved. As mentioned in the previous section, Vallires et al. [39] have divided spatial relations into metrical relations (i.e., deal with measurable characteristics of objects; e.g., distance and direction) and topological relations (i.e., concern geometric properties that are invariable to transformation; e.g., overlap and disjoint). Egenhofer and Franzosa [42] categorise spatial relations into topological, metric and order relations (i.e., order of objects such as *in front of* and *above*). Clementini [43] categorises spatial relations into topological, metric, and projective relations which are defined as "spatial relations that can be described by projective properties of space without resorting to metric properties". Projective relations include relations such as the align-along-road relation [41], line-of-sight relations [44], and building accessibility relation that is proposed in this thesis. According to Clementini [43], projective relations also include order relations from [42]. Table 2.5 shows the qualitative spatial relations for which models are proposed in this thesis, and each of these three relations is used in one case study in Chapter 7. The rest of this section overviews selected topics in spatial relations that are relevant to the research presented in this thesis.

TABLE 2.4: Categories of spatial relations commonly found in the literature.

Category	Description	Source
Topological	Describe spatial relationships that are invariable to transformation (e.g., touches and intersects)	[39, 41–43]
Metric	Describe spatial relationships that can be measured (e.g., distance and angle)	[39, 41–43]
Order	Describe the order of objects in space (e.g., behind or between)	[41, 42]
Projective	There are other relations not in these categories (e.g., visibility, alignment)	[41, 43, 44]

TABLE 2.5: Qualitative ternary spatial relations analysed in case studies in Chapter 7 of this thesis.

Relation	Description	Category
Line connections	Refined topological relations for line objects (RTL, Chapter 4) describe whether a core line object is connected to other objects on one or both sides (i.e., line ends)	Topological
Between	The Ray Intersection Model (RIM, Chapter 5) can be used to describe whether one spatial object is positioned between other two spatial objects.	Order
Accessibility	RIM can also be used to analyse if one spatial object (e.g., a building) is a accessible from another spatial object (e.g., a road) in a straight line without any obstacles obstructing the way.	Other

2.3.1 Topological relations

Topological relations are one of the most studied kinds of qualitative spatial relations, which find wide applications in GIS, computer vision, natural language understanding, and spatial databases [45]. There have been numerous studies and approaches to formalising topological relations, and a thorough overview can be found in [45]. However, the two most notable streams of research in this field are focused around the RCC model [46], and around the n-intersection model (e.g., 4-intersection model [42], and 9-intersection model [20]). In short, the Region Connection Calculus (RCC) defines a set of relations that may occur between two closed regions (i.e., polygons with interiors, boundaries and exteriors) and describes them with a basic relation $C(x, y)$ (i.e., x connects with y). The n-intersection model is grounded in point-set theory and uses a matrix form to encode if intersections between point-sets such as objects interiors, boundaries, and exteriors exist (i.e., if they are empty or non-empty). Although variants of both models are used today, a variant of the n-intersection model, the dimensionally extended 9-intersection model (DE-9IM) [47], is accepted as the standard model for topological relations by ISO [11], and implemented in most ISO/OGC compliant software. This thesis aligns with the n-intersection model tradition, as the 9-intersection model is extended in the calculation of both models proposed here (RTL in Chapter 4 and RIM in Chapter 5).

The first notable n-intersection model, the 4-intersection model, was proposed by Egenhofer and Franzosa [42] based on point-set analysis of binary relationships between two objects' interiors and boundaries. 4IM was succeeded by the 9-intersection model (9IM) [20] which, in addition to objects' interiors and boundaries, takes into consideration their exteriors to provide more expressivity. As a result, 9IM can differentiate cases that 4IM cannot. Both 4IM and 9IM denoted the intersection of point sets as *non-empty* if they share at least one point between them, and *empty* otherwise. Such intersection test results are usually represented in a 3×3 matrix (i.e., 1 or $True$ for a non-empty point set intersection, 0 or $False$ for an empty point set intersection).

The 9IM has later been extended into a dimensionally extended 9-intersection model (DE-9IM) [47–49] which captures not only the binary results for intersection tests between objects' interiors, boundaries and exteriors, but also analyses the dimensionality of the intersection set. In DE-9IM, the results of intersection test between two point sets would show $False$ for an empty intersection, 1 for a 1-dimensional (i.e., point) intersection, and 2 for a 2-dimensional (i.e., line or polygon) intersection. After DE-9IM, many other extensions of the 9IM followed, such as the Voronoi-based 9-intersection model (V9I) [50] and the 9+-intersection model [51].

However, neither of these models can handle the ternary topological relations in a way that enables to analyse the connections of a linestring on both sides, or the *betweenness* of spatial objects because they support only binary relations (i.e., 9IM, 9+-intersection model, V9I).

2.3.2 Spatial relation "between"

Betweenness has been mentioned in the literature in the context of ternary qualitative spatial calculi. Dylla et al. [52] claim that the importance of ternary qualitative spatial calculi is

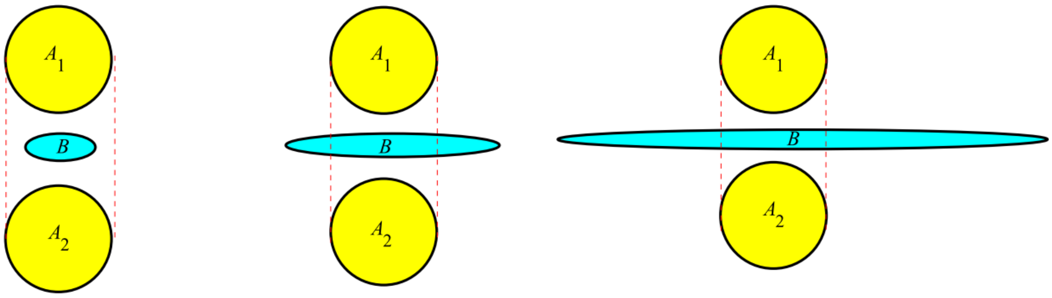


FIGURE 2.2: Three cases from Bloch et al. [9] where the degree to which B is between A_1 and A_2 may vary depending on the context and on the definition.

in the ability to describe relative orientation and betweenness of spatial objects. However, they conclude that a unifying algebraic framework for n-ary calculi is yet to be established. Scivos and Nebel [53] have proposed a calculus for ternary relations of point objects, but it only supports reasoning about the betweenness of collinear points and is not suitable for spatial objects of higher dimensionality (i.e., lines and polygons). Only Bloch et al. [9] have made a thorough analysis of this ternary spatial relation, which they claim is fuzzy and depends on the context. They have proposed multiple geometrical definitions of *between*, each addressing a different problem – ranging from relationships of fuzzy objects to the impact of relative sizes of objects (Figure 2.2).

In the RIM model proposed in Chapter 5, two notions from Bloch et al. [9] are adapted and built on. First, to define if one (core) spatial object is located between two other (peripheral) spatial objects, we adopt the concept of the defining the "area between" those peripheral objects, and only then we analyse the relationship of the core object to this area. Although this notion is not included in the theoretical definition of RIM, it is applied as a convenient shorthand in the computational implementation of RIM (Chapter 5).

The second notion is that parts of the area between two objects should only be taken into consideration if they are visible from both objects. For example, if the area between two objects is defined by the convex hull of the union of the two objects minus the objects themselves (Figure 2.3), then for any concave objects the area inside the concavity would also be treated as between. Instead, Bloch et al. [9] propose that any such concave region be excluded from the area deemed between two objects. This argument is adopted in RIM as well, where the definition of a *ray* achieves the same effect.

Another group of studies on modelling the spatial relation *between* were carried out by

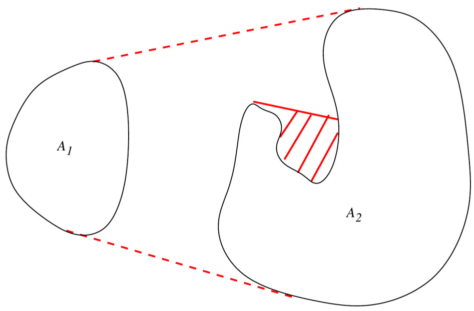
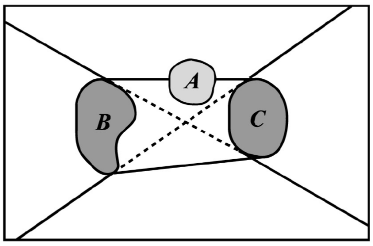


FIGURE 2.3: An example from Bloch et al. [9] where a concavity in A_2 is lacking visibility from A_1 (dashed area).

Clementini and Billen [10], Billen and Clementini [54, 55]. Their approach is based on the collinearity of points, generalised into the collinearity of regions [56]. Their model partitions the plane around two peripheral objects into five separate zones using their common internal and external tangents. The common external tangents of peripheral objects (e.g., B and C in Figure 2.4) are also tangents to the convex hull of the union of the peripheral objects. The common internal tangents intersect inside the convex hull of the union of peripheral objects. A relation of the third (core) object to the peripheral objects is analysed by testing which of the five zones does the core object intersect (Figure 2.4), and this same approach was later implemented by other researchers as a refinement of the visibility based ternary model [57]. Clementini and Billen [10], Billen and Clementini [54, 55] demonstrate that this model for ternary projective relations is able to distinguish between 34 different relations. In their model, object A is considered to be between objects B and C whenever the interior of A intersects the interior of the convex hull of the union of B and C, minus the objects B and C themselves. The shortcoming of capturing betweenness in such binary manner is that some important differences within scenarios where A is *between* B and C cannot be expressed. For example, if A is a bridge, it is important that it touches both B and C, while if A is a fence between two properties, it is important that it completely obscures 2D visibility between B and C. Another shortcoming of the ternary projective relations model is that it is susceptible to the previously mentioned problem of concave polygons. Thus, the RIM model proposed in Chapter 5 enables the expressivity needed to further differentiate scenarios where A is between B and C and it takes into consideration the effect of concave polygons on the visibility between B and C.



$$\begin{pmatrix} & 1 & \\ 0 & 1 & 0 \\ & 0 & \end{pmatrix}$$

FIGURE 2.4: Projective relation between objects B, C, and A [10].

2.3.3 Visibility relations

The RIM model proposed in Chapter 5 of this thesis enables to describe a multitude of scenarios for triplets of spatial objects. The model is focused on capturing the betweenness of spatial objects but is refined in Chapter 6 to analyse the accessibility between two spatial objects (i.e., a road and a building) given a set of occluder objects that may stand in the way. In the building accessibility analysis approach, the 2D visibility between the building and the road is used as a proxy for accessibility. In other words, if occluder objects are positioned between the building and the road in such a way that there is no visibility between them, then it is considered that there is no accessibility between them either. This section reviews the existing research on visibility relations and explains how the approach proposed in Chapter 6 of this thesis contributes to this field.

2.3.3.1 Qualitative spatial reasoning

There have been several studies in qualitative spatial reasoning (QSR) on the visibility and occlusion of spatial objects. Galton [44] was the first to model visibility with *line-of-sight* relations (LOS-14) inspired by the region connection calculus (RCC) [46]. LOS-14 consists of 14 qualitatively distinct configurations of two convex spatial objects as seen by the observer in their field of view (e.g., A hides B or A is in front of B). LOS-14 was also used in [58] to interpret relations between objects in digital images. It was superseded by Köhler's occlusion calculus (OCC) [59] which is slightly different in its calculation and in the composition table used for qualitative reasoning with these relations (e.g., if A partially hides B what are the possible occlusion relations from B to A?). Randell et al. [60] have introduced the region occlusion calculus (ROC-20) which has expanded the LOS-14 by also allowing concave objects in their model. They have also studied the motion parallax and how does the occlusion change when the objects in the scene are moving, and when the viewpoint of the scene is changed. Some interesting applications of ROC-20 include formalisation and exploration of shadows [61] and celestial eclipses [62].

The next improvement was introduced by Tarquini et al. [63] in their "qualitative model for visibility relations" that has enabled the "point of view" to be not only a point but any convex geometry. They have achieved this by modelling the visibility as a ternary spatial relation based on the ternary projective relations model [10, 55, 56]. Their method divides the space based on the projective lines between the point of view object C and the obstacle object B into three types of acceptance areas: a shadow zone, a twilight zone, and a light zone. The third object A is then assigned any of the seven distinct visibility relations (e.g., visible, partially visible or occluded) based on its placement within these acceptance zones. This model has also been applied to localisation and navigation of an autonomous agent with limited sensor abilities [57]. In this application the autonomous agent in each position analyses the obstacle objects in its surroundings with the novel approach of visual cyclic ordering. It then creates a topological map of visible and invisible areas using the visibility relations, and uses this map for qualitative navigation.

Guha et al. [64] have introduced their model that differentiates fourteen occlusion states (OCS-14) in which they additionally consider whether the visible parts in the scene are connected behind the occluder object or not, and whether the occluder object is static or moving, noting that these distinctions are important for computer vision applications. A more recent Interval Occlusion Calculus (IOC) [65, 66] takes into consideration multiple distinct viewpoints of a scene. This enables multiple agents to explore the environment simultaneously. A more detailed overview of visibility relations in spatial cognition can be found in [67], while Dylla et al. [52] places these visibility models inside the larger scope of qualitative spatial and temporal calculi.

The approach proposed in this thesis (Chapter 6) uses rays cast between a line object (e.g., a road) and a polygon object (e.g., a building), which is similar to lines of sight used in some of the approaches mentioned above. However, what sets the proposed approach apart is that it does not classify the entire scene into visible and occluded or light and shadow parts. Instead, it only searches for the existence of at least one line of sight between two designated objects.

While this does not provide a comprehensive visibility analysis of the scene, it enables faster computation for binary visibility analysis (i.e., there is / is no visibility between two objects).

2.3.3.2 Visibility in computer graphics research

In contrast to QSR, visibility analysis research in computer graphics revolves around quantitative approaches where the exact portions of visible objects are determined and the focus is on the efficiency and the speed of algorithms. Multiple approaches for building visibility graphs [68, 69] and visibility complexes [70, 71] have been developed over the years, always further optimising the computation. Similar to some of the QSR research, the goal is usually to analyse the entire scene and determine the visible and invisible parts of all objects in the scene [72], or light and shaded parts if the application is for lighting effects [73].

A thorough overview of the visibility related research in computer graphics has been presented by Bittner and Wonka [74]. It presents a taxonomy of visibility problems according to the problem domain. This taxonomy differentiates five visibility problems: visibility along a line, visibility from a point, visibility from a line segment, visibility from a polygon, visibility from a region, and global visibility. In computer graphics it is the usual practice to map the problem of finding unoccluded lines of sight into the n-dimensional space called the *line space*, and the previously mentioned domains map visibility to line spaces of different dimensions. What is different about the approach proposed in this paper is that it does not require the analysis of an entire scene for visible and occluded parts. It only needs to confirm the existence of a single line of sight between two objects – a road and a building. Given that these objects are usually represented as a polygon and a polyline, the visibility problem in this paper could be described as visibility from a line segment or a polygon depending on which object is designated as a source of visibility. However, the approach for assessing visibility in this thesis (Section 6.5.2) uses functions and operators that are familiar in GIS computations, and it does not map the problem into the line space.

2.3.4 Accessibility relations

Most of the studies on accessibility documented in the literature are concerned either with how accessible the buildings are for people with disability [75, 76], or with the ease of access to a building [77], a location [78, 79], or a service (e.g., such as transport [80], health care [81], or emergency services [82]). Similar to the case study in this thesis, the researchers in the latter group also see the road network as provider of access and they usually use travel time [81] or travel distance [82] to assess the accessibility of a location. However, the notions of accessibility in these studies are different from the one in this thesis where the goal is to assess whether a building is accessible from the road network without there being an obstacle between, physically preventing access. Consequently, these studies are not concerned with the quality of the map data in their spatial analyses. They use the usual routing methods that snap the origin and the destination locations of the route to the closest point in the road network, thus potentially suffering from the exact problem that this thesis is aiming to solve.

2.4 OpenStreetMap

This thesis proposes two approaches for detection of errors in map databases in Chapter 6. To test and evaluate these approaches, we need a map dataset that is sufficiently detailed and large to support data mining approaches and for which the assumption of mature spatial dataset can be made. One option are the proprietary datasets produced by local governments and agencies. They usually come with quality assurance, but might be limited by theme (e.g., a road dataset) and local extent (e.g., many datasets are limited to the state or country). These datasets also come at a certain cost and their use in research might be limited by license agreement. The other option are the volunteered geographical information (VGI [83]) databases that are created by a community of mappers and come free of charge. OpenStreetMap (OSM)[1] is currently the most prominent VGI map database. It covers the entire globe in much detail and it has a long track record of community edits that assure that data are of good quality and up to date. Because of this, all case studies in Chapter 7 are carried out with the OSM data. Moreover, given that the OSM is a global dataset, the approaches proposed in this thesis can be applied to any region in the world without any additional adaptations.

[1] https://www.openstreetmap.org/

2.4.1 OSM data structures

OSM represents physical features of the real world with three basic data structures (nodes, ways and relations), described by attributes in form of tags[2]. Nodes are defining features with point geometries, ways are defining linear features and boundaries, while relations capture how complex arrangements of features relate together (Table 2.6). In most studies, node and way features of OSM are analysed, leaving relation features out because of their complexity and relative under-representation in the dataset [84]. The same applies for this thesis.

The annotation model of OSM is non-restrictive, meaning that contributors can create and assign tags as they see fit. Features (i.e., objects in OSM) can have an unlimited number of tags, as well as no tags at all. The only restriction is that all attribute keys used on the same feature have to be unique (they form a set). As an attempt to standardise feature annotations, OSM provides contributors with *tagging guidelines*[3] through its OSM Wiki website. Table 2.6b shows some of the most important OSM tag keys (i.e., attribute names) that are used to annotate spatial objects. *Common attributes* (*id*, *user*, *uid*, *timestamp*, *visible*, *version*, *changeset*) are not created by contributors, but automatically generated for every OSM feature, and are thus excluded from the analyses in this thesis.

2.4.2 OSM data quality

There has been much work done on the quality assessment of the OpenStreetMap data. Some of the first studies have evaluated OSM by comparing it to proprietary datasets [85–87], mostly to find that the completeness and accuracy of the OSM data is higher in urban areas. A more recent study [88] claims that OSM has achieved more than 80% completeness for world roads, with more than 40% of countries having a fully mapped street network (i.e., $>$ 95% streets are mapped). Of interest for this thesis are also the studies that have tested OSM data and its fitness for use in car navigation and routing [89, 90], as well as the quality of mapped building footprints [91]. Graser et al. [89] have compared the OSM data to the official Austrian reference graph for routing in Vienna, showing that the median difference of the route lengths in the two datasets is only 1%. Zhang and Ai [90] has focused on quality

[2] http://wiki.openstreetmap.org/wiki/Elements
[3] http://wiki.openstreetmap.org/wiki/Category:Tagging_guidelines

TABLE 2.6: OSM data elements and important OSM tag keys that demonstrate the structure of OSM data (sourced from https://wiki.openstreetmap.org/wiki/Beginners_Guide_1.3).

OSM data elements (all elements carry tags)	
Nodes	Dots used to mark locations. Nodes can be separate or can be connected.
Ways	Connected line of nodes. Used to create roads, paths, rivers, and so on.
Closed ways	Ways that form a closed loop. Usually form areas.
Areas	Ways that form a closed loop. Usually form areas.
Relations	Used to create more complex shapes, or to represent elements that are related but not physically connected.

(A)

Important OSM tag keys	
Highway	For tagging highways, roads, paths, footways, cycleways, bus stops, etc.
Place	Used for tagging countries, cities, towns, villages, etc.
Amenity	Used for tagging useful amenities like restaurants, drinking water spots, parking lots, etc.
Shop	Used for tagging shops that you buy products from.
Building	Used for tagging buildings.
Landuse	Used for tagging land being used by humans.
Natural	Used for tagging natural land like forests.

(B)

issues of OSM data related to routing such as tagging, modelling of junctions as dual carriageways, and missing or inconsistent road names. They also briefly mention that missing segments are one of the problems that may cause incorrect routing results, which is the same problem that is discussed in Chapter 7. Touya and Brando-Escobar [92] have analysed the level of detail (LoD) inconsistencies in OSM by automatically inferring the LoD for map objects.

With the rapid growth, in certain areas OSM data has become richer, more complete, and faster updated than many authoritative datasets. This means that there are no reference data that OSM can be compared to for quality assessment. Consequently, there is an increasing trend for developing intrinsic approaches for assessing the quality of OSM data [12, 29, 30, 84, 93, 94], as well as detecting and fixing errors [1, 5, 28, 90, 95]. Intrinsic approaches use solely OSM data, which solves the problem of reference dataset unavailability. The approaches proposed in Chapter 6 can be considered intrinsic approaches to detecting errors since they do not need any other dataset than the one being analysed - OSM.

All of these studies confirm that the OSM can be considered a mature spatial dataset where majority of the data are of acceptable quality. They also confirm that the roads and buildings data in OSM are fit for use in routing and similar spatial analyses such as the ones performed in this thesis.

3

Modelling ternary relations

This chapter discusses the issues related with the inadequate expressiveness of binary models for capturing complex spatial relationships. It focuses on the 9IM model for binary topological relations as it is the most widely accepted across different standards and tools. This chapter also sets the foundations for modelling ternary spatial relations. These foundations will support the proposals of two novel ternary spatial relationship models in Chapters 4 and 5. By providing better expressiveness, the proposed models will enable more sophisticated error detection approaches presented in Chapter 6 and the automated detection of errors in map databases in Chapter 7.

Parts of this chapter are based on published papers [1, 3, 4].

3.1 Assumptions

Before explaining the specific issues with spatial relations and the approaches for their modelling, we need to define which spatial objects are considered. This thesis is concerned with vector objects that are purposed for creation of generic maps such as OpenStreetMap, Google Maps, and Bing Maps. This excludes trajectories of moving objects and vector data aimed at specialised maps such as ocean current vectors, vector fields and meshes.

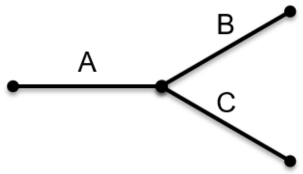
All vector objects in this thesis are set in a 2D plane and assumed to comply with the OGC simple features standard [96]. Objects can be represented using three geometry types: point, line or linestring, and polygon. Objects can have single-part or multi-part geometries (i.e., multi-points, multilinestrings, multipolygons). Finally, objects are assumed to be valid and not to form any self-intersections.

3.2 Inadequate expressiveness of binary relations

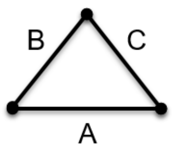
Binary topological relation models have long been the standard for capturing the topological relationships between spatial objects. However, their most limiting factor is that they have to isolate the relationship between only two objects. This means that if there are other objects that are potentially forming more complex relationships with the two objects, they will be disregarded when modelling. In other words, binary models will always constrain us to capture the relationships between objects as binary, even when they might involve more objects.

In many cases ternary relationships are important for spatial reasoning and analysis. One example are objects such as roads and rivers that are represented as lines in map databases. *Flow* is often the most important functional property of these objects. The river can flow between two lakes and the road can carry a traffic flow between two intersections. Thus, the topological relations between the ends of these objects (i.e., their start and end points) and other roads and rivers are important because they enable a certain flow to happen. For example, if a road is connected to other roads only on one end, then this is a no-through road and its traffic flow will be restricted. In a more general sense, ternary relations are

(A) Scenario 1



(B) Scenario 2



(C) Scenario 3

FIGURE 3.1: Three topological scenarios between three line objects *A*, *B*, and *C*.

TABLE 3.1: 9IM between object *A*, *B*, and *C* in Figure 3.1.

	Interior	Boundary	Exterior
Interior	*F*	*F*	*T*
Boundary	*F*	*T*	*T*
Exterior	*T*	*T*	*T*

(A) 9IM for the topological relation *meets*

	Interior	Boundary	Exterior
Interior	*F*	*F*	*T*
Boundary	*F*	*F*	*T*
Exterior	*T*	*T*	*T*

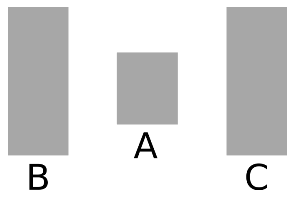
(B) 9IM for the topological relation *disjoint*

	Scenario 1	Scenario 2	Scenario 3
A and *B*	*meets*	*meets*	*meets*
A and *C*	*meets*	*meets*	*meets*
B and *C*	*disjoint*	*meets*	*meets*

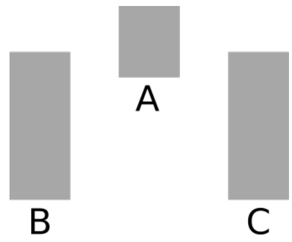
(C) Topological relations between objects in three topological Scenarios from Figure 3.1

often needed because binary relations are ambiguous - e.g., binary relation {*A right of B*} is ambiguous while ternary relation {*A right of B with respect to C*} is not.

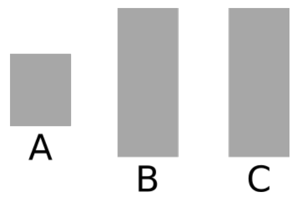
Figure 3.1 shows three different scenarios that include three objects with line geometries. In all three scenarios, objects *B* and *C* share exactly one point with object *A*. Furthermore, topological relations between *A* and *B* or *C* have the same 9IM representation in all cases from Figure 3.1 – *meets* (Table 3.1a). Table 3.1c shows the topological relations between



(A) Scenario 1



(B) Scenario 2



(C) Scenario 3

FIGURE 3.2: Three topological scenarios showing different arrangements of spatial polygon objects A, B, and C.

TABLE 3.2: Topological relations between objects in three topological scenarios from Figure 3.2

	Scenario 1	Scenario 2	Scenario 3
A and B	*disjoint*	*disjoint*	*disjoint*
A and C	*disjoint*	*disjoint*	*disjoint*
B and C	*disjoint*	*disjoint*	*disjoint*

objects A, B, and C in all three scenarios. From the first two columns of that table, it can be seen that these topological scenarios cannot be distinguished using only the 9IM representation of the topological relations between $\{A, B\}$ and $\{A, C\}$. On the other hand, if the topological relations between $\{B, C\}$ were also considered, the 9IM could show that Scenarios 1 and 2 are topologically different. However, in this case, Scenario 3 would be identical to Scenario 2 and different from Scenario 1 (Table 3.1c). If A was an object such as a bridge or a tunnel, the important distinction between these cases would be to capture the connectivity to other objects such as road segments *on both sides*, and the given examples show that a binary model such as the 9IM is not able to do that. Instead, we need a model that can also take into consideration the start and end points of the lines such as the Calculus Based Model [49] or the RTL model proposed in Chapter 4.

Another example is when we want to reason about the overall spatial arrangements of objects, not just the connections between them. We might want to answer the question “is object A positioned between objects B and C?”. Regarding this specific spatial relation *between*, the existing models are either descriptive at a courser granularity than the model proposed in Chapter 5 [10], or only provide reasoning about the relation *between*, without an adequate implementation strategy [9]. As a consequence, reasoning about different arrangements of three spatial objects is currently difficult until these relations are adequately captured by a ternary model.

Figure 3.2 shows three scenarios of different arrangements of spatial objects and the Table 3.2 describes them with 9IM relations. In these scenarios, all pairs of objects $\{A, B\}$, $\{A, C\}$, and $\{B, C\}$ can be represented with the same binary topological relation - *disjoint* (Table 3.1b). In other words, regardless of how we arrange these objects, their relationships will always remain identical with the 9IM. This is problematic because it means that all information about objects' positions relative to each other in these cases are lost due to inadequate expressiveness of the model. If we want to answer the question "is A between B and C?", we will not be able to do that. Thus there is a need to develop models capable of capturing such ternary and potentially n-ary relationships between spatial objects. However, the scope of this thesis is limited to ternary relationships.

3.3 Approach to ternary relationship analysis

Apart from the expressivity issues identified in the previous section, another problem with binary relation models is that they treat objects as equal participants in a relationship. The distinction between objects generally does not matter as binary models aim to equally describe both objects. The only difference that may be relevant is the order in which the objects are listed when the relationship is defined. For example, the same topological relationship can be defined as A *contains* B and B *is contained by* A. However, if objects in ternary relationships were also treated as equals, this would always create triangles of relations such as $\{A$ disjoint B, B disjoint C, and A disjoint $C\}$ for objects in Figure 3.2. As shown in the previous section, this approach results in limited expressivity and is not suitable for capturing ternary relations.

Instead, we focus our attention on one object that we want to characterise, and other objects with respect to which we want to characterise it. This thesis differentiates the *core* and *peripheral* objects in ternary relationships. The *core* object is the object that is being described by the ternary relationship. In the previous example of "Is O between A and B?", O is the *core* object we want to characterise as either between or not. This leaves the remaining objects in the ternary relationship as *peripheral* objects. Previous literature on frames of reference refers to the *core* object as *primary* object and to *peripheral* objects as *reference* objects [97, 98]. However, we prefer the terminology of *core* and *peripheral* objects because their meanings relate well to the problem of betweenness of spatial objects tackled in Chapter 5.

Depending on the specific model, there may be different constraints defined for *core* and *peripheral* objects. For example, we might constrain a *core* object to a line geometry if we are interested in connectedness of roads, or we may assume that *peripheral* objects may touch but must not overlap each other if *peripheral* objects are buildings. However, in this thesis we assume that there always exist exactly one *core* object and two *peripheral* objects in a ternary relationship.

3.4 Conclusion

This chapter has defined the scope of spatial objects that are considered in this thesis. In short, this thesis applies to the simple, valid, 2D vector objects in map datasets. It does not concern raster or other types of vector objects (e.g., 3D geometries).

This chapter has also demonstrated the lack of expressiveness when certain scenarios are described with binary relations. This highlights the need for more expressive spatial relation models that can simultaneously capture the relationships between more than two objects. As a response, two ternary spatial relation models will be proposed in Chapter 4 and Chapter 5. Both of those models are built on the principle of one *core* object being described relative to two *peripheral* objects, as described in this chapter.

4

Refined topological relations for line objects

This chapter proposes the refined topological relations for line objects (RTL) model to capture whether the linear objects such as roads and rivers are connected on both sides or only on one side. This is an important distinction that determines whether the network traffic can propagate further through that line segment. This chapter also presents the implementation of the RTL model and compares its expressiveness to that of the 9IM. The RTL model developed here will be used in the later chapters of this thesis. Chapter 6 utilises it in an approach for learning the patterns in topological relations between objects in map databases. Chapter 7 uses the RTL based approach from Chapter 6 to detect errors in a map database.

The content of this chapter is based on the peer reviewed journal paper titled *"Discovery of topological constraints on spatial object classes using a refined topological mode"* [1].

4.1 RTL model definition

The RTL model analyses the binary 9IM topological relations between the *core* line object and its *peripheral* objects that are topologically related to it. Its aim is to find out if these topological relations are symmetrical (i.e., present on both sides, like in Figures 3.1a and 3.1c) or asymmetrical (i.e., present only on one side of the line, like in Figure 3.1b). This refinement enables new insights to be gathered from spatial objects in map data, such as whether a road segment is connected to other road segments on both sides or only one (i.e., a no-through street). This is applicable to all objects that form linear networks in maps (e.g., tramways, railways, cycleways, power-lines) where it is an important distinction if a network segment is connected on both sides, or rather, if it is disconnected on one side. This would mean that the network traffic cannot propagate further through that disconnected segment end.

Symmetry in the RTL model is only defined and tested for peripheral objects which are topologically related to the end points of the core object (i.e., is not defined for a peripheral line crossing the core object). The start and end points of the core line object do not relate to its direction but are terms used only to differentiate the line object's two extremes. They are extracted directly from the geometry of the line object and are usually determined at the time when the geometry is created (i.e., the first point of the geometry that user enters will be its start point, and the last entered point will be its end point). Thus, the approach proposed here differs from approaches based on directed line segments [99] or oriented regions [100] that are often used for modelling movement trajectories. Also, we do not make any further claims of the mathematical properties of this model (e.g., homeomorphism).

The computational implementation of the RTL model is achieved in two steps. Applied in combination, *Algorithm 1* and *Algorithm 2* enable us to capture the relationships between more than two non-disjoint objects through their refined topological relation.

The goal of Algorithm 1 is to compute the topological relation between the core object and a single peripheral object in a way that shows if the peripheral object is connected to the start point or the end point of the core object. It requires a single core line object and a single peripheral object as input (e.g., objects *A* and *B* from Figure 3.1). Although the core object may have many peripheral objects, Algorithm 1 only considers one peripheral object at a time, and ought to be run separately for all peripheral objects. Let the geometry of the core object be called *geom1*, and the geometry of the peripheral object be called *geom2*. First, the

Algorithm 1 Refined topological relations for line objects

Require: A line geometry (geom1) of one object from the class that is being analysed (e.g., bridge), and a line or polygon geometry of another object (geom2)

Input: $REL \leftarrow relate\_9IM(geom1, geom2)$, $RS \leftarrow relate\_9IM(start\_point(geom1), geom2)$, $RE \leftarrow relate\_9IM(end\_point(geom1), geom2)$

1: In RS and RE, keep only the 3 values that regard the relations between the points' interiors and the other object
2: Change all positive (True) values in RS to 'S' in order to denote that this relation refers to start point
3: Change all positive (True) values in RE to 'E' in order to denote that this relation refers to end point
4: $i \leftarrow 0$
5: $result \leftarrow \}\}\varepsilon$
6: **while** $i \leq 3$ **do**
7:     $x = RS[i] + RE[i]$
8:     **if** $x =$ `FF` **then**
9:         $y \leftarrow$ `F`
10:     **else if** $x =$ `SF` **then**
11:         $y \leftarrow$ `S`
12:     **else if** $x =$ `FE` **then**
13:         $y \leftarrow$ `E`
14:     **else if** $x =$ `SE` **then**
15:         $y \leftarrow$ `B`
16:     $result \leftarrow result + y$
17:     $i \leftarrow i + 1$
18: In REL, replace the values regarding the boundary of geom1 with result – a combination of relations between start point of geom1 with geom2, and end point of geom1 with geom2
19: $REL[3:6] \leftarrow result$
20: **Return** REL

9IM relation between *geom1* and *geom2* is calculated. Then, the 9IM relations between each of the *geom1*'s boundary points (the start and end point of a line constitute its boundary) and *geom2* are calculated. These steps produce three topological relations which are used as input in the Algorithm 1; one between *geom1* and *geom2* (i.e., REL in Algorithm 1), one between the start point of *geom1* and *geom2* (i.e., RS in Algorithm 1), and one between the end point of *geom1* and *geom2* (i.e., RE in Algorithm 1).

Next, these three topological relations need to be conflated into one topological relation between *geom1* and *geom2*, which also distinguishes the side of *geom1* that *geom2* is related to. Unlike line geometries, point geometries do not have a boundary. If a point geometry is related to another geometry, its interior must be related to that other geometry (for more examples and specifics of topological relations with different geometries see [101]). As a result, in the last two topological relations above, only the parts of the 9IM that consider the interior of the point are considered. As shown in Table 3.1a, this information is usually contained in the first three elements (i.e., first row) of the 9IM (Algorithm 1 line 1).

For the first three values in the 9IM between the start point of *geom1* and *geom2*, `T` values are substituted with `S` to denote that *geom2* is topologically related to the start point of *geom1*.

Similarly, in the 9IM for the end point of *geom1*, `T` values are substituted with `E`. As a result, topological relations between start and end point of *geom1* and *geom2* are now each represented with three values that show if the interior of the point is related to the interior, boundary, and exterior of *geom2* (Algorithm 1 lines 2-3).

To combine these relations into one relation, each of the three values in relation for the start point of *geom1* and *geom2* is compared to the corresponding value in the relation between the end point of *geom1* and *geom2* (Algorithm 1 lines 6-16). If both values are negative (`FF`), this means that neither the start nor the end point of *geom1* intersect the interior, boundary, or exterior of *geom2*). Thus, `F` is returned as a value for the unified relation in this case. If values are `SF`, only the start point of *geom1* is related to *geom2*, and `S` is returned. Accordingly, if values are `FE`, only the end point of *geom1* is related, and `E` is returned. Finally, values `SE` mean that both start and end point of *geom1* are related to *geom2*, and `B` is returned into the unified relation. At this point, a three-value representation of the topological relation between the start and end point of *geom1* and *geom2* is available. This representation is finally used to replace the values in the standard 9IM between *geom1* and *geom2* that regard the boundary of *geom1* (Algorithm 1 lines 20-21). These values are usually in the second row of the 9IM matrix (Table 3.1a).

Algorithm 2 shows the second step of the 9IM refinement. Here, the topological relations between the core object and all the peripheral objects in the dataset have already been analysed with Algorithm 1. The goal of Algorithm 2 is to analyse these relations to see if they are symmetrical for the core object.

Because of Algorithm 1, relations for objects that are related to the boundary of the core object are encoded with `S` if they are only related to its start point, `E` if they are only related to its end point, and `B` if they are related both to its start and end point. For relations that are encoded with `S` or `E`, this algorithm determines if the same relation is present on the opposite side of the core object. To test this, when a relation contains `S` or `E`, its reverse relation is calculated. This is done by substituting all occurrences of `S` in the relation with `E`, and by substituting all the occurrences of `E` with `S` to construct the same topological relation, but touching the other point of the core object's boundary (Algorithm 2 lines 3-4). Then if the reverse relation is also present in the relations that the core object has with other peripheral objects, it can be concluded that the relation is symmetrical.

Algorithm 2 Grouping refined topological relations for a line object

Require: A set of all topological relations that one line object has with other objects, where topological relations are represented as refined topological relations for line objects as per Algorithm 1

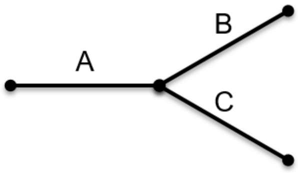
```
1:  result ← []
2:  for relation in relations do
3:      if S in relation and E in relation then
4:          Calculate the reverse relation by replacing all occurrences of S in relation with E, and by replacing all occurrences of E in relation with S
5:          if reverse in relations then
6:              Replace all occurrences of E and S in reverse relation with T
7:              reverse_relation = reverse_relation + -both
8:              result = result + reverse_relation
9:              Delete the first occurrence of reverse_relation in relations array
10:         else
11:             Replace all occurrences of E and S in reverse relation with T
12:             reverse_relation = reverse_relation + -single
13:             result = result + reverse_relation
14:     else
15:         result = result + relation
16: Return result
```

If a relation is present on both sides of the core object, it gains a suffix `both`. On the other hand, if a relation is present only on one side of the core object, it is added a suffix `single` (Algorithm 2 lines 5-14).

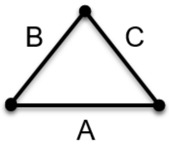
4.2 Comparison of the 9IM and the RTL model

To illustrate the differences between the RTL and the 9IM, Table 4.1 shows their comparison based on the topological scenarios from the Figure 4.1. Figure 4.1 shows the scenarios that include more than two peripheral lines (Figures 4.1d, 4.1e, and 4.1f), and scenarios where peripheral objects have polygon geometries (Figures 4.1h and 4.1i). Figures 4.1g and 4.1j show scenarios where the core line object is connected to a single peripheral object on both sides. In these examples it can be seen that Algorithm 1 substitutes occurrences of `T` with `S` and `E` to encode if the object is related to the start or the end point of the line geometry (i.e., object A in Figure 4.1). This information is then used in Algorithm 2 to explore if a relation is symmetrical and present on both sides of the core line geometry (Figure 4.1a), or if it is present only on one side of the geometry (Figure 4.1b). It is apparent that the RTL model offers superior insight because it recognises that the core line object is connected on both sides in six scenarios in Table 4.1 (i.e., scenarios 1, 3, 4, 5, 6, 8), while keeping the same expressiveness as the 9IM in the remaining four scenarios.

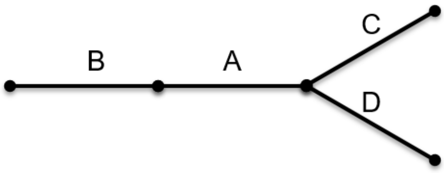
(A) Scenario 1



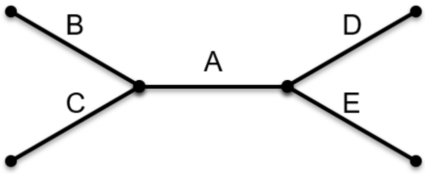
(B) Scenario 2



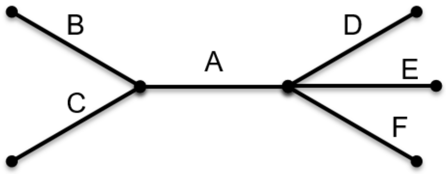
(C) Scenario 3



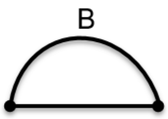
(D) Scenario 4



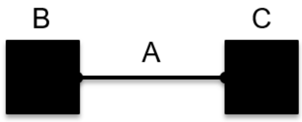
(E) Scenario 5



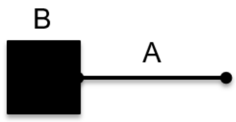
(F) Scenario 6



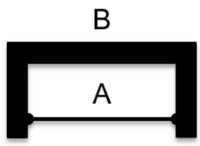
(G) Scenario 7



(H) Scenario 8



(I) Scenario 9



(J) Scenario 10

FIGURE 4.1: Ten different topological scenarios between spatial objects. First three scenarios are duplicated here from Figure 3.1 for improved legibility.

TABLE 4.1: Topological relations between the core object *A* and other peripheral objects in Figure 4.1 represented with the 9IM, Algorithm 1, and Algorithm 2.

	9IM	**RTL**	
Scenario		Algorithm 1	Algorithm 2
Scenario 1	FFTFTTTTT,FFTFTTTTT	FFTFSETTT,FFTFESTTT	FFTFTTTTT-both
Scenario 2	FFTFTTTTT,FFTFTTTTT	FFTFESTTT,FFTFESTTT	FFTFTTTTT-single, FFTFTTTTT-single
Scenario 3	FFTFTTTTT,FFTFTTTTT	FFTFSETTT,FFTFESTTT	FFTFTTTTT-both
Scenario 4	FFTFTTTTT,FFTFTTTTT, FFTFTTTTT	FFTFSETTT,FFTFESTTT, FFTFESTTT	FFTFTTTTT-both, FFTFTTTTT-single
Scenario 5	FFTFTTTTT,FFTFTTTTT, FFTFTTTTT,FFTFTTTTT	FFTFSETTT,FFTFSETTT, FFTFESTTT,FFTFESTTT	FFTFTTTTT-both, FFTFTTTTT-both
Scenario 6	FFTFTTTTT,FFTFTTTTT, FFTFTTTTT,FFTFTTTTT, FFTFTTTTT	FFTFSETTT,FFTFSETTT, FFTFESTTT,FFTFESTTT, FFTFESTTT	FFTFTTTTT-both, FFTFTTTTT-both, FFTFTTTTT-single
Scenario 7	FFTFTFTFT	FFTFBFTFT	FFTFBFTFT
Scenario 8	FFTFTTTTT,FFTFTTTTT	FFTFSETTT,FFTFESTTT	FFTFTTTTT-both
Scenario 9	FFTFTTTTT	FFTFSETTT	FFTFTTTTT-single
Scenario 10	FFTFTFTTT	FFTFBFTTT	FFTFBFTTT

4.3 Conclusion

This chapter has proposed a refinement of the 9IM for line objects. The proposed RTL model addresses the problem presented in Chapter 3, where it was demonstrated that for some configurations of at least three spatial objects the 9IM is not able to make distinctions that are important for networks. RTL captures these nuances by analysing the symmetry of topological relations that a core line object has with peripheral objects (e.g., connected on single end vs both ends).

The computational implementation of the RTL model was also presented. This is a two-step implementation, and each step was presented with an algorithm. To make the RTL model applicable to map data, it has also been realised in the form of a PostgreSQL function and is available as open-source software (https://github.com/SelfHealingMapsProject/Extended-topological-relations-for-line-objects). This enables the RTL to be easily applied to spatial data that is stored in a PostgreSQL database.

The expressiveness of the proposed RTL model was compared to the 9IM across ten scenarios showing different configurations of spatial objects that can be expected in spatial datasets. This comparison has demonstrated that the RTL successfully differentiates between scenarios that the 9IM cannot.

Chapter 6 uses RTL to capture important information in an approach for mining and learning topological constraints from data. In Chapter 7 this RTL based approach is used to test if bridges are connected to other objects on both ends as expected. Bridges that are connected to other object only on one end are considered to be potentially erroneous. Given the large number of bridges, such analysis would not be possible without the RTL model in an efficient manner.

5

The ray intersection model

In Chapter 3 we have identified a problem where the 9IM is not able to describe arrangements of three disjoint spatial objects (Figure 3.2) and support analyses such as "is core object between the peripheral objects?". Since the RTL model proposed in Chapter 4 only supports peripheral objects that are not disjoint from the core object, there is still a need for a more generic ternary model that will be applicable outside of linear networks. This chapter proposes a novel model for ternary spatial relations called the Ray Intersection Model (RIM) to fulfil that need. Apart from the formal definition of RIM, this chapter also demonstrates its expressiveness on twenty-three different scenarios of three spatial objects and by calculating the number of theoretically possible distinct RIM cases. It also shows the implementation of RIM. In Chapter 6, RIM will be modified and applied in the approach for the building accessibility assessment. Chapter 7 will show two case studies where RIM is utilised to analyse betweenness of university campus buildings and to detect errors in map databases by

The content of this chapter is based on the peer reviewed journal paper titled *"RIM: a ray intersection model for the analysis of the between relationship of spatial objects in a 2D plane"* [3].

finding inaccessible buildings.

5.1 RIM model definition

Consider a core object O and two peripheral objects A and B in a 2-dimensional plane (Figure 5.1), such that they may touch but not overlap each other. The RIM defines rays between the two peripheral spatial and determines the spatial relation of the core object to the two peripheral objects based on its intersections with these rays. These binary intersections between individual rays and the core object O in RIM are calculated using the 9IM and then combined into a single ternary RIM relation (i.e., RIM(A,B,O)).

A ray is defined as a straight line segment which shares exactly one end point with each of the peripheral objects A and B. The set of all rays is called *the ray set* and is denoted by R. R may be an empty set, a set of one ray, or an infinite set, and its elements will be denoted in the style r_i. The area covered by all the existing rays between A and B is called the ray area (R_A) and is used to represent the space *between* A and B. In point-set topology, spatial objects such as O, A, B, and each of the rays $r_i \in R$ can be considered point sets. As such, they consist of an interior (O°), a boundary (∂O), and an exterior (O^-). For peripheral objects represented by a polygon, rays will only touch their boundaries. For objects represented by a line, rays may also touch their interiors (Figure 5.2), and for objects abstracted to points (i.e., points have no boundary), rays will only touch the point's interior.

One type of rays that can be distinguished within R is the *extreme rays*. They enable the model to distinguish if the core object O is extending outside of the R_A or not. Remember that for point x in topological space X, a subset $U \subset X$ is called a neighbourhood of x if there is an open set V such that $x \in V$ and $V \subset U$ [102]. Extreme rays will be those rays or parts of rays that coincide with the boundary of the R_A, and whose every point has a point

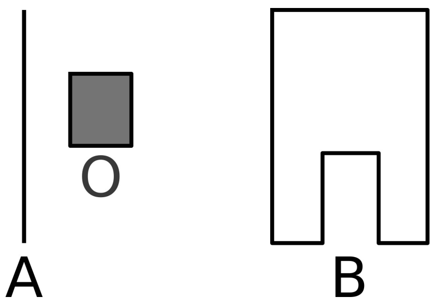


FIGURE 5.1: An example of the the core object O, and peripheral objects A and B in a 2D plane.

inside the R_A and a point outside the R_A in its neighbourhood. Extreme rays will be denoted in the style e_i.

In most scenarios where peripheral objects are disjoint there will be two extreme rays between them (e.g., as in Figure 5.2 where they are denoted as e_1 and e_2). However, there are some configurations of three polygon objects where the number or shapes of extreme rays may differ, and six such scenarios are illustrated in Figure 5.3. Figure 5.3a shows a case where the two extreme rays are intersecting each other. Because parts of these rays are completely within the R_A, only the parts for which each point of the ray has a point that is outside of the R_A in its neighbourhood are considered as extreme rays. In this case, extreme rays are only parts of rays between A and B, and the term *ray* is used to preserve the simplicity of the model. In the case in Figure 5.3b there is only one extreme ray due to the triangular shape of the R_A where two sides are boundaries shared with peripheral objects. Figure 5.3c shows a case where one peripheral object is spiralling around the other. In this case, there will be only one ray that coincides with the boundary of the R_A, and the part of that ray that is not within the R_A will be an extreme ray. Figures 5.3d and 5.3e show the cases where peripheral objects are enclosing the R_A from all sides. Because the R_As share all boundaries with peripheral objects, no extreme rays exist. The scenario in Figure 5.3f shows peripheral objects that are touching in such a way that there does not exist a straight line sharing exactly one point with each of them (i.e., a ray). Thus, there are no rays at all between peripheral objects. Also, the number of rays and extreme rays may vary with different types of 2D geometries. For example, if both A and B are point objects, there will exist only one ray between them, and this ray will also be a single extreme ray. Thus, the extreme rays can be expressed as $E = \{e_i\}$, where $i \in \{0, 1, 2\}$ and $E \subseteq R$. Similar to RIM, Billen and Clementini [56] have defined collinearity regions between polygons (i.e., equivalent to the ray area in Figure 5.2) as areas bounded by their common external tangents (i.e., equivalent to extreme rays in Figure 5.2), but this approach cannot define ray areas in all cases from Figure 5.3.

A point-set intersection model such as 9IM may be used to describe the relationship between object O and the ray set R, which allows to reason about O being between A and B. Although there can be an infinite number of rays in R, there will be a finite number of distinct topological relations, modelled with the 9IM (Table 5.1), between individual rays r_i and O. For example, in Figure 5.2 both extreme rays will form the same 9IM with O (i.e.,

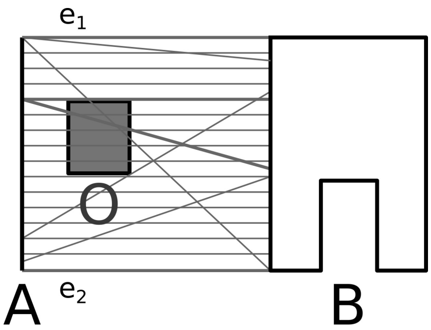


FIGURE 5.2: Some of the rays between A and B, and the two extreme rays e_1 and e_2.

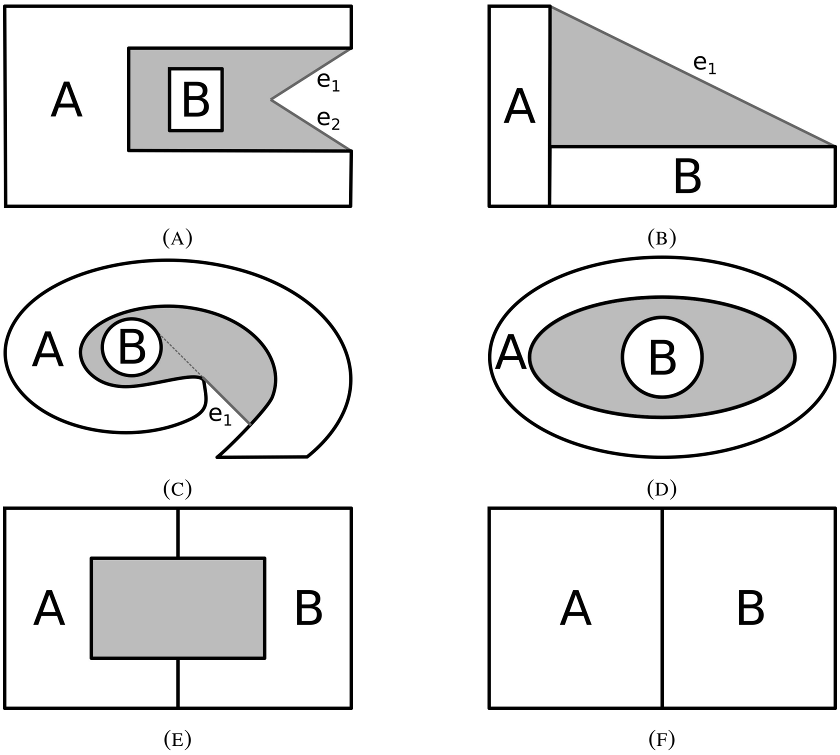


FIGURE 5.3: Cases where there are (a) two intersecting extreme rays, (b) and (c) only one extreme ray, (d) and (e) no extreme rays, and (f) no rays between A and B. The R_A between A and B is shown in gray.

disjoint). Other rays are either also disjoint from O, touching O (i.e., second bold ray from the top in Figure 5.2), or crossing O (i.e., third bold ray from the top in Figure 5.2). These distinct topological relations can be collated into a single model representing the relationship between R and O.

Let a and b be two distinct point-sets from Table 5.1. For the proposed method, the empty and non-empty intersections between a and b will be numerically represented in Table 5.1 as 0 for False, and 1 for True.

TABLE 5.1: The 9IM topological relation between ray r_i and the object O.

	O°	∂O	O^-
r_i°	$r_i^\circ \cap O^\circ$	$r_i^\circ \cap \partial O$	$r_i^\circ \cap O^-$
∂r_i	$\partial r_i \cap O^\circ$	$\partial r_i \cap \partial O$	$\partial r_i \cap O^-$
r_i^-	$r_i^- \cap O^\circ$	$r_i^- \cap \partial O$	$r_i^- \cap O^-$

$$a \cap b = \begin{cases} \emptyset & \rightarrow 0 \\ \neg\emptyset & \rightarrow 1 \end{cases}$$

RIM combines the distinct 9IM topological relations between the ray set R and object O. However, since there will be a finite number of distinct topological relations between R and O, there will be a finite number of the distinct 9IM topological relations to combine (i.e., i in Table 5.1 is finite). In case when the same topological relation exists for an extreme ray, and a non-extreme ray, RIM will not be affected if that topological relation is disregarded for a non-extreme ray since it will already be present in calculations through the extreme ray. Thus, to reduce the number of distinct rays and simplify the calculation of RIM, the topological relations between each of the existing extreme rays and the object O are calculated first. The topological relations of the other, non-extreme rays and O will be taken into consideration only if they are different from the topological relations between the extreme rays and O. To accommodate all this information, the RIM between R and O can be arranged as the 4×3 RIM matrix shown in Table 5.2.

The first two rows of the RIM matrix describe the combined relations between the interiors and boundaries of all the rays (i.e., both extreme and non-extreme rays) and the interior, boundary, and exterior of O:

$$\mathrm{RIM}(R,O)_{[1-2,1-3]} = \frac{\sum_{i=1}^{n} e_i^{(\circ|\partial)} \cap O^{(\circ|\partial|-)} + \sum_{i=1}^{m} r_i^{(\circ|\partial)} \cap O^{(\circ|\partial|-)}}{N} = \begin{cases} 0 & (none) \rightarrow \square \\ 1 & (all) \rightarrow \blacksquare \\ otherwise & (some) \rightarrow ◪ \end{cases}$$

TABLE 5.2: The RIM between the ray set R, and object O.

	O°	∂O	O^-
$r_i^\circ \wedge e_i^\circ$	$\frac{\sum_{i=1}^{n} e_i^\circ \cap O^\circ + \sum_{i=1}^{m} r_i^\circ \cap O^\circ}{N}$	$\frac{\sum_{i=1}^{n} e_i^\circ \cap \partial O + \sum_{i=1}^{m} r_i^\circ \cap \partial O}{N}$	$\frac{\sum_{i=1}^{n} e_i^\circ \cap O^- + \sum_{i=1}^{m} r_i^\circ \cap O^-}{N}$
$\partial r_i \wedge \partial e_i$	$\frac{\sum_{i=1}^{n} \partial e_i \cap O^\circ + \sum_{i=1}^{m} \partial r_i \cap O^\circ}{N}$	$\frac{\sum_{i=1}^{n} \partial e_i \cap \partial O + \sum_{i=1}^{m} \partial r_i \cap \partial O}{N}$	$\frac{\sum_{i=1}^{n} \partial e_i \cap O^- + \sum_{i=1}^{m} \partial r_i \cap O^-}{N}$
$e_{1,2}^\circ$	$\frac{\sum_{i=1}^{n} e_i^\circ \cap O^\circ}{n}$	$\frac{\sum_{i=1}^{n} e_i^\circ \cap \partial O}{n}$	$\frac{\sum_{i=1}^{n} e_i^\circ \cap O^-}{n}$
$\partial e_{1,2}$	$\frac{\sum_{i=1}^{n} \partial e_i \cap O^\circ}{n}$	$\frac{\sum_{i=1}^{n} \partial e_i \cap \partial O}{n}$	$\frac{\sum_{i=1}^{n} \partial e_i \cap O^-}{n}$

where the first number in brackets specifies the row number of the RIM matrix (e.g., 1-2 means rows 1, and 2), the second number in brackets specifies the column number of the RIM matrix (e.g., 1-3 means columns 1, 2, and 3), $n = |E|$ is the number of extreme rays, m is the number of rays with distinct 9IM intersections other than extreme rays, $N = n + m$ is a total number of distinct kinds of rays including the extreme rays, and "|" symbolises a logical *or* operation. The results of this expression can be categorised into three groups: 0, 1, and all other values between 0 and 1, which have the following meaning. The first group collects cases where the interiors/boundaries of *none* of the rays are intersecting the interior/boundary of the object O. The second group collects cases where *all*, and the third group where *some* but not *all* of the rays are intersecting the interior/boundary of O. Each of these groups also has a corresponding graphical representation as either white (□), part white part black (◩), and black square (■).

The last two rows of the RIM matrix describe the combined relations between the interiors and boundaries of only the extreme rays and the interior, boundary and exterior of O:

$$\mathrm{RIM}(R, O)_{[3-4,1-3]} = \frac{\sum_{i=1}^{n} e_i^{(\circ|\partial)} \cap O^{(\circ|\partial|-)}}{n} = \begin{cases} 0 & (none) \rightarrow \square \\ 1 & (all) \rightarrow \blacksquare \\ otherwise & (some) \rightarrow ◩ \end{cases}$$

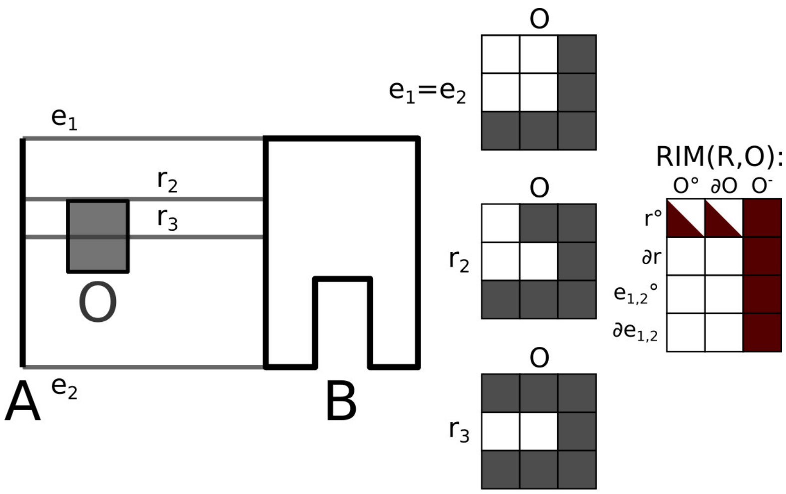


FIGURE 5.4: Spatial objects *A*, *B*, and *O* (Figure 5.2), distinct rays between *A* and *B*, 9IM between distinct rays and *O*, and RIM between the ray set *R* and *O*.

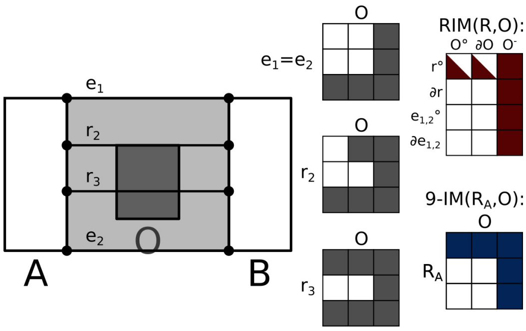
To illustrate RIM in application, Figure 5.4 shows RIM for peripheral objects *A* and *B*, and core object *O* from Figure 5.2. The three illustrations of matrices on the left side show the distinct 9IM topological relations between individual rays and *O*, while the matrix showing the RIM between *R* and *O* is shown on the right. For cases where there are no extreme rays or even no rays at all between *A* and *B* (e.g., such as in Figure 5.3), this can be described with rows where all three values are □. For example, in scenarios where no extreme rays exist (Figures 5.3d and 5.3e), all values in the bottom two rows of the RIM matrix would be □. Similarly, in a special case where no rays exist between *A* and *B* (Figure 5.3f), all fields in the RIM matrix would be □.

5.2 Relations between three polygon objects

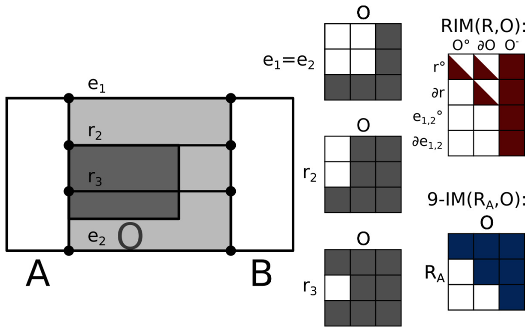
This section demonstrates the expressiveness of RIM, and its ability to distinguish different configurations of triplets of polygon objects in a 2D plane. Although the RIM can be applied to all geometry types (i.e., points, lines, and polygons), the cases demonstrated in this section are limited to closed single-part convex polygons where peripheral objects are disjoint and the position and size of the polygon core object are the only changing variables. The reason for doing so is that showing all the possible combinations of different geometry types in core and peripheral objects would demand much more space. For an example of RIM with a peripheral line object see Figure 5.4, and for examples of RIM where peripheral objects are touching each other see some examples from the case study in Section 7.1. A comprehensive discussion on the number of all the possible RIMs between closed single-part polygons is shown in Section 5.3.

Figure 5.5 shows 23 triplet configurations, all of which are differentiated by RIM (i.e., no two configurations have the same RIM representation). Each configuration is showing peripheral objects A (white) and B (white), core object O (dark gray), the ray area R_A (light gray), extreme rays (denoted e_i), and all other distinct rays that occur (denoted r_i). Note that rays do not have to be parallel to each other, nor perpendicular to the peripheral objects. They are shown in that way to preserve the simplicity of visualisation while demonstrating all distinct types of rays expressible with 9IM in the given ray areas. The same applies to the polygon objects in the figures, which do not have to be rectangular nor completely aligned for the demonstrated RIM representations to be valid. On the right side of each configuration are the matrices showing the 9IM topological relations between extreme rays and O (grey), the 9IM between non-extreme rays and O (grey), the RIM between all of the rays and O (red), and the 9IM between the R_A and O (blue). The 9IM between the ray area and O is shown as a reference to which the expressiveness of RIM can be compared to.

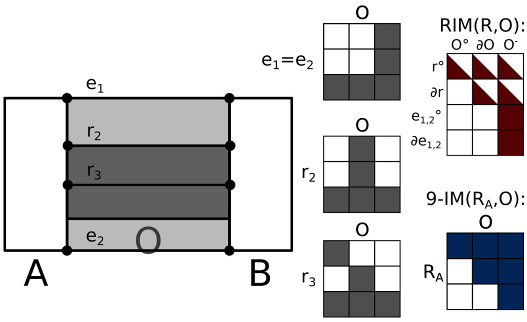
Examples shown in Figure 5.5 demonstrate that RIM is much more expressive, and can differentiate more ternary configurations, i.e., 23 configurations compared to the 9IM with eight distinctions. These refined distinctions may still be important for nuanced qualitative reasoning. For example, if O was a barrier (e.g., a wall), to analyse if A can be seen from B and vice versa it is necessary to make distinctions between RIM4 and RIM7.



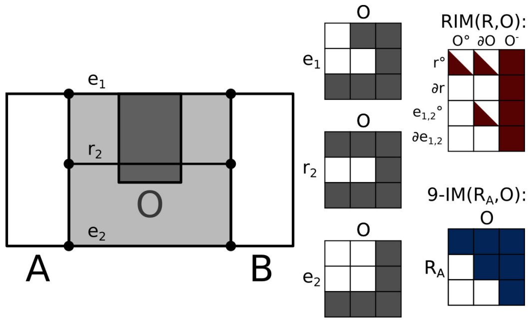
(RIM1)



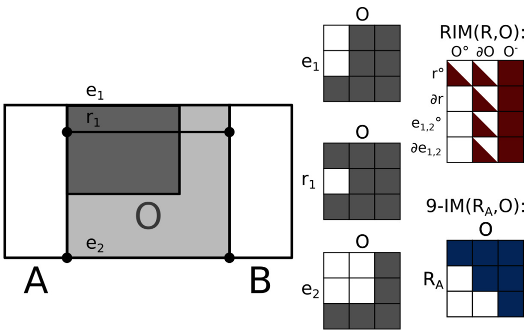
(RIM2)



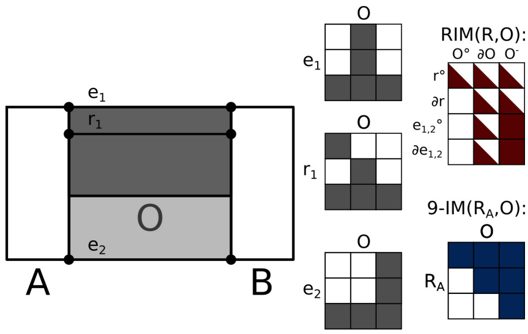
(RIM3)



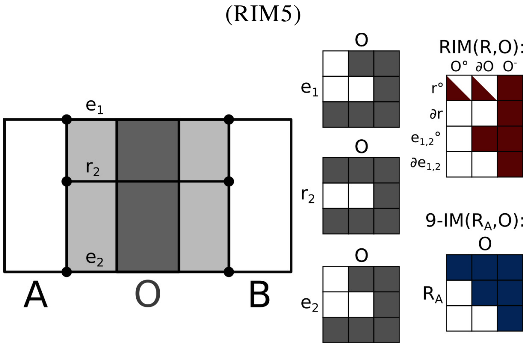
(RIM4)



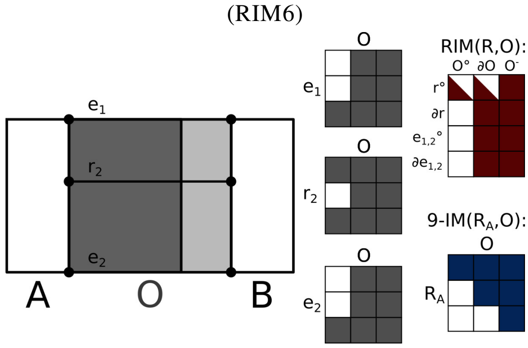
(RIM5)



(RIM6)

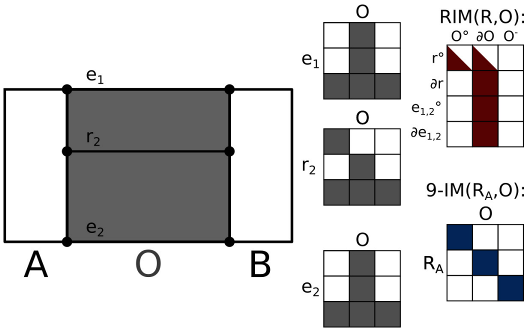


(RIM7)

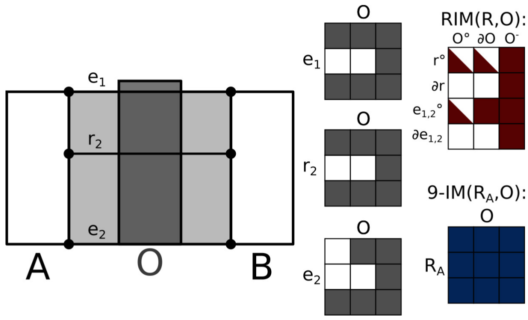


(RIM8)

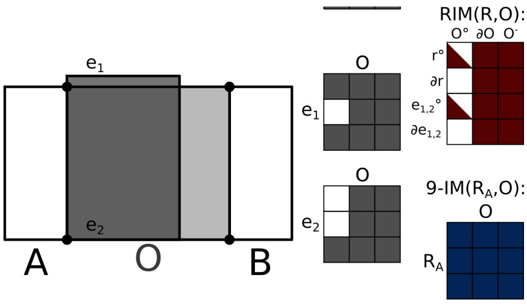
continued



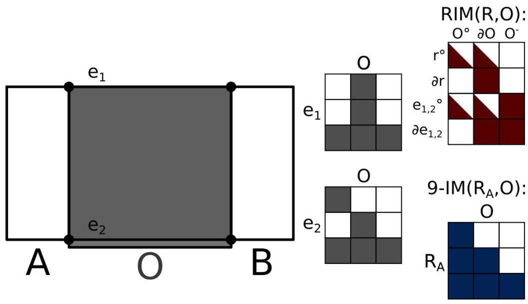
(RIM9)



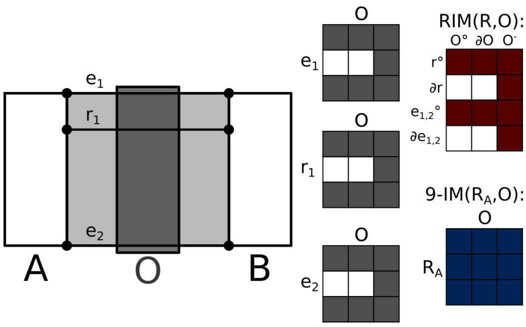
(RIM10)



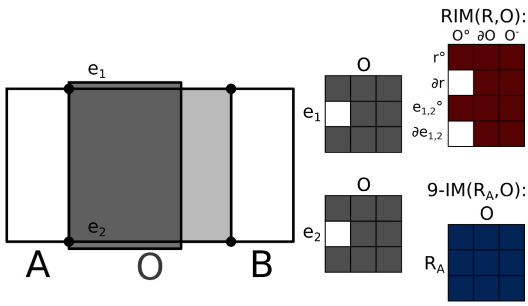
(RIM11)



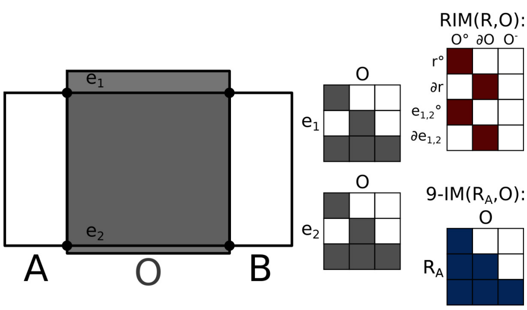
(RIM12)



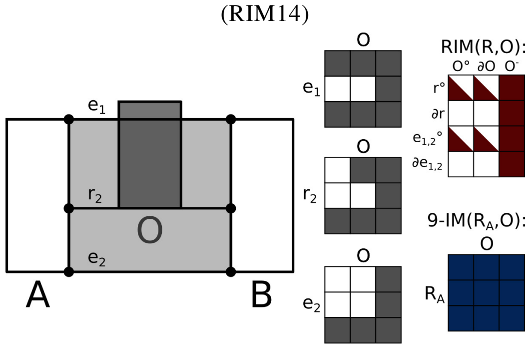
(RIM13)



(RIM14)



(RIM15)



(RIM16)

continued

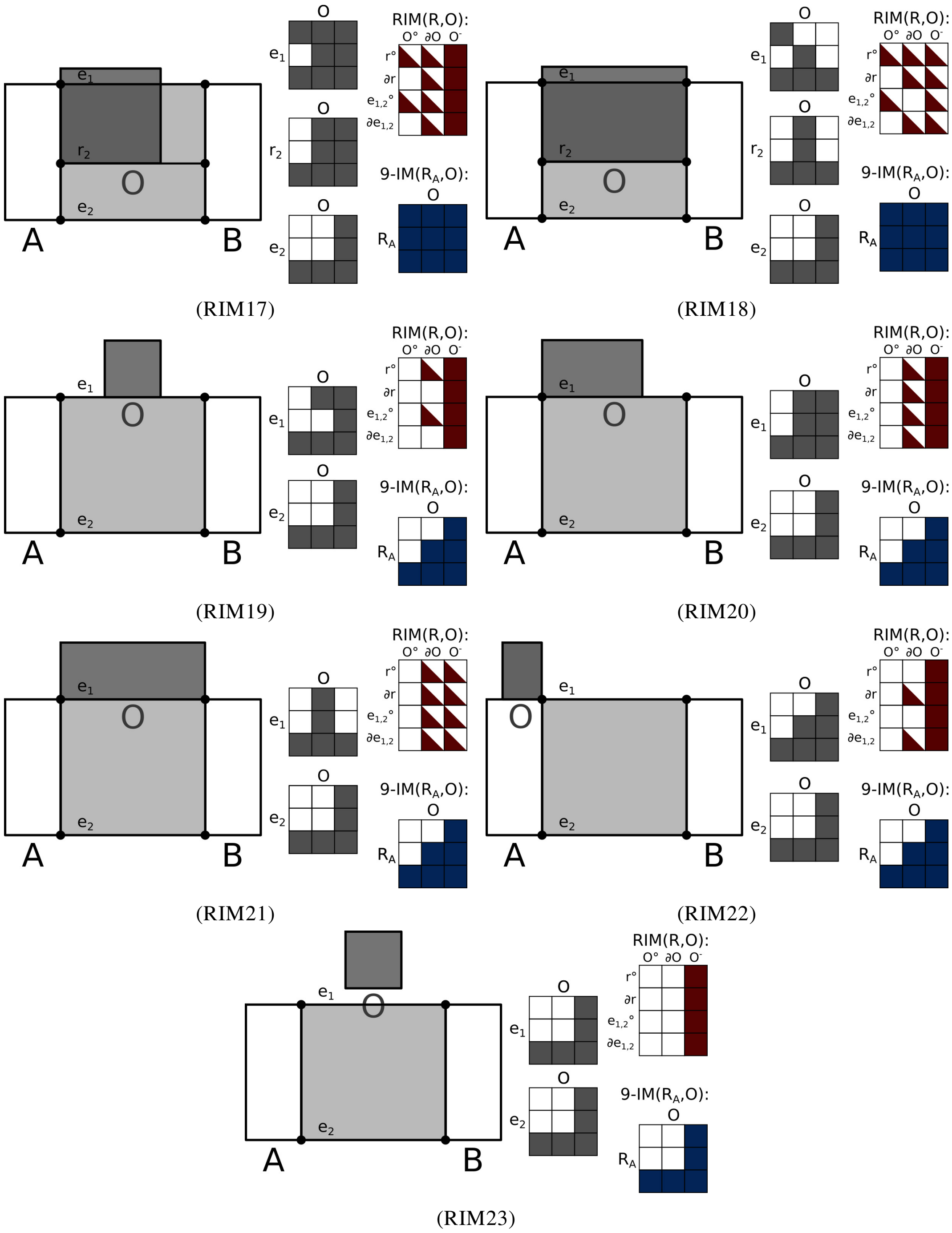


FIGURE 5.5: Twenty-three distinct RIMs between three polygon objects (A, B, and O), and the corresponding 9IM relations between the ray area (R_A) and the object O.

5.3 Number of theoretically possible distinct RIM cases

A 12 field matrix where each field can have one of three values (□, ◩, or ■) – has a total of 3^{12} distinct RIM combinations. However, certain correlations exist in the RIM matrix, which restricts the number of distinct RIM combinations that may occur. For example, if the value in the field $\text{RIM}_{[1,1]}$ says that interiors of all rays intersect the interior of the object (i.e., ■), this means that interiors of all extreme rays in the field $\text{RIM}_{[3,1]}$ (i.e., which are a subset of all rays) must also have the intersections with the interior of the core object. Similarly, when none of the rays' interiors has an intersection with the interior of the core object (i.e., value in $\text{RIM}_{[1,1]}$ is □), this implies that none of the extreme rays' interiors has the intersection with the interior of the core object (i.e., value in $\text{RIM}_{[1,1]}$ will be □). Only when some, but not all of the rays' interiors intersect the interior of the core object (i.e., value in $\text{RIM}_{[1,1]}$ is ◩), no implication for the intersections of the extreme rays' interiors with the interior of the core object can be made, thus all three values are possible for the field $\text{RIM}_{[3,1]}$ (Table 5.3). In other words, the fields from the top two rows of the RIM matrix implicitly determine the possible values of respective fields in the bottom two rows.

TABLE 5.3: Values in the top two rows of the RIM matrix (second column) determine the possible values in the respective fields in the bottom two rows of the RIM matrix (third column). This results in 5 possible outcomes for each pair of ($\text{RIM}_{[i,j],i\in\{1,2\}}$, $\text{RIM}_{[i+2,j]}$) fields.

$\text{RIM}_{[i,j],i\in\{1,2\}}$		$\text{RIM}_{[i+2,j]}$	pair outcome #
□	→	□	1
◩	→	□	2
◩	→	◩	3
◩	→	■	4
■	→	■	5

The above analysis means that the RIM matrix can be seen as a combination of 6 pairs of correlated fields, where each pair can have one of 5 different combinations of values (Table 5.3). Thus, the total number of possible RIM cases can be calculated as:

$$N_{\text{RIMs}} = 5^6 = 15{,}625$$

Condition 1: non-overlapping polygons. Furthermore, since this study only considers non-overlapping polygons, an intersection between the rays' boundaries and the object's interior cannot exist. This means that the fields $\text{RIM}_{[2,1]}$ and consequently $\text{RIM}_{[4,1]}$ will always have values □, □. Given that one of 6 pairs of fields in RIM now has fixed values, the total number of possible RIM cases can be calculated as:

$$N_{\text{RIMs}} \mid \mathit{Condition1} = 5^5 = 3{,}125$$

Condition 2: all rays must exist in the interior, boundary, or exterior of O. Because the rays between peripheral objects have to exist somewhere in space relative to the core object O, the sum of values in each of the first two rows of the RIM matrix has to be ≥ ■. If the sum of the values in the row is □, this means that the rays exist neither in the interior, nor in the boundary, nor the exterior of O. If the sum of the values is ◩, this means that only some rays exist either in the interior, or in the boundary, or the exterior of O, while the rest of the rays are not accounted for. This means that there are four combinations of values that are not allowed to occur in any of the first two rows of the RIM matrix: {□, □, □}, {◩, □, □}, {□, ◩, □}, {□, □, ◩}. For the first combination, there is only one invalid option given the pair outcomes in Table 5.3. For each of the other three combinations, there are three possible outcomes in the bottom half of the RIM matrix. This means that there are a total of $1 + 3 \cdot 3 = 10$ impossible combinations. As a result, for three pairs of correlated values, there are $5^3 - 10$ possible combinations.

Since the first row is independent of the second row, the same argument can be applied. However, the first value of the second row will always be □ due to the objects being non-overlapping polygons. Because of this constraint, the combination {◩, □, □} cannot appear in the second row. Thus, the first combination listed above will yield one invalid option, and the third and fourth combinations will each have three possible outcomes in the bottom half of the RIM matrix according to the pair outcomes in Table 5.3, which in total results in $1 + 2 \cdot 3 = 7$ impossible combinations. This means that for the pairs of correlated values in the second and the fourth row of the RIM matrix there are $(5^2 - 7)$ possibilities. Finally, when all the conditions are applied the number of possible RIM cases will be:

$$N_{\text{RIMs}} \mid \mathit{Condition1} \wedge \mathit{Condition2} = (5^3 - 10) \cdot (5^2 - 7) = 2{,}070$$

5.4 Implementation of RIM

The implementation of RIM departs from the formal model presented in Section 5.1. There exist an infinite number of rays between two spatial objects, all of which need to be considered in RIM, thus the computational implementation is impractical. Instead, the implementation of RIM relies on the R_A between A and B, and then on testing if the necessary conditions for a specific ray to exist between A and B are satisfied. This approach is equivalent for any pair of A and B that are simple, single-part geometries as it creates a maximum convex shape. The complete Python code implementation of RIM is available online as a Python3 package (https://pypi.org/project/rim/).

5.4.1 Algorithm for calculating the ray area

Algorithm 3 and Figure 5.6 show the computation of the ray area between two non-overlapping polygons A and B. This ray area will later be used to find out if specific kinds of rays exist between A and B. The computation starts with the *find_main_area(A, B)* function (Algorithm 3 line 21). This function first creates a convex hull of A and B (Figure 5.6b), and takes the difference of the convex hull and A and B in order to get the area between A and B (Algorithm 3 line 2, and Figure 5.6c). From this newly created area which might be a multigeometry (e.g., more than one polygon in a single geometry), only the parts which are touching both objects are kept to remove any island polygons (Algorithm 3 lines 3-5, and Figure 5.6d).

Next, the points that cannot be reached by any rays from the opposite object are found with the *find_unreached_points(A, B, main_area)* function (Algorithm 3 line 22). The function first divides the boundary points of the area between A and B (i.e., main_area) based on which of the two starting objects (i.e., A and B) they are touching (Algorithm 3 lines 8-9, and Figure 5.6e). Then the lines (or rays) are created between all the combinations of the points on the opposite sides of the area (Algorithm 3 lines 10-15, and Figure 5.6f), to find out if any points cannot be reached by any rays from the opposite object (Algorithm 3 lines 16-19, and Figure 5.6g).

Then, by removing unreachable points from A and B, the parts of the area that cannot be reached by any rays will be removed, but so will some parts of the area between A and

Algorithm 3 Calculating the ray area between two spatial objects

Require: Geometries of two spatial objects (A and B) between which the ray area is to be calculated

```
function FIND_MAIN_AREA(A, B)
    main_area ← (convex_hull(A,B).Difference(A)).Difference(B)
    for geom in main_area do
        if geom.Disjoint(A) or geom.Disjoint(B) then
            main_area.RemoveGeometry(geom)
    Return main_area

function FIND_UNREACHED_POINTS(A, B, main_area)
    points_A ← (main_area.Intersection(A)).GetPoints()
    points_B ← (main_area.Intersection(B)).GetPoints()
    lines_not_crossing_AB ← MakeMultiLine()
    for p1 in points_A do
        for p2 in points_B do
            line ← MakeLine(p1,p2)
            if line.Touches(A) and line.Touches(B) then
                lines_not_crossing_AB.AddLine(line)
    unreached_points ← MakeMultiPoint()
    for point in points_A.Union(points_B) do
        if point.Disjoint(lines_not_crossing_AB) then
            unreached_points.AddPoint(point)
    Return unreached_points

function CALCULATE_RAY_AREA(A, B)
    main_area ← FIND_MAIN_AREA(A, B)
    unreached_points ← FIND_UNREACHED_POINTS(A, B, main_area)
    main_area_points_removed ← main_area.RemovePoints(unreached_points)
    polygons_remaining ← main_area.Difference(main_area_points_removed)
    cumulative_areas_to_add ← MakePolygon()
    for polygon in polygons_remaining do
        area_to_add ← MakePolygon()
        for line in lines_not_crossing_AB do
            if line.Intersects(polygon) then
                extended_line ← line.Extend()
                new_boundaries ← extended_line.Union(polygon_remaining.Boundary())
                polygon_splits ← new_boundaries.Polygonise()
                for geom in polygon_splits do
                    if geom.Disjoint(unreached_points) then
                        area_to_add ← area_to_add.Union(geom)
        cumulative_areas_to_add ← cumulative_areas_to_add.Union(area_to_add)
    ray_area ← main_area_points_removed.Union(areas_to_add)
    Return ray_area
```

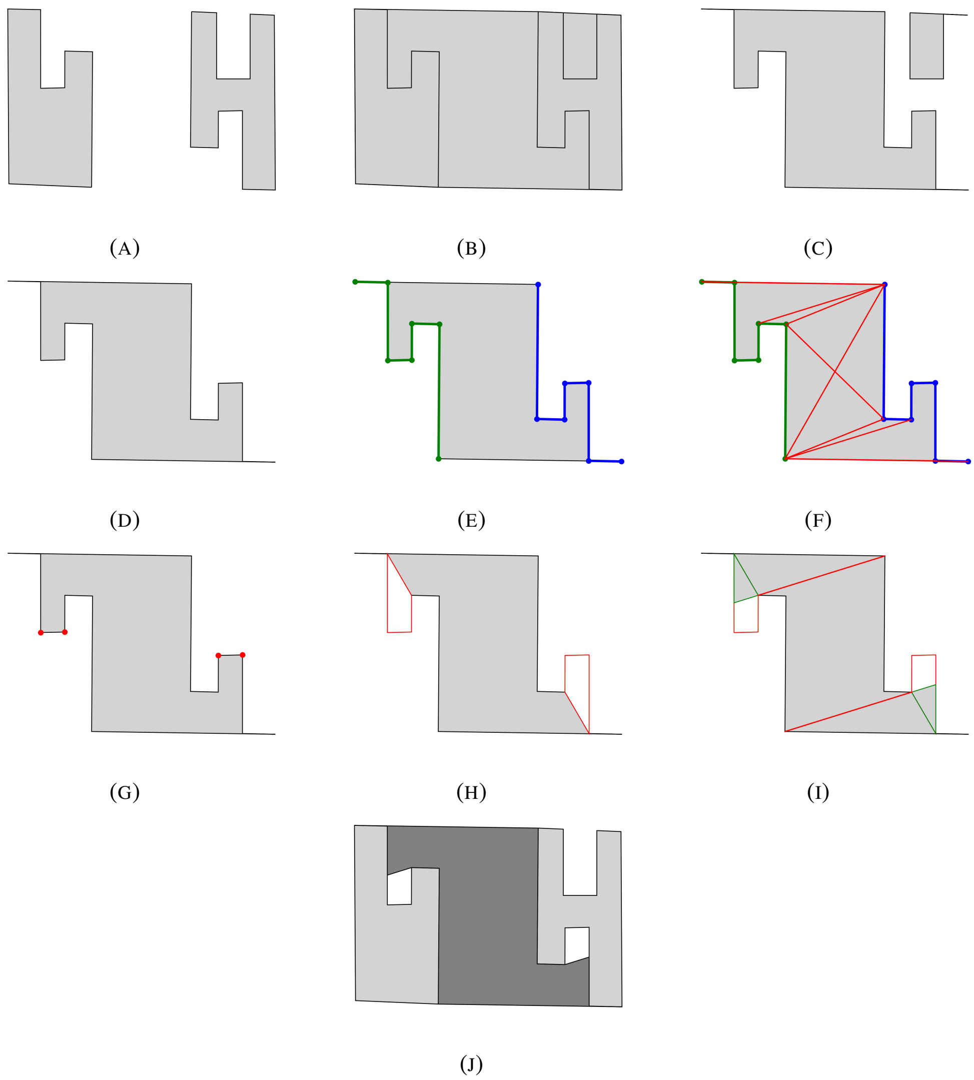


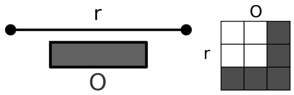
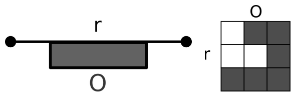
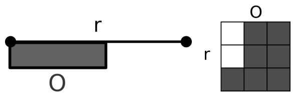
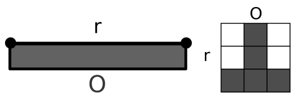
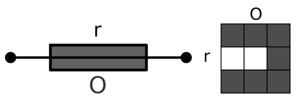
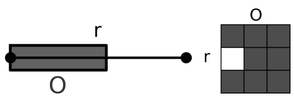
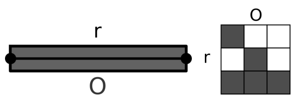
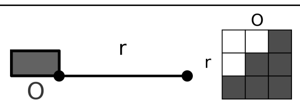
FIGURE 5.6: Calculating the ray area between two spatial objects.

B which can be reached by rays (Algorithm 3 line 23, and Figure 5.6h). These reachable areas are then found by extending the previously calculated rays to split the areas removed in Figure 5.6h, and then by accepting those parts of the created splits which are not touching the unreachable points (Algorithm 3 lines 24-37, and Figure 5.6i). Finally, when these parts are added, the complete ray area between A and B is calculated (Algorithm 3 line 38, and Figure 5.6j).

5.4.2 Finding the distinct kinds of rays

After the ray area between A and B is computed, the geometries that are available in the RIM computation are objects A, B, and O, the ray area between A and B, and the extreme

TABLE 5.4: Eight distinct rays that can be found in Figure 5.5, and the necessary conditions for them to exist as non-extreme and extreme rays.

Case	Conditions for ray to exist	Conditions for extreme ray (e) to exist
1) 	O disjoint R_A OR (O.difference(R_A) intersects A AND O.difference(R_A) intersects B)	O disjoint e
2) 	O.intersects(R_A) AND O.difference(R_A) intersects A AND O.difference(R_A) intersects B	e touches O AND boundary(e) disjoint O
3) 	(O.intersection(R_A) intersects A AND R_A.difference(O) intersects A) OR (O.intersection(R_A) intersects B AND R_A.difference(O) intersects B)	e touches O AND (startpoint(e) touches O XOR endpoint(e) touches O)
4) 	O.intersection(R_A) intersects A AND O.intersection(R_A) intersects B	e within boundary(O)
5) 	O within R_A OR O overlaps R_A	e crosses O AND boundary(e) disjoint O
6) 	O.intersection(R_A) intersects A XOR O.intersection(R_A) intersects B	e crosses O AND (startpoint(e) touches O XOR endpoint(e) touches O)
7) 	O equals R_A OR O contains R_A OR (O.intersection(R_A) intersects A AND O.intersection(R_A) intersects B)	e within O
8) 	–	e touches O

rays between A and B which can be retrieved as a difference of the ray area's boundary and A and B. The remaining challenge is to find all the distinct rays that exist in the given scenario. This is achieved by extracting all the distinct rays from all the scenarios presented in Figure 5.5, and then by devising a set of necessary conditions which must be satisfied for a specific ray to exist. There are eight distinct rays in this case, and they are presented together with their corresponding conditions in Table 5.4. The first seven rays may exist in the given scenario as a non-extreme ray and/or as an extreme ray, while the eighth ray can only exist as a non-extreme ray. This is why both sets of conditions need to be checked separately. The difference in these conditions is mostly that conditions for extreme rays can utilise the geometry of the extreme ray itself, as opposed to the non-extreme rays where the geometry of the ray is not available. After the existence of each of the eight distinct rays and extreme rays has been tested, and it is known which rays and extreme rays exist in a given scenario, they are combined into the matrix representation of RIM using the formulas presented in Section 5.1.

5.5 Conclusion

This chapter has presented a novel model for ternary spatial relations – RIM, for which the formal definition has been presented. The expressiveness of RIM was demonstrated across twenty-three different configurations of spatial objects. When compared to the 9IM based alternative for describing these scenarios (i.e., 9IM between the ray area and the core object), RIM was demonstrated to be much more descriptive. This study does not give a complete list of all the different scenarios distinguishable by RIM. However, its expressiveness was investigated by calculating the number of all different theoretically possible RIM scenarios.

This chapter has also presented the implementation of RIM. This was not a straightforward conversion of algorithm to code, but a pragmatic and computationally effective adaptation of the concept. Because there may be an infinite number of rays in RIM, numerous logical and topological tests were necessary to conclude which rays exist in a given scenario and which rays cannot possibly exist.

Finally, RIM can be applied to study the betweenness of spatial objects and other ternary relationships. Case studies that apply RIM for the analysis of betweenness of objects and for the analysis of ternary visibility relations are presented in Chapter 7. According to the study of betweenness in [9], the most accurate representation of the space between the two spatial objects is based on visibility and does not include the areas that cannot be seen from both objects (i.e., concavities of concave objects). Although theoretically RIM does not define an area, the rays from RIM implicitly represent the area between the peripheral objects since each ray can only touch each peripheral object once.

6

Error detection approaches

Chapters 4 and 5 have introduced two novel models for ternary spatial relations, namely RTL and RIM. This chapter presents two approaches for detection of errors in map databases that are utilising the RTL and RIM models. Such approaches are needed because manual detection of errors is not feasible for large spatial databases used today and the existing rule-based approaches are often limited and avoid complex spatial analyses (e.g., analyses looking at multiple spatial objects simultaneously). This chapter first introduces the notion of mature datasets and the use of outlier detection for their intrinsic assessment. It then introduces an approach for discovery of topological constraints automatically from data. The discovered constraints can be applied to all objects in the database to label objects that do not satisfy them as erroneous. The second approach uses RIM to analyse if the building can be accessed from its closest road, or if the access is completely obscured by other buildings that stand in the way. If a building is not accessible, this often indicates the existence of nearby errors in map data (e.g., the access road might be missing from the map). Both approaches will be used for error detection in case studies in Chapter 7.

6.1 Mature datasets and outlier detection

Data-driven error detection approaches in this thesis are based on the assumption that in some spatial datasets, the majority of data is of acceptable quality. This assumption is motivated by the fact that the widely used map databases are usually subject to different quality control processes and that errors are more likely rare than common. For example, OSM has a strong community of contributors and a long lineage of edits where contributors are encouraged to correct errors and update the database. On the other hand, government-managed map databases usually have to adhere to strict protocols that assure high quality of data. In this thesis, datasets for which the majority of data is assumed to be of acceptable quality are called *mature datasets*.

If the majority of data in mature datasets is considered a norm, then the deviations from that norm can be seen as anomalies or outliers [18]. Thus, outlier detection can be applied to detect potential errors in data [19]. For example, if the majority of objects of a class frequently form the same topological relation with other objects, this pattern can be learned and used to classify further spatial objects. If the majority of roads in the dataset are connected to other line objects (e.g., other roads), then this pattern can be adopted as a rule. Thus, if some roads are not connected to any other linear objects, they would be seen as potentially erroneous and should be inspected.

In this thesis, it is assumed that there is one dominant pattern of topological relations for a given object class. Nonetheless, there may be cases where the object class is a generalisation of multiple sub-classes with highly distinct patterns. In such cases, defining the entire object class based on only one pattern may be too general. However, this non-trivial problem of detecting aggregate classes of objects (e.g., *transport infrastructure*) based on detected patterns is not in the scope of this thesis.

This chapter is based on published papers [1, 3, 4].

TABLE 6.1: Variants of contingency tables showing error detection with ground truth and intrinsic error detection with a model. True positives (TP) are objects that are correctly classified as *class* in the dataset, false positives (FP) are objects that are incorrectly classified as *class* in the dataset. False negatives (FN) are objects that have been omitted, and true negatives (TN) are objects that have been correctly classified as *not class*. Objects prefixed with *False* (i.e., FP and FN) are detected as potential errors.

		Ground truth	
		class	*not class*
Dataset	*class*	TP	FP
	not class	FN	TN

(A) Dataset vs Ground truth

		Model	
		class	*not class*
Dataset	*class*	TP	FP
	not class	FN	TN

(B) Dataset vs Model

6.2 Intrinsic error detection

The approaches proposed in this chapter differ from the usual quality assessment approaches that use the ground truth to validate data [85–87], and follow the trend of more recent studies that propose intrinsic approaches which do not require the ground truth [12, 94]. This can be achieved either by learning the patterns that occur in the majority of the data or by assuming those patterns heuristically. In this thesis, the pattern that is informed by the majority of objects of the studied class is called the *model*.

As the model is learned based on all labelled instances of a class, it operationalises an extensional definition of an object class [5]. This can also be seen as a data-driven or bottom-up approach [103], which differs from intensional definitions that are formalised before object creation and classify objects based on their alignment with the definition (e.g., definitions from Table 6.3).

Once created, the model can be used to validate the existing objects in the dataset, or to classify new data records. The aim is to find objects in the specified class that differ from the model that represents the majority of objects in the same class. Table 6.1a shows a typical contingency table when the errors in data are found by comparing the dataset against the ground truth. Columns in the contingency table show how features should be classified, while the rows show how the corresponding data are classified in the dataset. On the other hand, the approaches proposed in this chapter compare the existing dataset to the model, as shown in Table 6.1b.

TABLE 6.2: Variants of contingency tables for evaluation of intrinsic approaches. If manual classification is not available, then the dataset can be used as a reference to evaluate a model. Otherwise, the model-based classification can be compared against the manual classification. TP and TN represent objects that were classified correctly by the model, while the FP and FN represent objects where the model classification was incorrect.

		Dataset	
		class	*not class*
Model	*class*	TP	FP
	not class	FN	TN

(A) Model vs Dataset

		Manual classification	
		class	*not class*
Model	*class*	TP	FP
	not class	FN	TN

(B) Model vs Manual classification

There are two types of errors that the proposed approaches can detect in the classification of objects. The first type is when an object is classified into the given class in the data, but the model classifies it as not a part of that class. These errors correspond to *FP* in Table 6.1b. The other type is when objects should belong to the given class in the data, but they do not. These are errors by omission, and correspond to *FN* in Table 6.1b.

6.3 Evaluating error detection

After the models have been created and used to detect errors in data, their performance needs to be evaluated. Since the ground truth data are not available and cannot be used as a reference, different approaches need to be applied here. One approach is to use the same dataset that was used for learning the model and detection of errors as a reference dataset (Table 6.2a). It is expected that this dataset will contain some errors, but we have already assumed that the majority of the data are of acceptable quality. The other approach is to manually classify the whole or apart of the dataset and use these results as a reference for model evaluation (Table 6.2b). This will also grant cognitive plausibility to the model since it will be compared directly against human judgement.

For both evaluation approaches, the performance of of the model can be described with the following quality measures that are calculated from the contingency tables (Tables 6.2a and 6.2b):

- Precision: the fraction of objects in the dataset classified as *class* by the proposed approach (model) and also classified as *class* in the dataset.

$$precision = \frac{TP}{TP + FP}$$

- Recall: the fraction of objects in the dataset that are classified as *class* in the dataset and were also classified as *class* by the proposed approach.

$$recall = \frac{TP}{TP + FN}$$

- Accuracy: the ratio of all the objects that were correctly classified as either *class* or *not class* to all the incorrectly classified objects. It captures how the approach (classification) compares to the existing classification in the dataset.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Specificity: the ratio of all objects that were correctly classified as *not class* by the proposed method, and all objects that were classified as *not class* in the dataset. It captures how many objects that are not classified as *class* in the dataset have been correctly classified as *not class* by the proposed method.

$$specificity = \frac{TN}{TN + FP}$$

- F1 score (less commonly called *Sørensen-Dice index*): the harmonic mean of precision and recall. This measure is commonly used for evaluation of machine learning classification tasks (for a review, see [104]).

$$F1score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

6.4 Discovery of topological constraints for classes of spatial objects

The first error detection approach in this chapter is based on a methodology that mines topological relations for a given object class using the frequent itemset mining method [105] to learn the most frequent topological relations that the objects of that class have with other objects in the dataset. Figure 6.1 shows the steps of the proposed method.

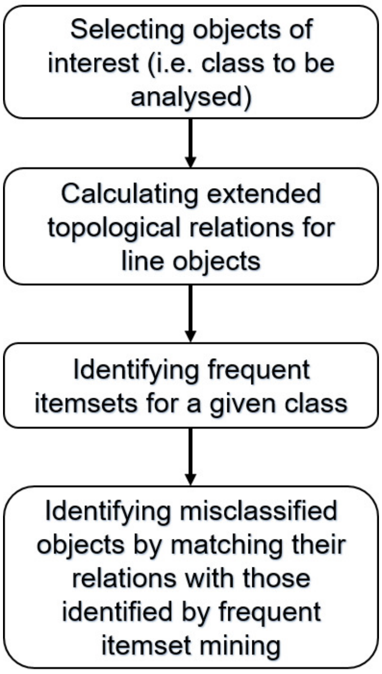


FIGURE 6.1: Steps of the proposed method.

The first step in this approach is to select objects of interest from a spatial dataset – these will represent the set of core objects. For example, in the case study presented in Chapter 7, all the instances of the class {bridge} in a spatial dataset are selected as core objects.

After selecting objects of interest, the topological relations between these core objects and all other peripheral objects in the dataset are computed using the RTL model that was introduced in Chapter 4. In this approach, only the peripheral objects that are not disjoint from the core object have been considered. Since these topological relations are the input of the frequent itemset mining process, this step can be seen as feature engineering.

Finally, for each of the selected core objects, their topological relations with peripheral objects are analysed to extract the most frequently occurring topological relations. This enables automated inference of principal topological constraints for the given class of spatial objects.

6.4.1 Topology and classes of spatial objects

For many classes of spatial objects, the topology may define their function. For example, a road segment should be connected to other road segments, so that the road network in the dataset remains routable. However, in some specific situations, the topological properties of objects may be violated. Considering the road example, if a road is disconnected from the network for some reason such as on-going construction or a disaster that has affected the surrounding roads, it does not change the fact that it is still a road. On the other hand, the question of whether such road is still fully functional may be open for discussion.

Most current approaches for spatial data quality assurance manually define which topological relations an object of a given class should have with other objects of other classes to be considered valid. These topological constraints are usually defined per object class [106], constrained to binary relationships, and captured in validation rules. Tarquini and Clementini [107] have also shown how multiple spatial relation constraints can be combined to define sub-types of an object class (i.e., rivers). An example of a topological constraint for the road class is *"a road should be connected to at least one other road"*. All the roads disconnected from the network will then be classified as invalid due to the violation of this topological constraint.

Most spatial databases currently define object classes through manual attribute annotation. However, there has been work on utilising the semantic similarity measurements to provide annotators with recommendations, and thus reduce the semantic heterogeneity in the data [108]. Past work has also investigated how object classification is dependent on a set of attributes of the objects [28, 109].

Table 6.3 shows three different definitions of bridges. These definitions are based on a combination of functional properties of objects strongly dependent on the satisfaction of topological relationships (e.g., bridge *spans* obstacles and *carries* a road). Such verbal definitions may be incomplete and insufficiently specific, thus interpreted differently by different operators during the manual object classification. For example, the first two definitions both

TABLE 6.3: Alternative definitions of a *bridge*

Source	**Definition of a bridge**
Merriam-Webster	*"Bridge is a structure carrying a pathway or roadway over a depression or obstacle (such as a river)."*
Wikipedia	*"Bridge is a structure built to span physical obstacles without closing the way underneath such as a body of water, valley, or road, for the purpose of providing passage over the obstacle."*
OpenStreetMap Wiki	*"Bridge is an artificial construction that spans features such as roads, railways, water ways or valleys and carries a road, railway or other feature."*

identify the function of a bridge is providing passage over an obstacle (i.e., "carrying a pathway", and "providing passage"). However, the word *obstacle* is absent from the third definition where it is only stated that a bridge spans features. Also, it is not completely clear from the third definition that a bridge should provide passage. All the examples of the features carried by a bridge do provide passage, but the term "other feature" may be too open for interpretation by the operators.

6.4.2 Frequent itemset mining

This approach aims to learn which topological relations occur frequently among spatial objects of a given class. This is a typical *frequent itemset mining* task and can be achieved by retrieving the topological relations for the objects of interest and mining the recurrent topological patterns among them: Frequent itemset mining is the first step in the association rule mining process, aiming to identify associations between items in transactions [105]. In a database, a transaction usually corresponds to a row and items in a transaction correspond to column cells in a row. The aim is to find a set of items that frequently appear together [110].

Let a set $B = \{i_1, \dots, i_m\}$ of items be called the item base, and a database $T = \{t_1, \dots, t_n\}$ contain transactions. In this case item base B contains all items that occur in database T. The term itemset refers to any subset of B and can be denoted as $I \subseteq B$. A typical way to express how frequently an itemset occurs in the database is through computing the *support* of an itemset. The support of the itemset I is the number of transactions, i.e., $t_j, 1 \leq j \leq n$, in T that contain itemset I. It can also be expressed as the ratio of transactions that contain I to all transactions in T [110].

The proposed approach adopts the a-priori algorithm [105] to find the frequent itemsets in the dataset. There is no constraint set on the length of itemsets that should be retrieved, and the only threshold used is the minimum support of an itemset. In this thesis:

- *items* are topological relations that objects of a given class have with other objects in the database,
- *transactions* are rows of a database table where one row contains all topological relations between a core object and peripheral objects in the database (i.e., there is exactly one transaction for each core object), and
- *itemsets* are sets of topological relations that occur together in the same transaction, i.e., for one core object.

Figure 6.2 shows three examples of bridges (core objects) and their topological relations with other (peripheral) objects. The topological relations between each bridge in the examples and the peripheral objects can be seen as items and can be listed as transactions (Table 6.4). Then, itemsets can be formed by finding combinations of different topological relations that occur together in transactions. Finally, support for these itemsets can be calculated by dividing the count of all transactions where they occur with a total number of transactions, as shown in Table 6.4.

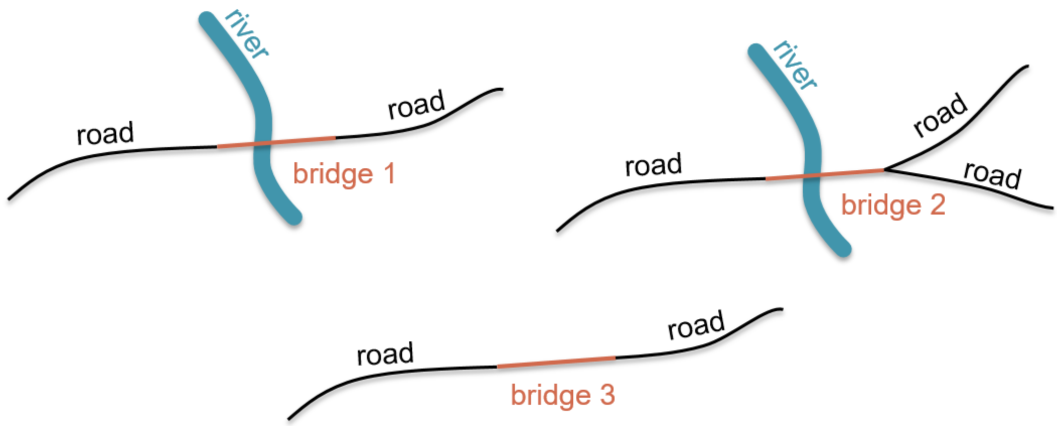


FIGURE 6.2: Examples of bridges and their topological relations with other objects.

TABLE 6.4: Dataset transactions consisting of topological relations (left), and frequent itemsets and their supports (right) for examples of bridges shown in Figure 6.2.

Topological relations		**Frequent itemsets**	
Bridge	Member relations	Itemset	Support
1	meets, crosses, meets	(meets)	$3/3 = 100\%$
2	meets, crosses, meets, meets	(crosses)	$2/3 = 66\%$
3	meets, meets	(meets, crosses)	$2/3 = 66\%$

6.5 Building accessibility analysis with RIM

The second approach in this chapter is concerned with the accessibility of buildings from the road network. Here, we define accessibility of a building as the ability to approach it in a straight line without having to avoid other buildings. The method proposed here uses the building's visibility from the road network to determine if it is accessible. For visibility we adopt a definition that "two points are mutually visible if the line segment connecting them is unoccluded" [74]. The method also assumes that access to building if it is accessible will be provided by the road that is closest to it. Although it is possible that some buildings are accessible from roads other than the closest, this raises questions such as when to search for other roads, what threshold should be used, and how to ensure that inaccessible buildings are not incorrectly classified as accessible? The answers are not trivial and this added complexity will be addressed in future research (see Chapter 9). Thus, the problem addressed here is: *given a dataset of buildings and a road network, determine which buildings are accessible from the road network*. The presented method shows how to determine the accessibility of a single building and can be applied iteratively to analyse all buildings in a given dataset.

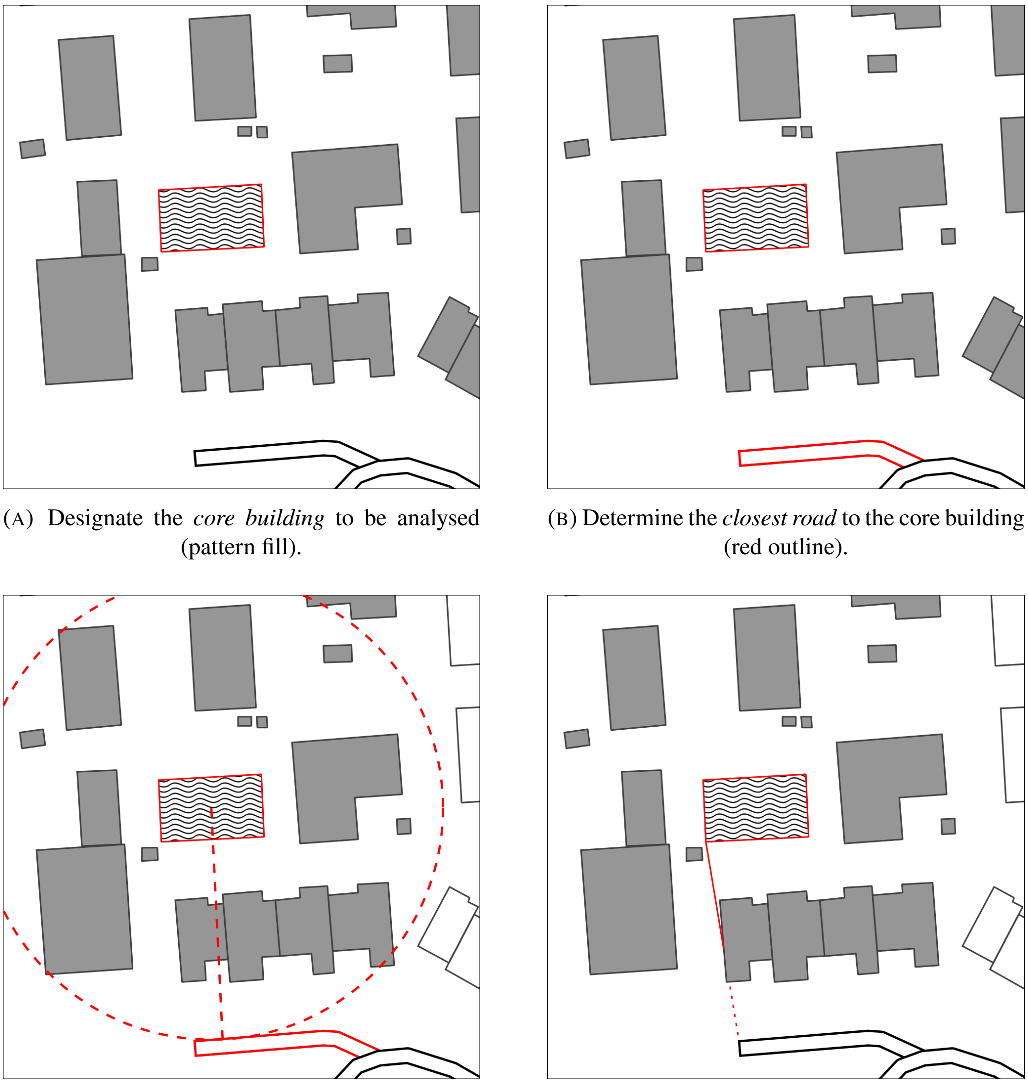
6.5.1 Refinement of RIM

To find out if a building is accessible from its closest road, our method proposes a modified version of RIM. RIM originally focuses on the core object whose position is being tested in regards to the two peripheral objects. The motivation for RIM was to determine if the core object is positioned *between* peripheral objects. In the building accessibility analysis, the spatial objects involved are the building being analysed – the *core building* $\boldsymbol{C}$, its *closest road* $\boldsymbol{R}$, and a set of *potentially occluding objects* $\boldsymbol{O}$ that may occlude the visibility between $\boldsymbol{C}$

and R. The distance between C and R will be denoted as d_{CR}. Here the focus changes from the object in between to one of the objects that would be considered peripheral in RIM – the *core building*. Figure 6.3 shows four major steps in the building accessibility analysis. First, the core building is designated (Figure 6.3a). Then, the roads in the proximity of the core building are analysed to find its closest road (Figure 6.3b). Then the potentially occluding objects are identified. Here, the algorithm only considers buildings for which the distance to the core building is less than d_{CR} (Figure 6.3c). It is still possible for buildings where the distance to the core building is greater than d_{CR} to be positioned between C and R and partially occlude access (e.g., if the closest road in Figure 6.3c was extended to the right, the white buildings in the bottom right of the scene would be between C and R). However, given the optimisation step introduced prior to the execution of this method where R is cut to the portion that is relevant to the context of C (see Section 7.3.3), such buildings are less likely to affect the outcome of the accessibility analysis. The last step is to analyse all the rays that exist between the core building and its closest road, and their intersections with the potentially occluding objects (Figure 6.3d). If there exists such a ray that is not intersected by any of the potentially occluding objects, then this ray represents a line of sight by which the core building is visible from its closest road. This also means that the core building can be accessed from the closest road and the algorithm will classify it as *accessible*. On the other hand, the potentially occluding objects between the core building and its closest road might be intersecting all of the rays completely occluding one from the other. This means that the core building cannot be accessed from its closest road and it will be classified as *inaccessible*.

6.5.2 Visibility between C and R given the occluder O

To determine whether the core building C can be seen and consequently accessed from its closest road R, visibility analysis is undertaken. The goal here is not to determine which parts of C are visible from R or obscured by O, but only to confirm if there exists at least one ray between C and R that is not intercepted by O. Such a ray represents the line of sight between C and R, and in the context of this method, proves that C is accessible from R. Algorithm 4 presents the computational steps of this analysis, while Figure 6.4 shows examples of three different scenarios of C, R, and O together with the visibility rays. Although



(A) Designate the *core building* to be analysed (pattern fill).

(B) Determine the *closest road* to the core building (red outline).

(C) Determine the relevant surrounding buildings that are within distance d_{CR} to the core building (grey fill).

(D) Test the visibility between the core building and its closest road, with respect to the surrounding buildings (red line shows an intercepted line of sight).

FIGURE 6.3: Illustration of steps in the proposed approach for the building accessibility analysis.

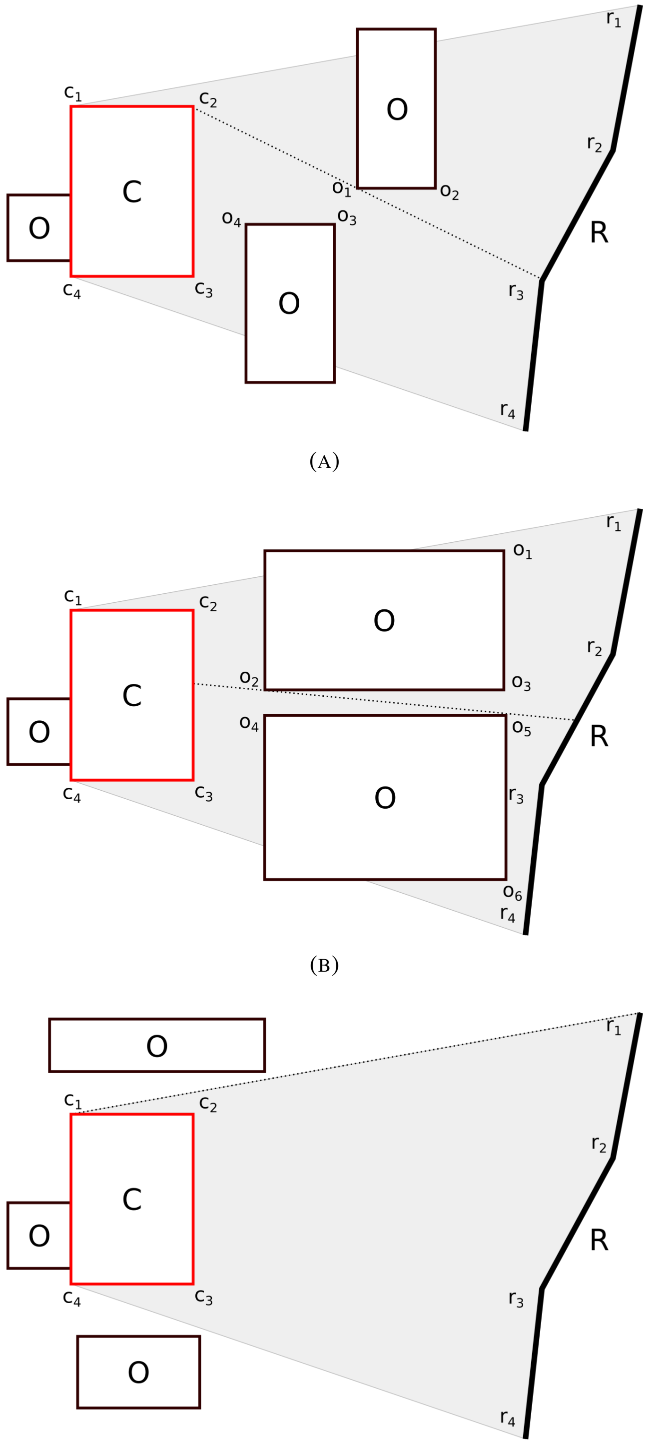
we have referred to O as a set of objects, O is here treated as a single mutli-geometry object (i.e., a Multipolygon consisting of one or more polygon parts).

The first step in this analysis is to find the area between C and R (Algorithm 4 line 1; gray areas in Figure 6.4). In the RIM, this area is called the ray area R_A and is defined as the area covered by all rays between C and R (for details of the ray area computation algorithm see [3]). The next step is to define all the rays between C and R that can potentially provide visibility between them. For this purpose, the algorithm extracts all the vertices of C, R, and

O, and keeps only the ones that are intersecting R_A (e.g., $c_1 - c_4, o_1 - o_4, r_1 - r_4$ in Figure 6.4a). This is because any straight line passing through a vertex that is disjoint from the R_A is impossible to share a point with both C and R. Thus, such lines cannot be visibility rays between C and R and can be omitted from the rest of the analysis. After the algorithm has extracted and selected the vertices, it creates straight lines passing through pairs of vertices achieved by combining the vertices of C with the vertices of O, the vertices of R with the vertices of O, and the vertices of O with the vertices of O (Algorithm 4 line 5). These lines are also extended on both sides to reach both C and R. After each line is created, the algorithm tests if the line is crossing the occluder object O (Algorithm 4 line 10). If yes, then this means that the line does not provide visibility between C and R. If not, then the algorithm tests if this line has intersections with both C and R (Algorithm 4 line 12). It may happen that the line touches only one of these objects and in that case it still does not provide visibility between C and R. Only if the line does not intersect O and touches both C and R it can be considered as a valid line of sight between C and R. If this is the case, then the iterative search process can end and the algorithm can return `True` to confirm that there is visibility between C and R (Algorithm 4 line 14).

If O does not have any vertices inside the R_A, then the algorithm will not be able to find the visibility ray in the previously described iterative search process (e.g., see Figure 6.4c). In this case, the algorithm will enter a second search process where it creates candidate rays through the pairs of vertices achieved by combining the vertices of C with the vertices or R (Algorithm 4 line 15). Because these lines are created from the vertices of C and R, they are guaranteed to intersect both of them. Thus, it is sufficient only to check if these lines are crossing O to test if they provide visibility between C and R. In the end, the algorithm will return `True` if there is visibility between C and R and `False` if there is not.

Figure 6.4 shows examples of three different placements of O relative to C and R. The core buildings C in these figures are shown in red outline, the ray areas between C and R are shown in grey, and the vertices of C, R, and O that intersect the R_A are denoted with lowercase letters (i.e., $c_i, r_i,$ ando_i). In Figure 6.4a, after the Algorithm 4 analyses the ray that passes through c_1 and o_1, it rejects this ray as the possible visibility ray because it crosses O. However, when the algorithm tests the ray that passes through c_2 and o_1, it will conclude that there exists visibility between C and R because this ray does not cross O and touches both C and R.



(A)

(B)

(C)

FIGURE 6.4: Illustration of the core object C, the closest road R, and potentially occluding objects O. The ray area between C and R is shown in gray and the dotted line between them is the line of sight.

Algorithm 4 Visibility test

Require: Geometries of two spatial objects (C and R) between which the line of sight is searched for, and a multi-geometry of the occluder object (O)

```
R_A ← CALCULATE_RAY_AREA(C,R)                          ▷ For this function, see [3]
cVertices ← Intersection(C.getVertices(),R_A)
rVertices ← Intersection(R.getVertices(),R_A)
oVertices ← Intersection(O.getVertices(),R_A)
for vertices in [cVertices, rVertices, oVertices] do
    for v1 in vertices do
        for v2 in oVertices do
            if v1 ≠ v2 then                ▷ Two distinct points needed to construct a line
                candidateRay ← MakeLine(v1,v2).Extend()
                if candidateRay.Crosses(O) then
                    visibility ← False
                else if candidateRay.Touches(C) AND candidateRay.Touches(R) then
                    visibility ← True
                    return visibility             ▷ Visibility ray found and search ends
for v1 in cVertices do
    for v2 in rVertices do   ▷ When there are no oVertices, construct rays between cVertices and rVertices
        if v1 ≠ v2 then
            candidateRay ← MakeLine(v1,v2).Extend()
            if candidateRay.Crosses(O) then
                visibility ← False
            else
                visibility ← True
                return visibility                 ▷ Visibility ray found and search ends
return visibility                                 ▷ Visibility ray not found, False returned
```

In Figure 6.4b there is no visibility ray that passes through any vertex of C or R. Instead, the algorithm finds the visibility ray by analysing rays that pass through different vertices of O (e.g., o_2 and o_5).

In Figure 6.4c, O is completely outside (i.e., topologically disjoint) of the R_A. In this case the algorithm analyses the rays that pass through vertices of C and R. Since the ray that passes through c_1 and r_1 does not cross O, the algorithm accepts it as the visibility ray between C and R.

6.6 Conclusion

In this chapter, two intrinsic error detection approaches have been proposed. These approaches rely on the assumption that in mature spatial dataset , the majority of the data is correct and the errors in data are outliers. Because of this assumption, the proposed approaches do not require any additional data and can detect errors in situations where the ground truth data are not available.

The first approach is called *discovery of topological constraints for classes of spatial objects*. This approach utilises the frequent itemset mining method to detect if objects of a given class frequently have the same topological relationships with other objects around them. If there is a highly supported pattern of topological relationships, then it can be adopted as a topological constraint saying that all objects of a given class need to have these relationships to be considered correct. Objects that do not satisfy the constraint will be labelled as potentially erroneous and will need to be checked by human experts. This approach will be tested in a case study in Chapter 7 on OSM data with both RIM and RTL topological relation models.

The second approach is called *building accessibility analysis with RIM*. It assumes that all buildings in a map database should be accessible from the road network and that this access is provided by the road that is closest to the building. If there are other buildings between the core building and its closest road, the approach tests if these buildings completely occlude the visibility between them. In this approach, visibility is used as a proxy for accessibility and if the building is not visible from its closest road then it means that it is inaccessible. This approach will also be tested in a case study in Chapter 7 on OSM data for the state of Victoria, Australia.

7

Experiments

The RTL and RIM models introduced in Chapters 4 and 5 have been embedded with the error detection approaches introduced in Chapter 6. This chapter presents three case studies that demonstrate the usage of these models and approaches to both analyse and detect errors in map data. The first case study looks at a set of university campus buildings and utilises RIM to determine which buildings are positioned *between* which other buildings. The second case study automatically discovers topological constraints on bridges in a map database and detects erroneous bridges using those constraints. The third case study classifies buildings in the map database as either accessible or inaccessible from the road network. Inaccessible buildings are then treated as indicators of potential errors and checked by expert mappers.

This chapter is based on published papers [1, 3, 4].

7.1 Analysing the betweenness of buildings with RIM

This case study demonstrates a possible use case, where RIM is used to analyse the betweenness of buildings at a university campus (Figure 7.1). There are twenty-nine buildings in this experiment, whose geometries were sourced from the OSM. They are first analysed to identify triplets of buildings that are candidates for betweenness. Then, these triplets are classified according to whether the core building is positioned *between* the peripheral buildings.

7.1.1 Betweenness based on RIM

This section discusses how RIM can be used to answer the question "Is object O between objects A and B?". This is the task of mapping between a formal model and common-sense understanding covered by linguistic terms. The most simple approach would choose a Boolean function, deciding whether this question is True or False. In that case, there may be several options for how this distinction could be made. The first option is to claim that O is not *between* A and B only when it is disjoint from all the rays between A and B (RIM23 in Figure 5.5), and that O is *between* A and B whenever it is intersecting any of the rays between A and B (RIM1-RIM22). However, this permissive option would also always treat the borderline cases such as RIM19, RIM20, RIM21, and RIM22 as being *between*, although some could argue otherwise.

Another option is to claim that O is clearly *between* A and B only when it is completely within the ray area formed by rays between A and B (RIM1-RIM9), and that O is otherwise not *between* A and B. Being more conservative than the previous option, this option considers nine out of 23 cases as O being *between* A and B. However, it is hard to argue that O is not *between* A and B in RIM10, since there is no ray between A and B which does not cross O.

It is thus demonstrated that answering the question of one spatial object being *between* other two is not trivial, and may depend on the application and context. Furthermore, it has been demonstrated that RIM can represent configurations of triplets of spatial objects in enough detail to provide a basis for reasoning about nuanced betweenness. The remainder of this case study will use three classes to differentiate cases of betweenness:

- ***O* is between** *A* and *B* when *O* is completely within the ray area between *A* and *B* (e.g., RIM1-RIM9),
- ***O* is debatably between** *A* and *B* when *O* is intersecting the ray area between *A* and *B*, but is not completely within it (e.g., RIM10-RIM22),
- ***O* is not between** *A* and *B* – when *O* is disjoint from the ray area between *A* and *B* (e.g., RIM23).

7.1.2 Refinement of candidate triplets for betweenness

Now that we have defined classes of betweenness for his case study, we need to calculate RIMs between the twenty-nine university campus buildings. Since RIM is computed on triplets of spatial objects, for RIM to be computed between all possible combinations of buildings would require $29^3 = 24,389$ RIM computations. Thus, the candidate combinations of buildings for betweenness are first filtered via Delaunay triangulation between buildings' centroids to capture the adjacency graph (Figure 7.2). Centroids are here only a suitable approximation. This step is reminiscent of the filter-refine strategy in evaluating spatial queries assisted by spatial indices in spatial databases [111]. Then, only the triplets of buildings connected by a path of length two are considered as candidates for immediate betweenness (e.g., buildings 10, 12, and 13 in Figure 7.2). This way, the number of required RIM computations is lowered to 337.

7.1.3 Results

In terms of the topological relations between object *O* and ray area R_A, only four distinct relations (according to [96]) occur in this case study: *disjoint*, *touches*, *overlaps*, and *within* (Table 7.1). Following the classification of betweenness presented in Section 7.1.1, cases where *O* is disjoint from R_A are seen as *not between*, cases where *O* touches or overlaps R_A are seen as debatably *between*, and cases where *O* is within R_A are seen as *between* (Table 7.1).

FIGURE 7.1: A set of twenty-nine buildings belonging to the Parkville campus of the University of Melbourne.

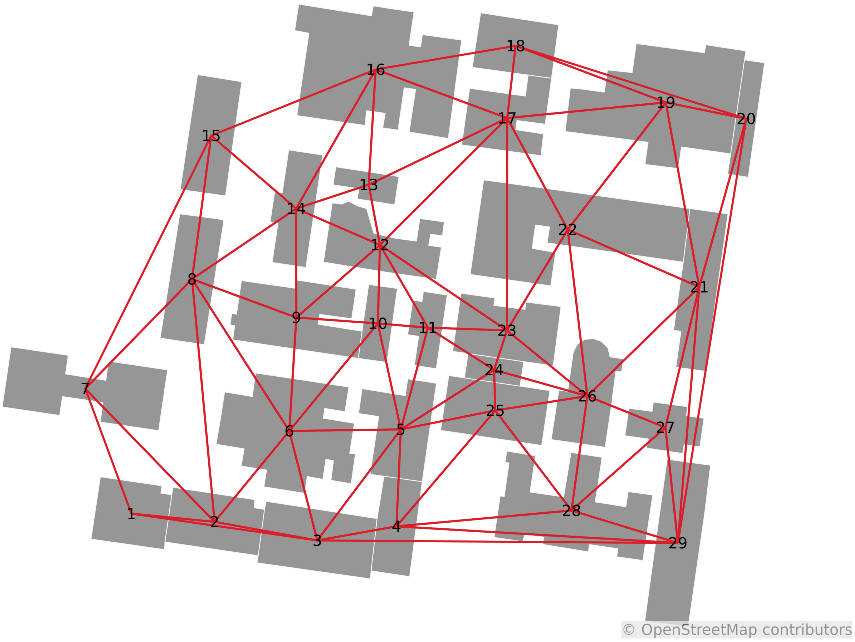


FIGURE 7.2: Delaunay triangulation (red lines) between centroids of buildings from Figure 7.1.

TABLE 7.1: Binary topological relations between O and R_A that occur in results.

O is not between A and B	O disjoint R_A	97
O is debatably between A and B	O touches R_A	6
	O overlaps R_A	221
O is between A and B	O within R_A	13
	triplets in total	337

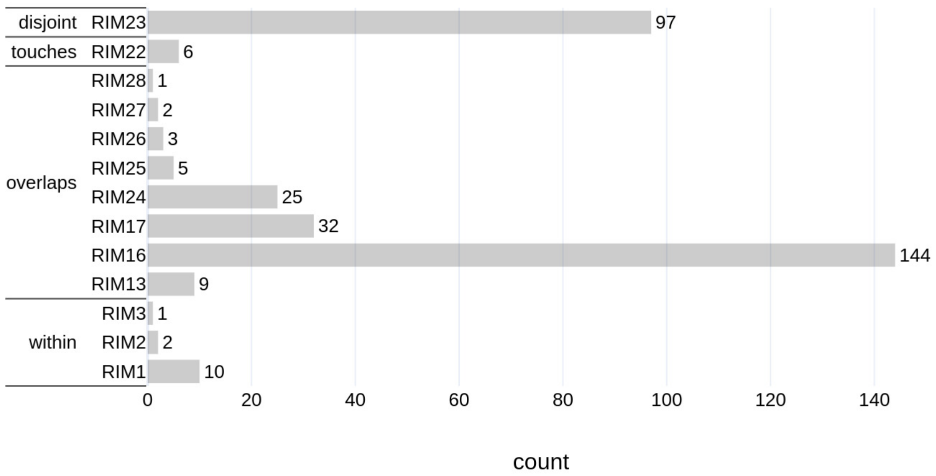
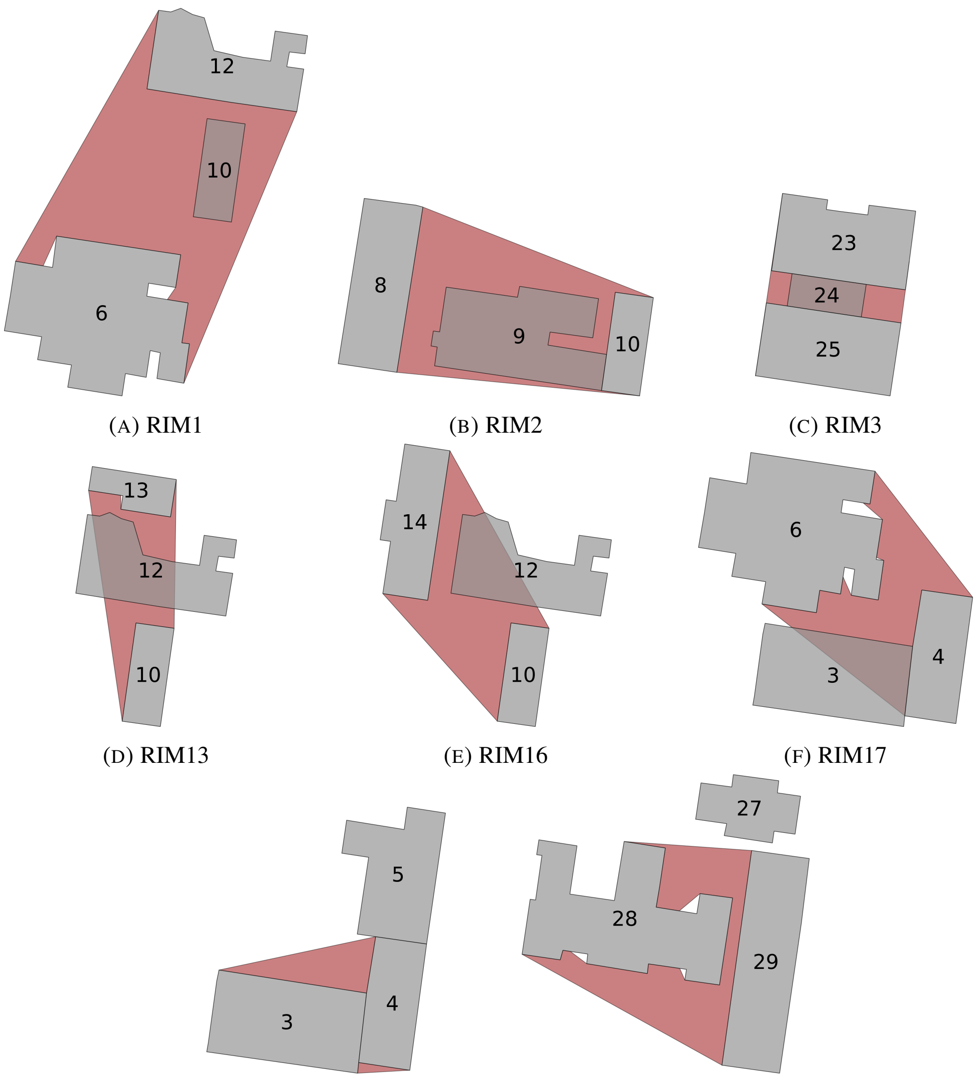


FIGURE 7.3: Distinct RIMs that occur in results.

Figure 7.3 shows these topological relations and the corresponding RIMs that occur in this case study. It can be seen that RIM was able to distinguish three times more cases of topological relations in triplets of buildings. While disjoint and touches each only have one corresponding RIM, the topological relation overlaps may be divided into eight different RIMs, and within may be divided into three different RIMs.

Figures 7.4 and 7.5 show examples of triplets of buildings for each of the thirteen distinct RIMs that occur in the results of this case study. Out of these thirteen RIMs, eight of them can be found in the twenty-three RIMs that were demonstrated in Figure 5.5. Examples for these RIMs are shown in Figure 7.4. The remaining five RIMs have not been identified in the simple theoretical cases of three non-convex polygons. The naming for these new RIMs continues on the numbering from the previous RIM cases from Figure 5.5 (i.e., there were



(A) RIM1 (B) RIM2 (C) RIM3

(D) RIM13 (E) RIM16 (F) RIM17

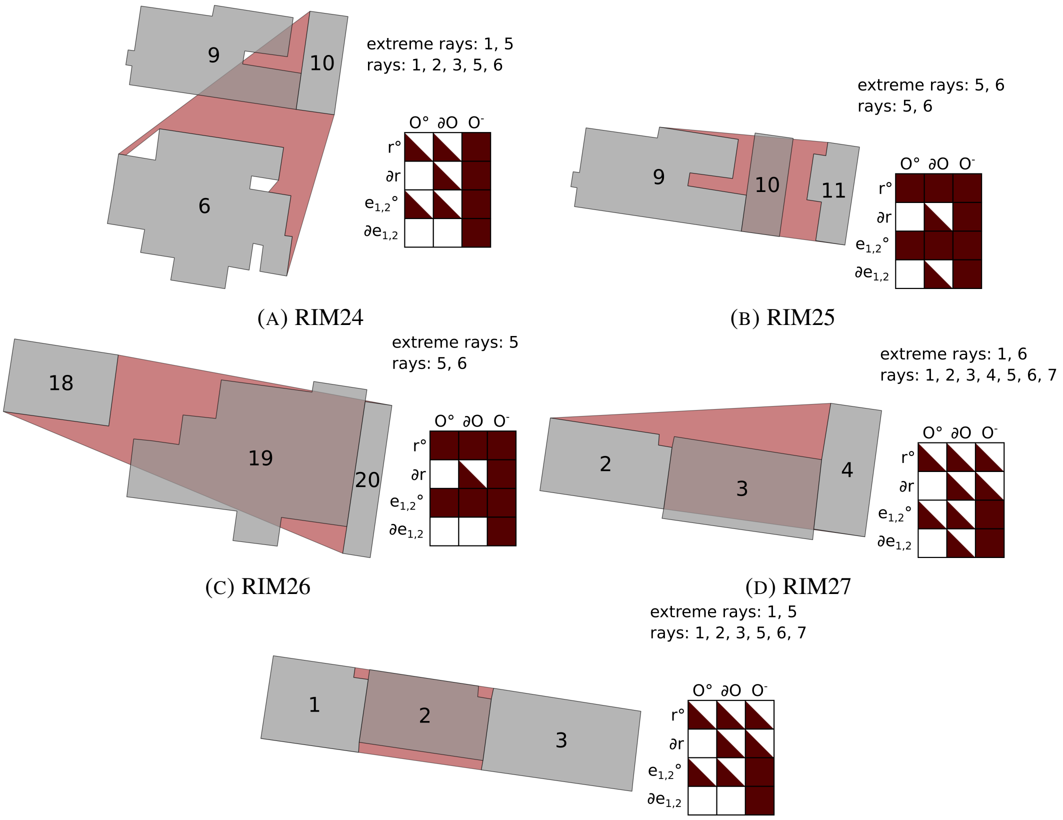
(G) RIM22 (H) RIM23

FIGURE 7.4: Examples of RIMs that occur in the case study, and that were demonstrated in Figure 5.5.

twenty-three), which gives RIM24, RIM25, RIM26, RIM27 and RIM28 (Figure 7.5). As such, it is important to note that RIM presents a means to characterise nuanced relationships, not a fully elucidated, exhaustive set of distinct relationships – these may be unbounded.

7.1.4 Discussion

For triplets of buildings in this case study, it was tested if one of the buildings is positioned between the other two. If this task was carried out by looking at the topological relation between the ray area (R_A) between the peripheral objects A and B and the core object O, it would be possible to differentiate only four scenarios – O is disjoint, or touches, or overlaps,



(A) RIM24 (B) RIM25 (C) RIM26 (D) RIM27 (E) RIM28

FIGURE 7.5: Examples of RIMs that occur in the case study, but were not previously defined. Each figure also lists rays and extreme rays that occur in that scenario, and the corresponding RIM matrix.

or within the R_A (Table 7.1). This is reminiscent of the projective ternary relation model proposed by Clementini and Billen [10] which defines an area between the peripheral objects and captures whether the core object intersects this area (see Figure 2.4). The results of the case study showcase that RIM can distinguish thirteen different scenarios for the given set of buildings which is over three times more. This shows that RIM can detect much more nuance in such cases which is useful when studying different cases of betweenness of spatial objects.

The case study used a simplistic definition of betweenness where O is considered between A and B if it intersects any ray between A and B. In the future, it would be sensible to carry out subject testing to evaluate which cases would people classify as between or not between and if they would see betweenness as a binary property where only those two outcomes are possible. The subjects' opinions could then be mapped to different RIM cases, which would give plausibility to the claims that certain RIMs represent betweenness of spatial objects.

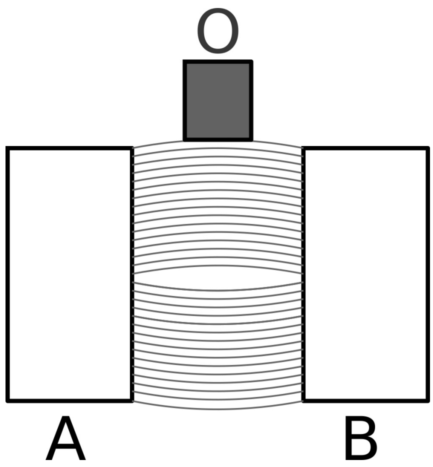


FIGURE 7.6: An example of curved or bulging rays between A and B.

It would also be interesting to see if people would find other relations such as *across* to be similar to *between* and if RIM could be applied to capture these relations as well.

Another direction of future work could go towards investigating different definitions of rays in RIM. At the moment, all of the rays in RIM are considered to be straight lines. However, people may be lenient to classify objects as between if they are slightly offset from the ray area between the peripheral objects. It may also be possible that people would tolerate different offsets from the ray area in different directions. For example, a larger offset of O might be tolerated in the direction perpendicular to the extreme rays than parallel to them. Mathematically, this could be represented with curved or bulging rays such as shown in Figure 7.6.

7.2 Topological constraints for bridges in OSM

This case study tests the approach for discovery of topological constraints proposed in Section 6.4 on bridge objects in the OSM dataset. In this case study, all the instances of class {bridge} are analysed to learn their most frequent topological relations with their surrounding objects. Apart from the topological relations themselves, the different types of geometry of the peripheral objects (i.e., line or polygon) are distinguished as well. Thus, results are shown in the form of frequent itemsets that consist of topological relations between bridges as core objects and other peripheral objects, together with corresponding supports.

TABLE 7.2: Numbers of line features and bridges in the OSM dataset for the state of Victoria, Australia.

Number of line objects	450, 230
Number of line objects annotated as bridge	10, 870
Number of polygon objects	404, 221

7.2.1 Experimental setup

The raw OSM data for the State of Victoria, Australia were extracted through the OSM *Overpass API*[1]. The experimental setup consists of a *PostgreSQL 9.5.13*[2] database with the *PostGIS 2.4.2*[3] extension, where OSM data were imported using the *osm2pgsql 0.88.1*[4] importer.

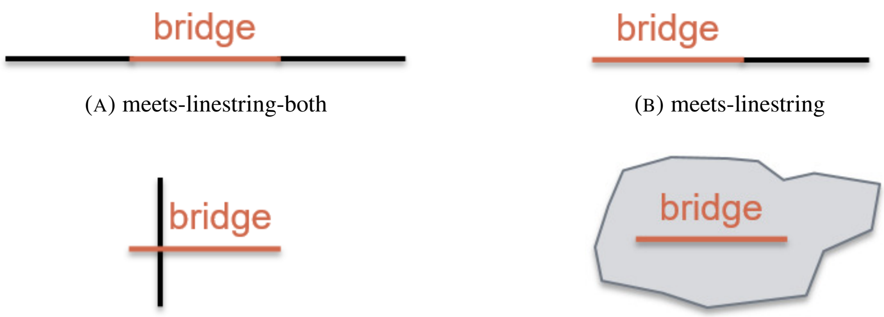
A preselection step was used to select bridge objects from the raw data. It was guided by the OSM wiki documentation which specifies that bridges should have a line geometry, and should contain a *bridge=** tag (i.e., * is a standard way of denoting any value used with a specific attribute key). Thus, objects that were annotated as bridges but did not have a line geometry were excluded from the experiment. Furthermore, objects tagged as *bridge=no* were also excluded. Note that this preselection was applied only to identify the set of core objects (*bridges*) in the dataset. For peripheral objects related to bridges, all objects with a line or polygon geometry were considered. The counts of line objects and bridges in this dataset are shown in Table 7.2. These counts are used to calculate the support of frequent itemsets and consequently to identify patterns.

This case study also compares the RTL model that was proposed in Chapter 3 to the 9IM. For the purposes of this comparison, all experiments were evaluated with both representations of topological relations. Furthermore, all topological relations in the results are represented with aliases for improved readability (Table 7.3), and the illustrations of the topological relations are shown in Figure 7.7.

[1]http://wiki.openstreetmap.org/wiki/Overpass_API
[2]https://www.postgresql.org/
[3]http://postgis.net/
[4]http://wiki.openstreetmap.org/wiki/Osm2pgsql

TABLE 7.3: Aliases for topological relations in this case study.

Alias	RTL	9IM
meets-linestring-both	`FFTFTTTTT-linestring-both`	—
meets-linestring	—	`FFTFTTTTT-linestring`
crosses-linestring	`TFTFFBTTT-linestring`	`TFTFFTTTT-linestring`
within-polygon	`TFFBFFTTT-polygon`	`TFFTFFTTT-polygon`



(A) meets-linestring-both

(B) meets-linestring

(C) crosses-linestring

(D) within-polygon

FIGURE 7.7: Illustrations of the topological relations that are present in the results between bridge (core) objects and other (peripheral objects).

7.2.2 Support threshold for frequent itemsets

The minimum support threshold for the discovery of the frequent itemsets has been set to 20% in this case study. This threshold was empirically determined – larger thresholds lead to frequent itemsets that are too specific, while smaller thresholds yield uninformative itemsets. Figure 7.8 shows how the number of itemsets diminishes with the increase of the minimum support threshold. At the threshold of 1%, both 9IM and RTL yield almost 50 itemsets, most of which have support lower than 5%. Also, if the threshold was set to 0.001%, there would be almost 2,000 possible itemsets. On the other side, thresholds of approximately 40% and higher yield less than five itemsets, and the highest possible threshold that would yield at least one itemset would be approximately 90%. Thus, given the assumption behind the approaches in this thesis that we want to learn the patterns of the majority of the data (i.e., those would be patterns with support of at least 50%), it can be concluded that the 20% threshold provides the best balance between specificity and accuracy of the itemsets. The selected threshold is used to limit the itemsets in this study to a number that is manageable for manual analysis and no patterns that describe the majority of the data will be omitted.

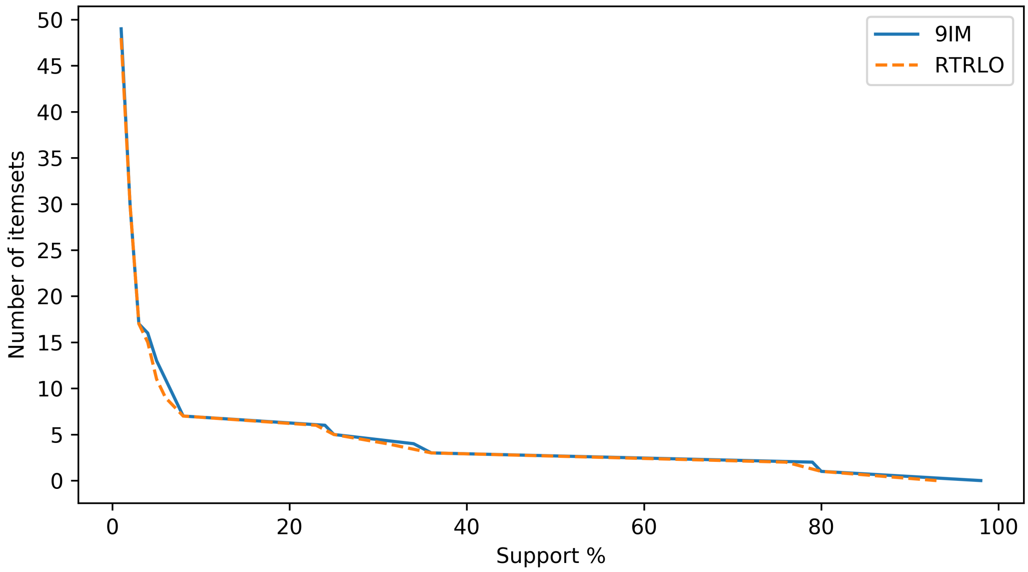


FIGURE 7.8: Numbers of the discovered frequent itemsets for different minimum support thresholds, dataset for Victoria.

7.2.3 Identified frequent itemsets

The proposed approach for the discovery of topological constraints was first applied to the preselected bridges, and the results of this step are shown in Tables 7.4 and 7.5. The rows in these tables represent the detected itemsets with support higher than the threshold (i.e., 20%). For example, the first line in Table 7.4 shows that there are 10,548 bridges in the dataset that have the topological relation *meets-linestring* with at least one peripheral object. This does not mean that these bridges do not have other topological relations with peripheral objects. A subset of them also have other topological relations such as *crosses-linestring*, *within-polygon*, or both *crosses-linestring* and *within-polygon* as shown in rows three, five, and seven respectively. Also, the examples of itemsets from Table 7.5 are shown in Figure 7.9.

Consecutively, the identified frequent itemsets were applied to all line objects in the dataset. The frequent itemset mining with the same minimum support threshold (i.e., 20%) was applied to all line objects in the dataset (Tables 7.6 and 7.7), to identify unclassified, or potentially misclassified objects that satisfy the constraints for the class {bridge}. This helps evaluate how specific the learned topological constraints are to the class {bridge}. In this step, no preselection other than restricting the set to line objects has been made.

TABLE 7.4: Frequent itemsets for bridges in the state of Victoria using the 9IM.

Item 1	Item 2	Item 3	Support	(%)
meets-linestring	—	—	10,548	97.04
—	crosses-linestring	—	8,648	79.56
meets-linestring	crosses-linestring	—	8,489	78.10
—	—	within-polygon	3,819	35.13
meets-linestring	—	within-polygon	3,638	33.47
—	crosses-linestring	within-polygon	2,623	24.13
meets-linestring	crosses-linestring	within-polygon	2,553	23.49

TABLE 7.5: Frequent itemsets for bridges in the state of Victoria, using the RTL.

Item 1	Item 2	Item 3	Support	(%)
meets-linestring-both	—	—	10,001	92.01
—	crosses-linestring	—	8,649	79.57
meets-linestring-both	crosses-linestring	—	8,190	75.34
—	—	within-polygon	3,821	35.15
meets-linestring-both	—	within-polygon	3,366	30.97
—	crosses-linestring	within-polygon	2,624	24.14
meets-linestring-both	crosses-linestring	within-polygon	2,441	22.46

TABLE 7.6: Frequent itemsets for all line objects in the state of Victoria using the 9IM.

Item 1	Item 2	Item 3	Support	(%)
meets-linestring	—	—	231,184	51.35
—	crosses-linestring	—	131,227	29.15
meets-linestring	crosses-linestring	—	83,406	18.53
—	—	within-polygon	211,203	46.91
meets-linestring	—	within-polygon	92,548	20.56
—	crosses-linestring	within-polygon	43,951	20.56
meets-linestring	crosses-linestring	within-polygon	23,072	5.12

7.2.4 Results

Assuming that the majority of data in the dataset is classified correctly, the existing classification in the dataset has been used as the ground truth for evaluation of the results. Contingency tables were created to evaluate the classification performed by the learned model

(A) meets-linestring-both

(B) crosses-linestring

(C) meets-linestring-both and crosses-linestring

(D) within-polygon

(E) meets-linestring-both and within-polygon

(F) crosses-linestring and within-polygon

(G) meets-linestring-both and crosses-linestring and within-polygon

FIGURE 7.9: Examples of the itemsets from Table 7.5, shown on bridges in OSM.

(Tables 7.8 and 7.9). They relate to the conceptual contingency table (Table 6.1b), as the ground truth for data in this kind of applications and this experiment is missing. These tables

TABLE 7.7: Frequent itemsets for all line objects in the state of Victoria, using the RTL.

Item 1	Item 2	Item 3	Support	(%)
meets-linestring-both	—	—	113,095	25.12
—	crosses-linestring	—	133,634	29.68
meets-linestring-both	crosses-linestring	—	49,427	10.98
—	—	within-polygon	218,512	48.53
meets-linestring-both	—	within-polygon	38,749	8.61
—	crosses-linestring	within-polygon	44,838	9.96
meets-linestring-both	crosses-linestring	within-polygon	11,849	2.63

TABLE 7.8: Contingency table for the results using the 9IM. ∧ means that this item is applied in conjunction with the previous item(s) to the left of it in an itemset, applied on the OSM dataset for the state of Victoria.

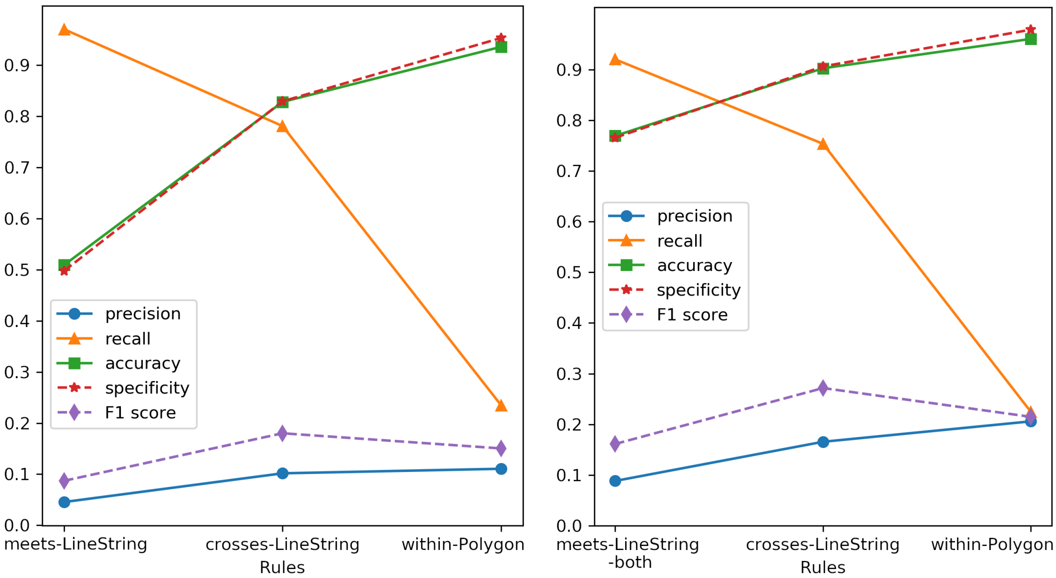
		Data					
		meets-linestring		∧ *crosses-linestring*		∧ *within-Polygon*	
		bridge	not bridge	bridge	not bridge	bridge	not bridge
Model	bridge	10,548	220,636	8,489	74,917	2,553	20,519
	not bridge	322	218,724	2,381	364,443	8,317	418,841

compare the numbers of features classified as either bridges or not bridges in the model and the data, across three combinations of topological rules that define bridges. For example, on the left side of Table 7.8, a feature must have a topological relation *meets-linestring-both* with other objects at least once to be classified as a bridge. In the middle columns of the same table, a feature is classified as a bridge only if it has both *meets-linestring-both* and (i.e., ∧) *crosses-linestring* topological relations. Finally, in the rightmost columns, a feature must have all three topological relations with peripheral features to be classified as a bridge.

The quality measures for the proposed method using the 9IM are shown in Figure 7.10a and in Table 7.10. The same measures for the proposed method using refined topological relations for line objects are shown in Figure 7.10b and in Table 7.10. The proposed method was also applied to the OSM dataset for Switzerland to test if the method will be general enough to yield similar results elsewhere in the world. The quality assessment of these results is shown in Figure 7.11.

TABLE 7.9: Contingency table for the results using the RTL. ∧ means that this item is applied in conjunction with the previous item(s) to the left of it in an itemset, applied on the OSM dataset for the state of Victoria.

		Data					
		meets-linestring-both		∧ *crosses-linestring*		∧ *within-Polygon*	
		bridge	not bridge	bridge	not bridge	bridge	not bridge
Model	bridge	10,001	103,094	8,190	41,237	2,441	9,408
	not bridge	869	336,266	**2,680**	398,123	8,429	429,952



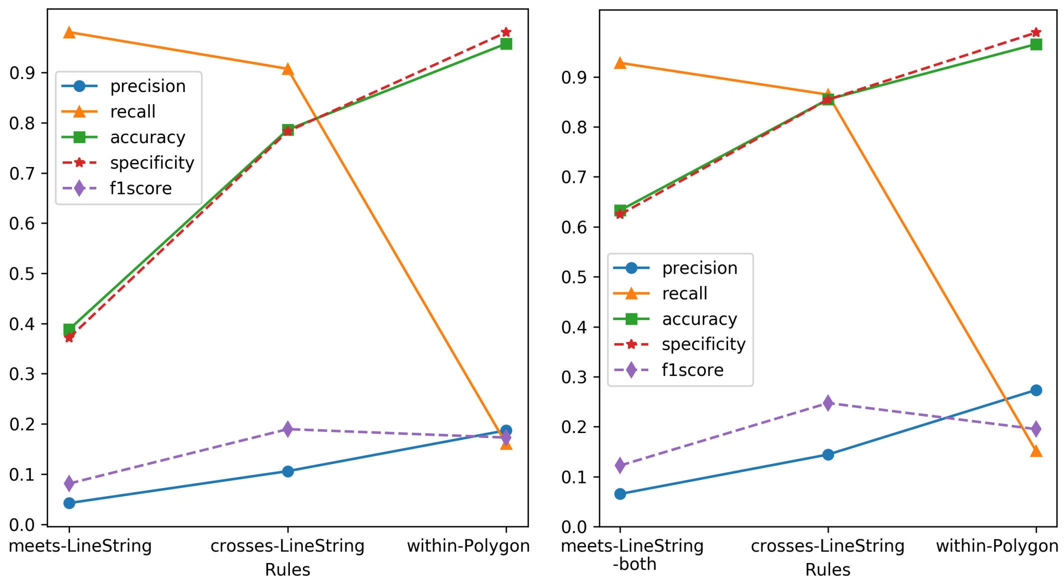
(A) The 9IM.

(B) Refined topological relations for line objects.

FIGURE 7.10: Quality measures for the results using the 9IM and RTL, applied on the OSM dataset for the state of Victoria.

7.2.5 Discussion

The first observation that can be made in the results of this case study is that the itemsets discovered in the experiment (Tables 7.4 and 7.5) are strongly comparable with the common-sense, textual definitions of bridges (Section 6.4, Table 6.3). These definitions state that a bridge should carry a path or a road over an obstacle. In the discovered itemsets, the bridge crossing an obstacle is represented with the *crosses-linestring* topological relation. In the OSM dataset, a bridge is also a part of the road or pedestrian network, and as such represents a specialisation of a road network. This means that a bridge is not only a structure that carries a path or a road but is the path or the road itself. This is the first, most common item in the itemsets discovered, captured as the *meets-linestring* relation.



(A) The 9IM. (B) Refined topological relations for line objects.

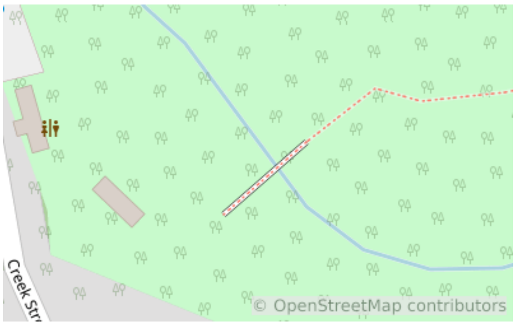
FIGURE 7.11: Quality of the results using the 9IM and RTL on the OSM dataset for Switzerland.

Tables 7.4 and 7.5 also show that the proposed method was able to discover seven different frequent itemsets for bridges in the dataset with the minimum support of 20%. An important difference between the analysis based on the RTL model (Table 7.5) and the 9IM (Table 7.4) is in the first (i.e., the most frequent) item. With the RTL, the proposed method was able to identify that 92% of bridges in the dataset are meeting other line objects on both sides (i.e., *meets-linestring-both*). In contrast, the 9IM was only able to learn that bridges usually meet at least one other line object in the dataset. The fact that the 9IM has also included the cases where these relations are present only on one side of the bridge has resulted in higher support of 97%. Thus, only 5% of bridges meet other line objects on one side, and these are likely to be outliers or erroneous representations warranting inspection (examples of these bridges are shown in Figure 7.12). Thus, the refined topological relations for line objects yield superior results enabling to identify and verify these marginal cases. A 5% error rate represents approximately 500 objects just in the limited dataset that was used.

This difference is also present in the results for all line objects (Tables 7.6 and 7.7) where many more line objects have the relation *meets-linestring* (51%), compared to the relation *meets-linsetring-both* (25%). This is to be expected because the latter is a proper subset of the *meets-linestring* relation set. Yet, it is also insufficiently specific to bridges. This effect propagates through the rest of the itemsets where these relations are present, while the other

TABLE 7.10: Quality measures for the results using the 9IM and RTL, applied on the OSM dataset for the state of Victoria.

	9IM		
	meets-linestring	∧ *crosses-linestring*	∧ *within-Polygon*
Precision	0.0456	0.1018	0.1107
Recall	**0.9704**	0.7810	0.2349
Accuracy	0.5092	0.8283	0.9360
Specificity	0.4978	0.8295	0.9533
F1 score	0.0872	0.1801	0.1504
	RTL		
	meets-linestring-both	∧ *crosses-linestring*	∧ *within-Polygon*
Precision	0.0884	0.1657	**0.2060**
Recall	0.9201	0.7535	0.2246
Accuracy	0.7691	0.9025	**0.9604**
Specificity	0.7654	0.9061	**0.9786**
F1 score	0.1614	**0.2717**	0.2149



(A) footway bridge connected on a single side

(B) cycleway bridge meeting cycleway at a non-break point

(C) footway bridge connected on a single side

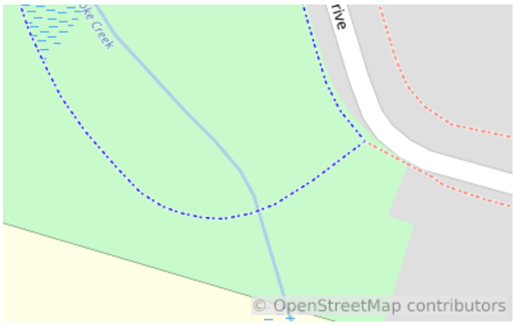
(D) road bridges connected on a single side

FIGURE 7.12: Examples of bridges meeting line objects only on one side.

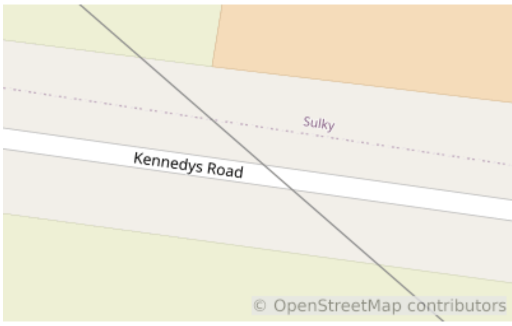
itemsets have similar support for both representations of topological relationships.

Evaluation of the results of this case study (Figure 7.10 and Table 7.10) shows how the quality of the classification changes with different itemsets. The evaluation is performed using three rules – *meets-linestring-both*, *crosses-linestring*, and *within-polygon*, which are being added from left to right with logical operator $\wedge$ (i.e., AND). In the last case, this means that an object needs to have all three relationships with its surrounding objects to be classified as a bridge.

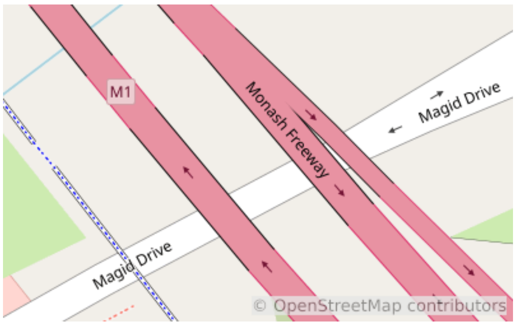
Figure 7.10b and Table 7.10 show that precision, accuracy, and specificity improve with added rules (items), while recall starts high (92% for the refined relations) and sharply decreases when the third rule is introduced. They also show that the refined topological relations for line objects perform better than the 9IM. Both accuracy and specificity have high values starting above 75% for *meets-linestring-both* and rising to over 95% when all three rules are applied. This shows that the proposed method can classify most of the features correctly and that the discovered rules are specific to bridges compared to other line objects in the dataset. On the other hand, the precision of the proposed method is low and reaches just 20% when all three rules are applied. The reason is that many features that are not bridges are committed – i.e., incorrectly classified as bridges (Table 7.9). In other words, these conditions are only necessary for bridges but are not unique to this class of objects. Figure 7.13 shows four examples of line objects which are not annotated as bridges in the data, but are classified as bridges by the model because they have topological relations *meets-linestring-both* and *crosses-linestring* with their peripheral objects. Example in Figure 7.13a shows a cycleway that is crossing a stream without a bridge which means that the bridge is possibly missing from the dataset. In Figures 7.13b and 7.13d, roads are crossing other line objects (i.e., a power-line and a footway) and meeting other road segments on both sides which makes them topologically identical to bridges in the model. The example in Figure 7.13c shows us that the obstacles being crossed by bridges can also have the topological characteristics of a bridge if they are a part of a network (i.e., meeting other line objects on both sides) because they will already have the topological relation *crosses-linestring* with the bridge. This problem may be solved by making further implications from semantics (i.e., a river is not likely to be a bridge, regardless of its topological relations), or by considering additional attribute information (i.e., tag *layer*=* is often used in OSM to describe vertical relationships between overlapping objects such as roads).



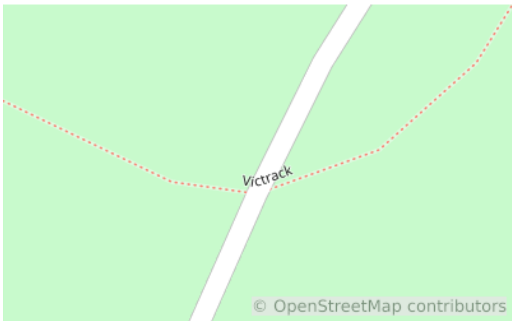
(A) cycleway classified as a bridge



(B) road crossing a power-line classified as a bridge



(C) road being crossed by bridges (white) classified as a bridge



(D) road crossing a footway classified as a bridge

FIGURE 7.13: Examples of lines incorrectly classified as bridges by the model

The F1 score peaks at 27% with the rules *meets-linestring-both* and *crosses-linestring* applied. The low value of the F1 score is primarily caused by low precision. The RTL outperforms the 9IM by improving precision by 6%, accuracy and specificity by 7%, and the F1 score by 9% on the combination of *meets-linestring-both* and *crosses-linestring*. The relative improvement on F1 score with the refined topological relations method is thus 50% over the 9IM-based rules. The 9IM achieves a slightly higher recall by 3%, since the relation *meets-linestring* is more general than the *meets-linestring-both* relation. However, recall drops significantly for both 9IM (from 78% to 23%) and the RTL (from 75% to 22%) when the topological relation *within-polygon* is introduced. This means that there are a lot of bridges that are not within any polygon objects, and these bridges are not recognised by the method when *within-polygon* is used. Thus, it would be best not to use the *within-polygon* relation to detect potentially erroneous bridges because the method will return too many false positives. In summary, the overall classification performs best when applying the proposed refined topological relations over the 9IM, with the combination of rules *meets-linestring-both* and *crosses-linestring*.

Regarding the detection of errors in OSM, we believe that this method would perform best if the itemset *meets-linestring-both* ∧ *crosses-linestring* was adopted as a topological constraint that defines which topological relations a valid bridge object must have with other objects. This itemset has a high support of 75% and is fit to describe the majority of bridges in the Victorian OSM dataset. When applied to all objects classified as bridges, it detects 2,680 potentially erroneous bridges that are missing one or both of these topological relations. When presented with a bridge that is potentially erroneous, mappers could quickly assess the situation from satellite imagery and determine whether these topological relations are missing due to missing features (e.g., roads or other objects that are present in satellite imagery, but are not mapped in OSM) or if the bridge is valid without them. This would be a useful addition to the existing OSM tools that alert users about problems such as disconnected roads, because they are not able to detect errors such as the ones shown in Figures 7.9c, 7.9e, and 7.12.

7.3 Inaccessible buildings in OSM

Routing algorithms depend on the quality of map data they use to calculate and provide users with correct routes to their destinations. The approach proposed in Section 6.5 enables to assess the accessibility of buildings in a map dataset. By detecting inaccessible buildings, it points to potential errors in map data. To demonstrate and test the proposed approach, it has been applied to buildings in the OSM dataset for the State of Victoria, Australia. This case study recognises that there are some exceptions, such as sheds, storage buildings, and other outbuildings that are often accessible only *through* the main building on the property. Thus, in this case study, we have only analysed buildings for which it is presumed they *should* be accessible from the road network. Using binary classification, the analysed buildings have been classified as either *accessible* or *inaccessible*. In the end, *inaccessible* buildings have been presented via an online mapping challenge to mappers who have then assessed individual situations and corrected any potential errors in OSM.

TABLE 7.11: Numbers of polygon objects, buildings, line objects, and highways in the imported OSM dataset.

Type of objects	Count
Imported polygon objects	566,265
Polygon objects annotated with *building*=*	397,215
Imported line objects	616,318
Line objects annotated with *highway*=*	471,046

7.3.1 Dataset and experimental setup

We obtained the OSM data for the State of Victoria, Australia as the OSM data extract[5] on March 26, 2020. The downloaded data extract was already confined to the territory of Victoria and no additional steps were taken in regards to the geographical extents of the dataset. The setup consists of a PostgreSQL database (version 9.5.21) extended with PostGIS (version 2.4.2). The data were imported using the osm2pgsql importer (version 0.88.1), and a part of the processing in the experiment was done with Python3 scripts and the RIM Python3 package (https://pypi.org/project/rim).

We have extracted the buildings and the road network data from the overall dataset since this is the only information that is needed for the proposed method. The osm2pgsql importer by default divides the data across four core tables: *planet_osm_point*, *planet_osm_line*, *planet_osm_roads*, and *planet_osm_polygon*. From the *planet_osm_polygon* table, all objects which are tagged with the *building*=*[6] tag were extracted and stored into the *osm_buildings* table. Likewise, all objects from the *planet_osm_line* table that are tagged with the *highway*=*[7] tag were extracted and stored into the *osm_highways* table. We have decided to use *planet_osm_line* table as a source of the road network data instead of the *planet_osm_roads* table because the latter also contains administrative boundaries and other objects missing the *highway*=* tag. Table 7.11 shows the relevant statistics for buildings and highways.

Figure 7.14 shows the most frequent building tags in the dataset. The majority of the buildings are tagged with the generic tag *building=yes*. This tag only confirms that the object is a building, but does not make it clear what is the type of the building. This is problematic when we want to focus our analysis on certain types of buildings and exclude others. Thus,

[5] http://download.openstreetmap.fr/extracts/oceania/australia/
[6] https://wiki.openstreetmap.org/wiki/Key:building
[7] https://wiki.openstreetmap.org/wiki/Key:highway

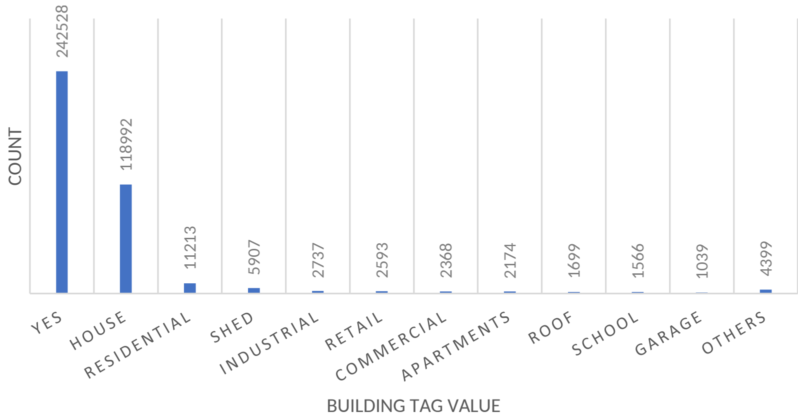


FIGURE 7.14: The most frequent building tags in the imported OSM dataset. *Others* represents all other tags that occur less frequently.

we have used the area of the building in the results section to semantically differentiate main buildings from the outbuildings. From the other tags, it is notable that there are almost $6,000$ sheds and over $1,000$ garages in the dataset. These can be considered outbuildings and are not expected to have access from the closest road. Instead, the access to these buildings is often provided through the main building on the property or other roads which are usually not mapped (e.g., private driveways and dirt tracks). Lastly, it is notable that there are over $4,000$ buildings tagged with other less frequent tags (135). These tags have not been analysed individually, since there are 135 of them and they all have less than $1,000$ occurrences (and often much less), which translates to 0.25% of buildings in the dataset.

Table 7.12 shows all of the highway tags that are present in the dataset and how they have been divided them into five categories. Since this study is concerned with the accessibility of buildings from the road network, this categorisation helps to differentiate OSM highways that are assumed to be a part of that network (i.e., *car* category) from those that are not. However, there are also more than $47,000$ highways that are generically tagged as *highway=unclassified*. For these highways we cannot be certain if they are part of the road network, so they could not be excluded from the analysis. Thus, only the highways categorised as car and unclassified are used in the rest of this case study.

TABLE 7.12: Highway categories, the highway tags that belong to each category, and their counts. Only highways categorised as car or unclassified were considered in this case study.

Highway category	Highway	Count
Car	living_street, motorway, primary, residential, road, secondary, service, tertiary, trunk	292,161
Foot and cycle	cycleway, footway, path, pedestrian, steps, track	113,761
Unclassified	unclassified	47,020
Car link	motorway_link, primary_link, secondary_link, tertiary_link, trunk_link	11,046
Other	abandoned, bridleway, bus_stop, construction, corridor, crossing, disused, platform, proposed, raceway, rest_area, services, traffic_island, yes	7,058

7.3.2 Experiment execution

The experiment was carried out following the four steps of the building accessibility approach as shown in Section 6.5 and in Figure 6.3. The analysis was partly executed in the database with the SQL queries and partly with Python scripts where the RIM Python3 package was utilised to finally assess the visibility of buildings from their closest roads.

All steps of the proposed approach were straightforward to execute apart from the second step where the core building's closest road is found. In this experiment we have limited the search distance for closest roads in this step to 400 m, as we believe this is the maximum realistic distance a person would walk from the road over empty ground to access a building. This distance is calculated from the building's bounding box to the road and out of all roads that satisfy this criteria the one that is closest to the core building is designated as its closest road. Buildings that have no roads within this search distance are by default classified as *inaccessible*.

Also, a pre-processing step is applied to cases where core buildings and the closest roads are intersecting each other (i.e., the distance between them is 0 m). In these cases the core building is automatically classified as *accessible* because it is impossible for its access to be completely occluded. This leaves less buildings for which the visibility analysis needs to be performed and speeds up the total execution time of the experiment.

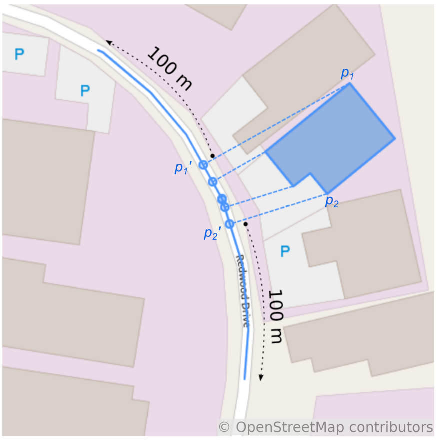


FIGURE 7.15: The closest roads were cut to a 100 m from the core building's projection points towards the ends of the road linestring to improve the speed of calculations.

7.3.3 Analysis optimisation

In all cases where core buildings and the closest roads are disjoint (i.e., the distance between them is > 0 m) the analysis is optimised by only considering a portion of the closest road that is relevant for the core building. This is needed because some road segments in OSM can be up to several kilometers long. Analysing the whole road segment in such cases would be computationally demanding and analysing the accessibility of a building to a point on the road which is far away (e.g., 1 km) would not produce useful results. Instead, only the part of the closest road that is relevant to the local context of the core building is cut out using a custom SQL function *ST_Building_Crop_Road(*geometry *building_geom,* geometry *road_geom,* numeric *cut_distance)* which was created specifically for this task. This function first projects the core building's vertices onto the closest road. From these projected points, it then finds two points (p_1, p_2) that are closest to the ends of the road linestring. The function's *cut_distance* parameter specifies how far from these points the road should be cut to leave only the portion that is relevant to the core building. In this experiment we have used a constant *cut_distance* of 100 m (Figure 7.15). Each of the projected points p_1 and p_2 will be closer to one end of the road than the other projected point and this determines the direction for the interpolation in the next step. Then, two new points (p_1', p_2') that are *cut_distance* removed from the corresponding projected points are interpolated on the road in previously determined directions. These interpolated points will be the end points of the cut road.

TABLE 7.13: The overall results of binary classification of buildings into *accessible* and *inaccessible*. Null contains cases where the computation could not be performed.

Accessibility	Count	Ratio
Accessible	393,123	99.44%
Inaccessible	2,055	0.52%
Null	152	0.04%

7.3.4 Results

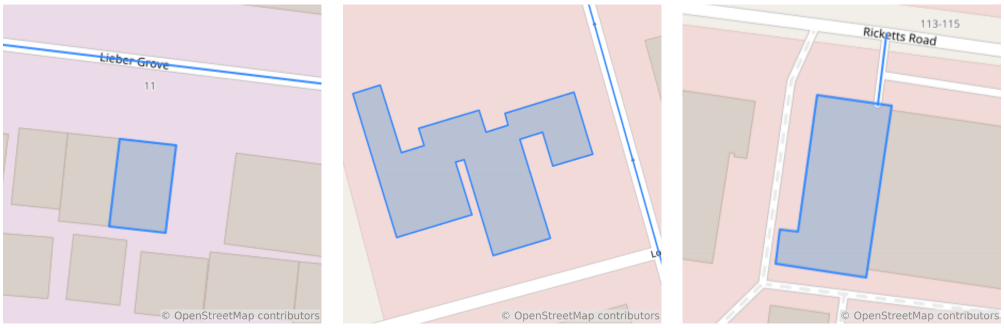
This section lays out the results of the case study where almost 400,000 buildings in Victoria were classified based on their accessibility from the road network into *accessible* and *inaccessible* buildings. Table 7.13 shows the overall results of the classification. *Accessible* buildings in the dataset far outnumber the *inaccessible* buildings as one would expect in a well covered, well mapped area of OSM. However, there are still more than 2,000 *inaccessible* buildings that potentially indicate erroneous data and need to be checked. There were also 152 buildings in Victoria for which the accessibility could not be computed due to problems such as invalid geometry of the core building, the closest road, or the occluder object.

Table 7.14 breaks down the *accessible* buildings based on the reason why they are classified as such. It is clear that for the vast majority of the buildings, the visibility analysis was needed to confirm that they are accessible from their closest roads. This table also shows the buildings that were classified as *accessible* by default, without the need to perform the full visibility. In 40,447 cases the buildings were classified as *accessible* because there were no occluder buildings in the surroundings of the core building, so there was nothing to obstruct the access to it. There were also 2,616 cases where the core building is intersecting its closest road which then means that the building can be accessed from the road.

Table 7.15 shows the results by the most frequent building tags in Victoria (i.e., building tags here correspond to building tags in Figure 7.14). It can be noticed that accessibility is very high for all building types, but this is expected given that more than 99% of all buildings in the dataset are *accessible* (Table 7.13). Although the ratios of *inaccessible* buildings across different types of buildings are low, they are the lowest in buildings where people live –

TABLE 7.14: Breakdown of *accessible* buildings based on the reason why they are classified as *accessible*.

Justification	Count	Ratio of all *accessible*
The accessibility analysis has confirmed the core building is accessible from the closest road (Figure 7.16a)	350,060	89.05%
No occluder buildings were found in the surroundings of the core building (Figure 7.16b)	40,447	10.29%
The distance between the core building and the closest road is 0 m (Figure 7.16c)	2,616	0.67%



(A) Accessibility analysis (B) No occluder buildings (C) $d_{CR} = 0m$

FIGURE 7.16: Examples of *accessible* buildings and different reasons for such classification.

namely *house* (0.10%), *residential* (0.34%), and *apartments* (0.78%). Buildings tagged as *roof* (3.18%) and *school*(2.87%) have the highest ratio of *inaccessible* buildings.

The frequencies of building tags that can be found in buildings that are classified as *inaccessible* are shown in the last column of Table 7.15. The frequencies of different building tags roughly follow the frequencies of tags for all buildings in the dataset (Figure 7.14). Once again, the majority of buildings (74%) are tagged with a generic *yes*. It can be seen that some of the most frequent tags found in *inaccessible* buildings are those that can be considered auxiliary to what are considered main buildings on the property. Buildings like *shed*, *roof*, *silo*, and *garage* are not always excepted to be accessible from the road and can often be accessible through main buildings on their properties.

The *inaccessible* buildings were filtered by selecting only buildings which are considered to be main buildings and hence expected to be accessible from the road network. Table 7.16 shows the two conditions used in this filtering process, where the conditions are connected with a logical OR operator. The first condition selects all buildings tagged with *house,*

TABLE 7.15: The classification results for most frequent building tags in the imported OSM dataset.

Building	*Accessible*	Ratio *accessible*	*Inaccessible*	Ratio *inaccessible*	Ratio of all *inaccessible*
Yes	239,209	98.63%	1,521	0.63%	74.01%
House	118,761	99.81%	122	0.10%	5.94%
Residential	11,170	99.62%	38	0.34%	1.85%
Shed	5,732	97.04%	103	1.74%	5.01%
Industrial	2,693	98.39%	34	1.24%	1.65%
Retail	2,564	98.88%	29	1.12%	1.41%
Commercial	2,347	99.11%	20	0.84%	0.97%
Apartments	2,156	99.17%	17	0.78%	0.83%
Roof	1,644	96.76%	54	3.18%	2.63%
School	1,517	96.87%	45	2.87%	2.19%
Garage	1,031	99.23%	8	0.77%	0.39%
Others (sum)	4,299	97.73%	64	1.45%	3.11%

TABLE 7.16: Criteria used to filter inaccessible buildings by removing small buildings and outbuildings from the results.

Condition 1	building IN ('house', 'school', 'residential', 'industrial', 'retail', 'commercial', 'apartments', 'university', 'train_station', 'hospital')
	OR
Condition 2	building = 'yes' AND area > 50

school, residential, industrial, retail, commercial, apartments, university, train_station, or *hospital*. The second condition selects buildings tagged with *yes*, but only if their areas are larger than $50m^2$.

After filtering, 1,585 buildings were identified in the dataset which according to their tags and size should be accessible from the road network, but are classified as *inaccessible*.

7.3.5 Online mapping challenge

The 1,585 buildings in Victoria that were classified as *inaccessible* (Table 7.17) have been published online for mappers from the OSM community to assess and fix any potential errors. The buildings were published together with the closest roads in the form of a *MapRoulette* challenge. *MapRoulette* is a micro-tasking platform for OSM. It presents mappers with small

TABLE 7.17: Filtered results for the most frequent buildings classified as inaccessible.

Building	*Inaccessible*	Ratio of all *inaccessible*
Yes	1,257	79.31%
House	122	7.70%
School	45	2.84%
Residential	38	2.40%
Industrial	34	2.15%
Retail	29	1.83%
Commercial	20	1.26%
Apartments	17	1.07%
University	11	0.69%
Train_station	7	0.44%
Hospital	5	0.32%

mapping tasks that can be completed in a short time and its goal is to improve the quality of OSM data. A *MapRoulette* challenge comprises of group of tasks dealing with the same problem. Multiple mappers can participate in the same challenge, and the progress of the challenge is tracked with various measures.

7.3.5.1 Mapping tasks

Each mapping task presented mappers with an *inaccessible* building and its closest road drawn over aerial imagery (Figure 7.19). The mappers can complete tasks in the following ways:

- Edit - the mappers can edit the OSM features presented in the task, or other features in that locality and mark the task with “I fixed it!”;
- Skip - the mappers can skip the current task and go to the next one (the skipped task will still be available to other mappers);
- Not an issue - if they believe that the problem presented in the task is not an issue in the OSM, the mappers can mark the task as such;
- Too hard / Can’t see - the mappers can mark the task if they are not able to complete it due to poor imagery or the difficulty of the task;

- Already fixed - the mappers can use this mark when they believe that the problem presented in the task has already been solved. This often happens when multiple inaccessible buildings share the same access road that is missing. When this road is added for one inaccessible building, other buildings fall into the already fixed category.

In the task description for this specific challenge, mappers were given the following list of possible problems in the OSM data that might cause the building in the task to be *inaccessible*, and how to fix them:

- missing road - map the missing roads around the building and mark the task as fixed, alternatively if the roads are not visible in the satellite imagery mark the task as *too hard* and *hard to see*
- outbuildings - if the building is tagged as building=yes change the tag to a more specific tag, such as building=shed and mark the task as fixed; otherwise, if the building is correctly tagged with something other than building=yes, and still does not have an access road, mark the task as *not a problem* and add MR tag 'small building'
- building inside building - if one of the features is incorrectly tagged as a building you could remove the building=* tag from it and mark the task as fixed; otherwise if both objects are buildings and you believe they are mapped correctly you could mark the task as not a problem and add MR tag 'building inside the building'
- building accessible from another highway - in this case the task can be marked as not a problem with MR tag 'another access'

7.3.5.2 Mapping progress and evaluation

This section shows the progress of the MapRoulette challenge until 28 May 2020. In two months that the challenge has been active, total of 577 tasks have been completed by mappers (Figure 7.18) which covers almost 65% of all identified *inaccessible* buildings in Victoria (Figure 7.17). Figure 7.17 shows the statistics of how mappers have completed these tasks. The majority of tasks have either been fixed (36% of all completed tasks) or marked as not an issue (53% of all completed tasks). There were much fewer tasks that were skipped, too hard, or already fixed.

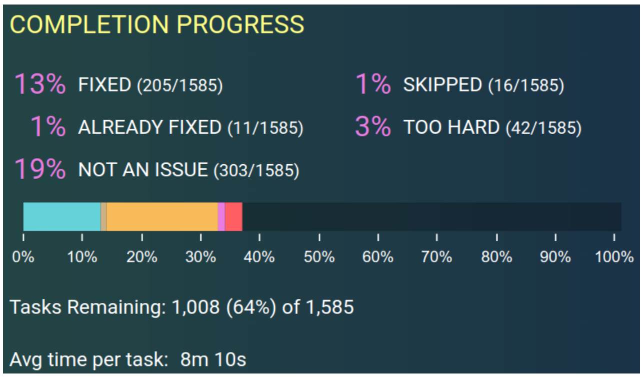


FIGURE 7.17: Statistics of the MapRoulette mapping challenge.

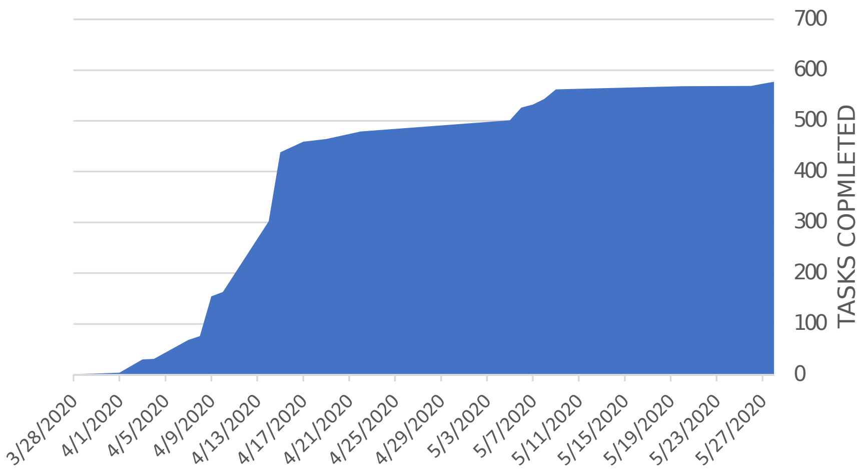


FIGURE 7.18: MapRoulette mapping challenge progress over time.

For the purposes of evaluation of the proposed method, the results of binary accessibility classification have been compared to the mappers responses in the mapping challenge. Buildings that were classified as *accessible*are considered *positives* and buildings classified as *inaccessible*are considered *negatives* in the contingency table (Table 7.18). The evaluation of *positives* was not possible in this study because buildings classified as *accessible* were not presented in the mapping challenge as they do not indicate potential errors. Thus, counts of *true positives and false positives* were not determined. The tasks that were completed as fixed can be seen as *true negatives* in the classification results because they imply that mappers have confirmed the presence of an error in data that was causing the building to be classified as *inaccessible*. Similarly, tasks marked as not an issue can be seen as *false negatives* in the classification results because they imply that the algorithm has incorrectly classified the building as *inaccessible*. However, we have observed that there have been cases where mappers have concluded that the building presented in the task is not expected to have access from the road based on their personal knowledge and experience of mapping

(A) Comment reports that building is a shed.

(B) Comment reports that building is within another building.

FIGURE 7.19: Examples where tasks are marked as not an issue, but comments indicate that issue exists.

TABLE 7.18: Binary classification contingency table.

		Mappers	
		Accessible	*Inaccessible*
Classifier	*Accessible*	TP	FP
	Inaccessible	247	261

practices in the OSM. In these cases, the proposed method has correctly identified that the building is *inaccessible* and the task should not contribute to the number of *false negatives*. This is also supported by our analysis of the comments that mappers have made when solving these tasks, where two predominant cases where this happens have been noticed. There are 6 tasks that were marked as not an issue, but the comment says that the *inaccessible* building is actually a shed that is tagged differently. Another 50 tasks marked as not an issue are commented as cases where the core building is located topologically within another building which always makes it completely occluded and thus *inaccessible*. Since both of these problems point to errors in OSM data that should be corrected, these tasks were counted towards *true negatives* in the classification evaluation shown in Table 7.18.

Finally, Figure 7.20 shows the distribution of tasks marked as fixed and not an issue across different types of buildings. It can be seen that residential buildings (i.e., tagged with house and residential) are among the most common where the problem of inaccessibility was correctly detected. In public buildings tagged with commercial, retail and school there is a much higher ratio of false negatives.

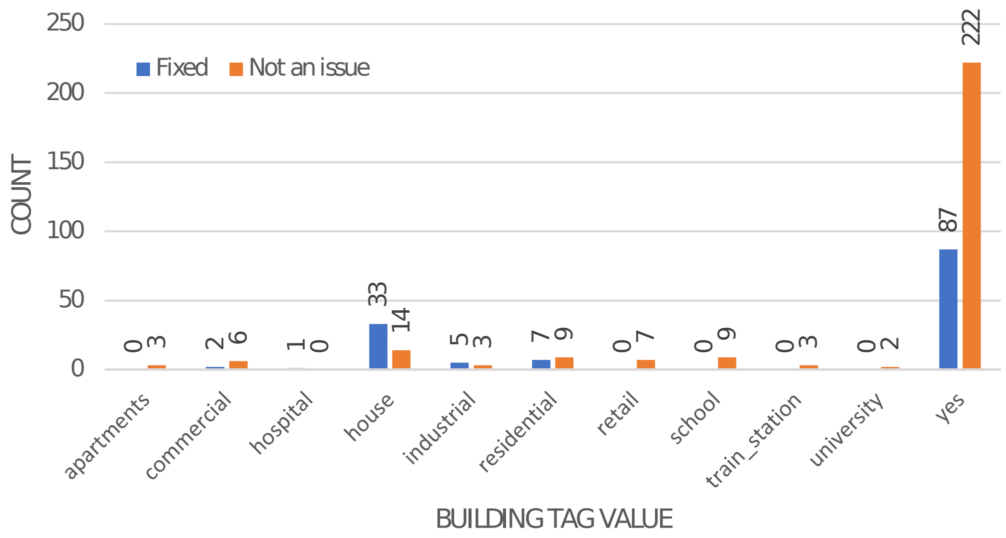


FIGURE 7.20: Distribution of fixed and not an issue MapRoulette tasks across different building tags.

7.3.6 Discussion

This case study has shown how the proposed building accessibility approach (Section 6.5) can be applied to detect inaccessible buildings and thus point to potential errors in the dataset. The fact that the vast majority of the buildings in the dataset were found to be accessible (Table 7.13) only justifies the assumption made in the proposed approach that the buildings should be accessible from the road network. In terms of patterns in the data, buildings that do not have access from the road network can be described as outliers. Given that the outlier detection is a method that is often used in data cleaning to indicate errors in data, it is reasonable to assume that in this case inaccessible buildings can serve as indicators of potential errors as well. The difference in this case study was that the pattern of majority buildings being accessible from the closest roads has not been learned from the data, but theoretically assumed and later confirmed by the analysis.

The online mapping challenge has been a great platform to present the detected potential errors to the OSM community so that they can be fixed. The response from the community has resulted in one third of detected inaccessible buildings to be assessed and potential errors corrected within 22 days. This highlights how useful the proposed method is when applied to improve map data. Even though Victoria can be considered a well mapped area in OSM where data are mature and have been repeatedly revised by active contributors to assure their quality, the proposed method has discovered errors that were previously unknown. This might be due to the fact that the accessibility analysis presented in Section 6.5 is a complex spatial analysis that simultaneously considers multiple spatial objects. Such task is not trivial and these errors might be hard to notice when a person is looking at the map with a naked eye. But when potential errors are detected automatically, mappers are able to direct their efforts towards correction of errors. This can save considerable time and resources in the data quality assurance process.

Apart from the obvious benefit of improving the quality of OSM data, the challenge has also provided valuable feedback from 18 different mappers. This provides the opportunity to gain a deeper understanding of specific OSM dataset for Victoria as well as the performance of the proposed approach. At first it may seem as though the majority of tasks addressed by mappers have been buildings incorrectly classified as inaccessible (Figure 7.17). However, a further analysis of the mappers' comments associated with those tasks show that a meaningful portion of these cases are indeed inaccessible buildings, but for which the mappers' believe it is acceptable to remain without access. These are exactly the kinds of buildings we have partially filtered out prior to the building accessibility analysis with conditions shown in Table 7.16. For example, buildings annotated as sheds have not been analysed in this case study because it is expected for sheds to often be located behind main buildings and be accessible indirectly through them. Furthermore, buildings tagged with the generic *building=yes* tag that are smaller than $50m^2$ have also been removed because they could be untagged sheds. We have instructed mappers to fix the tags of sheds that are annotated differently and have made it into the accessibility analysis because they are larger than $50m^2$. However, there were still cases where the mappers have recognised the sheds but have marked the task as not an issue instead of fixing the error (Figure 7.19a). Another example are buildings that are topologically positioned within other buildings (Figure 7.19b). Because the proposed approach is using topological relationships between the rays cast between the core building and the closest road and the occluder buildings, in these cases the occluder building containing the core building will intersect all rays. A potential solution for such situations may be to disregard those occluder buildings that contain both the core building and its closest road. An example where this often occurs are the shopping malls where individual shops are often mapped inside the outline of the whole complex. Topographically, schools, hospitals, and airports may also be composed from several features which are sometimes represented as individual buildings in OSM but have a joint function. This issue has been tackled in the

FIGURE 7.21: Buildings on the waterfront near Lakes Entrance that can only be accessed by boat.

field of map generalisation with methods proposed to detect such objects [112] and it may be useful to combine them with our approach in the future.

When the results of the online mapping challenge are adjusted for these cases the performance of the proposed approach improves as shown in Table 7.18. However, according to mappers there are still more false than true negative classification results. We suspect there are two major factors that affect this and that may further improve the performance of the proposed method. The first is the assumption that majority of buildings should have access from the road network. There are cases in Victoria where residential buildings are rightfully inaccessible from the road network. One example are the houses on the beach of Lakes Entrance that are accessible only by sea (Figure 7.21). The road network does not exist in that area and according to the proposed approach these buildings will always be inaccessible.

The other factor that affects the performance of the method is the way in which this assumption is realised in the proposed approach. We have only considered the building tags and areas of building objects. However, there are cases in OSM where the semantics of buildings are recorded with a combination of different tags. One such example are the silos that are usually annotated with *building* = *yes* and *man_made* = *silo*. Because they are usually larger than $50m^2$ they have not been filtered out of the analysis. Figure 7.22 shows a group of silos in a rural industrial area that are all classified as inaccessible. However, it is often not expected for them not to have a built access road or such road not to be mapped since it is on the private property.

FIGURE 7.22: Silos that are classified as *inaccessible*.

FIGURE 7.23: Building is accessible from the road on the right, but the algorithm has tested the accessibility from the road on the left as it is slightly closer to the building.

Lastly, it may also happen that the access to a building is provided by a road that is not the closest one. Although there have not been many such cases in the case study, Figure 7.23 shows one such example. The building is classified as inaccessible by the proposed approach based on the analysis of the road on the left. However, the building is actually accessible from the road on the right which is slightly further away. For situations like these it may be worth analysing the accessible from multiple closest road if the difference in distance the them is small. This would lower the number of buildings incorrectly classified as inaccessible, but would increase the time needed to perform the the analysis.

8

Discussion

This chapter discusses the outcomes and limitations of this thesis and reflects back on the research questions and hypothesis stated in Chapter 1. The first section discusses the modelling of ternary spatial relationships (Chapter 3), together with the proposed RTL and RIM models (Chapters 4 and 5). The second section discusses how data mining was applied with the proposed models to develop intrinsic error detection approaches in Chapter 6. The third section discusses the three case studies from Chapter 7 that demonstrate how the proposed error detection approaches can be used to detect errors in a map database. The last section briefly evaluates the main hypothesis of this thesis.

This chapter is based on published papers [1, 3, 4].

8.1 Modelling ternary spatial relationships

The aim of this thesis was to develop methods for automated detection of errors in map databases by utilising ternary spatial relation models. Thus, the first task in this thesis was to develop models that can adequately capture and express ternary spatial relationships. This is not a trivial task because the complexity of spatial relationships grows with the number of spatial objects that are included. For example, when describing the relationship between two objects - A and B, we could take the approach of describing the relationship of A towards B (e.g., A is within B) or B towards A (e.g., B contains A). This is straightforward and such binary relationships are sometimes even symmetrical, when the relationship is such that the order of objects does not matter (e.g., A touches B or A intersects B). However, when there are three spatial objects in a relationship, one cannot unambiguously capture the ternary scene using element-wise binary relationships as demonstrated in Chapter 3 (see Figures 3.1 and 3.2).

As a response, the concept of one *core* object being described relative to two *peripheral* objects was introduced in Chapter 3. This approach gives us a clear goal regarding which spatial object we want to characterise in a ternary relationship (i.e., the core object), and then use the remaining spatial objects (i.e., peripheral objects) in this relationships to aid us in this characterisation. Based on this concept, two ternary spatial relation models have been introduced - Refined topological relations for line objects (RTL, Chapter 4) and the Ray intersection model (RIM, Chapter 5).

RTL is a ternary topological relation model that models topological relations where three or more objects are intersecting each other (i.e., they share at least one point). RTL describes if the core line object is connected to other peripheral objects around it on one or both sides. This distinction is important in linear networks and their analyses (e.g., one-way streets in a road network).

RIM, on the other hand, is a spatial relation model because it can model objects that are topologically disjoint (i.e., they do not share any points). RIM describes the position of one core spatial object relative to two peripheral objects, even in situations where they are topologically disjoint. This relationship is captured with intersections that the core object forms with the rays cast between peripheral objects. The RIM model has multiple potential

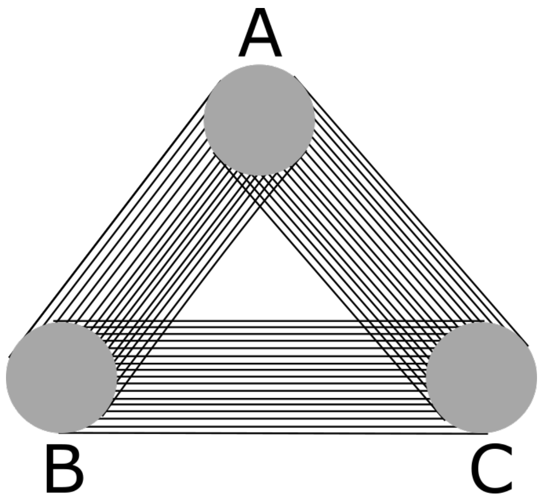


FIGURE 8.1: Rays cast between three peripheral objects may leave a hole in the ray area.

applications, but is particularly useful as a basis for reasoning about one spatial object being positioned between the other two.

Both RTL and RIM models are conceptually and computationally based on the binary 9IM, hence they are compared against the performance of the 9IM for the same tasks. In Chapters 4 and 5, it was demonstrated that both RTL and RIM can differentiate more configurations of three spatial objects than the 9IM. This was also confirmed in the first two case studies in Chapter 7, confirming that the proposed models better capture complex spatial relationships. Thus, the proposal of RTL and RIM models answers the first research question of this thesis – *How can ternary spatial relationships be modelled?*

There are a few important scope limitations to spatial relationship modelling addressed in this thesis. Firstly, the spatial relations considered in this thesis are limited to the maximum of three spatial objects. RIM requires a minimum of three spatial objects to express a relationship, while RTL describes binary spatial relationships as asymmetrical (i.e., present only on one side of the core line object). However, both models require exactly one core object and a maximum of two peripheral objects. There are cases, especially when analysing if one spatial object is between others, when consideration of more peripheral objects would be beneficial (e.g., is the core object positioned between three or more peripheral objects?). This would require a different definition of the space in between the peripheral objects and the ray casting approach of RIM would leave some gaps in the triangle (Figure 8.1). RIM relies on the analysis of the ray area (i.e., the are covered by rays cast between peripheral objects), and therefore would not include those gaps in the analysis.

The second limitation is that this thesis is constrained to the spatial objects in a 2D plane. This means that the geographical objects on a sphere or ellipsoid are not supported and map datasets need to be projected to a planar coordinate system for the RTL and RIM models to work. This limits their applicability to the reasoning on a global scale (e.g., is New York between Paris and Tokyo?), but does not affect their performance with map objects in a localised region such as a city, a country, or even a continent. The topological relation models in this thesis also do not support 3D spatial objects. Although majority of map databases are still based on 2D data, 3D objects are becoming more common and they will be particularly important in defining the vertical hierarchy of objects in a map (e.g., tunnels, bridges, and overpasses).

8.2 Data mining for intrinsic error detection

The two approaches for detection of errors in map databases proposed in Chapter 6 are based on the RTL and RIM models. These approaches are intrinsic and only need the map dataset itself to detect errors. This alleviates the problem of needing the ground truth, which is often not available for large regions covered by global map databases, to identify errors. The proposed error detection approaches are also based on the assumption of *mature spatial datasets* for which the majority of data are assumed to be of acceptable quality and the deviations from the pattern set by the majority of data are considered outliers and potential errors (Chapter 1). This enables the first approach – *discovery of topological constraints for classes of spatial objects* (Section 6.4) to use data mining methods such as frequent itemset mining to detect patterns of the majority of data, and to treat the outliers as potential errors. This approach mines the topological relations between a given class of objects and other objects, expressed with a topological relation model (e.g., RTL). The topological relations that occur most frequently (i.e., frequent items) have the highest support and can be adopted as topological constraints. Then, all objects from that class must satisfy the topological constraints to be considered valid and are otherwise labelled as potentially erroneous.

The second approach – *building accessibility analysis with RIM* (Section 6.5) is also an intrinsic approach that is based on the assumption of mature spatial datasets. However, the pattern of the majority of data in this approach is not learned with data mining, but it is theoretically grounded that buildings should have access from their closest roads. This assumption is later confirmed in the case study that has analysed the accessibility of buildings from the road network. To enable the building accessibility analysis, RIM had to be modified to test if the occluder object (a multipolygon geometry) is positioned between the core building and the road in such a way that completely occludes the visibility between them. If the building is not visible from the road, then it is assumed to be inaccessible and potentially erroneous. Such buildings later need to be checked by human operators, same as objects that do not satisfy the topological constraints defined with the first approach. These two error detection approaches address the second research question – *How can errors be intrinsically detected with the analysis of ternary spatial relations?* They also show that models for ternary spatial relationships (i.e., RTL and RIM) are useful and needed to capture and make use of complex interactions between objects in map datasets.

The limitation of these error detection approaches is that they still require considerable involvement of human experts. First, a good knowledge of the analysed dataset is required in order to anticipate which classes of objects are correlated and in which manner (e.g., there is a correlation between bridges and rivers). Then, human experts are needed to evaluate the potential errors that are brought up by the error detection approach. This means that the proposed approaches still cannot achieve fully automated (i.e., autonomous) error detection that takes any map dataset as an input and outputs potential errors. Yet, it is unclear whether such autonomous methods are possible or even desirable. The MapRoulette experiment shows that the automated error detection methods can find and bring attention to potential errors and that contributors will gladly correct errors if they exist. However, it also shows that automated methods can produce many false positives which can deter users from evaluating the potential errors. As more automated methods become available and implemented, we will get a clearer sense of how they are accepted by mappers and the community. I believe that the users' toleration for false positives will depend on how difficult the detected errors are to detect manually - i.e., if the method detects an error that is hard to notice manually, users will have more tolerance for false positives produced by the method.

8.3 Detection of errors in a map database

The proposed error detection methods have been tested in case studies on the OSM map database for the state of Victoria, Australia (Chapter 7). The first case study demonstrated how the RIM model can be used to analyse betweenness of buildings on a university campus. This is the first study to automatically capture and describe the betweenness of spatial objects in detail sufficient to classify them as between, debatably between, or not between other spatial objects. This approach could be easily transformed into an error detection approach for any constraints that demand one spatial object to be positioned between the other two, or constraints defining that there should exist an object between two specific spatial objects (e.g., there should be a fence between a high-speed highway and residential buildings).

The second case study has tested the approach for the discovery of topological constraints for the class of bridge objects. It has been shown that the proposed approach can successfully learn that a bridge object should meet a linestring objects on both of its ends and should cross an obstacle. The support for the topological constraint formed based on these relations is 75% which means that 25% of bridges in the database have been labelled as potentially erroneous. The evaluation of these results against manual classification of bridges has shown that the proposed approach was able to automatically classify bridges with 90% accuracy and 75% recall. We consider these results satisfactory for an automated method that will enable mapping experts to focus their error rectification efforts on the 25% of bridges labelled as potentially erroneous instead of all bridges in the database.

The third case study has tested the approach for the building accessibility analysis with RIM. The analysis of buildings in Victoria has shown that over 99% of buildings are accessible from their closest roads. This confirms the assumption that buildings should be accessible from the roads that are closest to them. However, given the large amount of buildings in the database, less than 1% of buildings that were detected as inaccessible amount to sizeable $1,585$ buildings. These buildings were then labelled as potentially erroneous and published in the form of a MapRoulette mapping challenge online where other mappers from the OSM community can contribute by correcting errors in OSM and providing feedback on the analysis. With the feedback on over 500 inaccessible buildings that were assessed by mappers in MapRoulette, there were 205 buildings in the OSM data that were fixed. The number of data that were classified as inaccessible by the algorithm but were marked as “not an issue” by mappers initially seemed high. However, after some of the mappers' comments were analysed, we concluded that 56 of those cases actually point to errors in the data, but were not fixed. After adjusting for those values, the evaluation of the proposed approach provides balance between the number of detected inaccessible buildings that point to actual problems in the data (i.e., 261), and the number of inaccessible buildings that mappers believed are acceptable to be inaccessible (i.e., 247).

These case studies have demonstrated how errors can be detected in the real world map database. They have also demonstrated two different approaches for evaluating the automated error detection – by manual evaluation of the detected errors and by feedback from mappers in the OSM community. Thus, these two case studies have addressed the third research question – *how can the potential errors be detected and evaluated in a prominent map database?*

The first limitation of these case studies is that they did not cover different geographical extents in order to test if the errors and patterns would change in different locations. It would also be beneficial to test the geographical extent for which a single set of constraints can be defined. For example, does it make sense to define topological constraints for a class of spatial objects on a level of a city, a state, a country, or globally? Also, does this differ for different object classes or for different types of constraints?

The second limitation of these case studies is that they use only one source of map data - OSM. Although we have taken care to develop generic approaches that are compatible with all map databases, they were only tested with OSM data. It is challenging to find other datasets that can match the completeness, the level of detail, and the heterogeneity in data quality that OSM offers, and that are also available to researchers.

8.4 Hypothesis evaluation

In Chapter 1, we hypothesised that *the analysis of ternary spatial relationships between map features enables the detection of otherwise undetectable errors in map data.* And that *this would result in an automated, intrinsic error detection method with formal grounding, eliminating the need for ground truth or reference data.* By successfully addressing all three research questions, we consider that the above stated hypothesis is confirmed.

9

Conclusions and future work

This chapter discusses the overall research presented in this thesis and highlights the key takeaways. It also gives the directions for the future work in the area of automated error detection and spatial relation modelling.

9.1 Conclusions

This thesis was first to apply ternary spatial relation models and data mining approaches to detect errors in map databases. It has demonstrated that complex spatial relationships often include more than two objects at a time and are not expressible with the currently available binary spatial relation models. The non-trivial task of modelling ternary spatial relations was achieved by differentiating core and peripheral objects in a relationship. This has enabled two novel ternary spatial relation models to be developed and capture relationships between map objects that we were previously not able to capture.

This additional information about how objects in map databases are related to each other has been especially helpful in the consequent data mining approaches. This thesis has also introduced an assumption of mature spatial datasets in which the majority of the data are of acceptable quality and fit for use. This assumption allows us to treat the patterns of the majority of the data as the norm to which all data should conform, and to use data mining techniques to detect outliers as potential errors. With the introduction of data mining to the field of spatial data quality, there are numerous opportunities to develop automated and streamlined approaches that save time and human resources. Another approach to automated error detection that was presented in this thesis is to ground the data constraints in theory and then confirm them with the statistical analysis of the data. Both approaches to defining constraints for error detection have shown good results in the case studies, and they both work without the need for ground truth.

The three case studies have confirmed that the developed error detection approaches are able to detect errors in one of the most prominent global map databases today - OSM. By capturing and expressing the relationships that we were previously not able to capture, we have detected errors that were previously not detectable, in an automated way. The error detection results have been manually evaluated by mapping experts and they confirm that errors were successfully detected that would otherwise likely go unnoticed.

9.2 Future work

This thesis ends by giving numerous directions where this research can be further improved. There are also some suggestions for further development of automated error detection in map databases beyond spatial relations.

9.2.1 Improvements to the topological constraint detection with the RTL

The RTL model offers an alternative to the 9IM in cases where symmetrical connectivity of lines is important, such as networks. These topological constraints are important, yet may not be sufficient to distinguish between classes of objects with functionally similar properties (hence the low precision in the second case study when the method is applied to the entire dataset). This is in particular true for classes of objects that belong to the same semantic hierarchy, i.e., present specialisations of a concept (e.g., bridge vs overpass vs elevated pathway). In future studies, we recommend to extend the approach for automated learning of topological constraints with additional geometrical, hierarchical [41] or even attribute properties, enabling to further differentiate classes of objects. Extensions of the proposed method are likely to be of particular interest to classes of objects that assure connectivity across the 3D dimension (i.e., tunnels and other crossings) and objects that assure transitions between environments (e.g., jetties and piers). Tunnels in particular would be a good class of objects to detect errors in with the RTL model because they are topologically similar to bridges. The challenge lies in defining which properties should be studied, as they may differ between different geometry types. For example, length, number of nodes, angles, and whether a line is open or closed might be relevant geometrical properties of line objects to consider.

Another consideration that would improve the results of the second case study is to utilise more of the semantic information that are available in OSM. For example, if we differentiate whether a polygon for the *within-polygon* relation is a forest, a lake, a river, a parking, or a built-up area, we might get different results in the frequent-itemset analysis. Similarly, semantic distinctions of linear objects when searching for omitted bridges may show that some objects such as power-lines should be excluded from the analysis because they often produce false positives. In addition to semantic and topological features, there are geometrical properties of bridges such as curvature (bridges are mostly straight) that would help distinguish them from other linear objects.

9.2.2 Improvements to the building accessibility analysis

There are improvements that can be made to the building accessibility analysis, but require extensive studies and additional data modelling. The first improvement that can be made is to analyse all classes of objects in the map database as potential occluders of building accessibility. In the case study in Chapter 7, only the buildings were considered as potential occluders while in reality there are many more types of spatial objects that can prevent access from the road to the building such as fences, walls, canals, trees. Some of these objects will be present in the map database, while others will not (e.g., topographical features of the terrain such as slope, depression, or rocks). Also, some objects that are in the map database may be positioned between the building and the closest road, but not occlude the access (e.g., grass, footpath, cycleway). The complexity of this issue grows with the number of different classes of spatial objects that are considered. To determine which objects should be considered as potential occluders of accessibility, a comprehensive study should be carried out and the results should be reviewed by mapping experts. Given the large number of different object classes, this study would also benefit from data mining and machine learning techniques to help automate the process. This would be especially beneficial in cases where potential occluder objects change with different locations and geographical extents of map databases.

The second improvement to the building accessibility analysis is to consider that access may sometimes be provided by the second or third closest road. Repeating the proposed approach n times for n nearest roads could potentially solve this problem, but would also increase the time needed for the algorithm to execute. The final improvement to the building accessibility analysis could be to consider other types of objects as occluder objects, other than buildings. There are cases when a wall or a fence is preventing the user to access the building directly from the road segment. These objects are sometimes even mapped in the OSM and could be added to the building accessibility analysis to improve the robustness of the approach.

However, these are considerations for real life deployments of the proposed approach, not to prove the usability of n-ary relationships and data mining for error detection.

9.2.3 Automated error detection not based on spatial relations

Map databases contain much more than objects' geometries that we can compute spatial relations on. Attribute data are already an important part of error detection approaches proposed in this thesis. However, they are not utilised to their full capacity and there are many more analyses possible to detect errors in attribute data. The future approaches could range from statistical approaches that detect column-based outliers (e.g., outlier in the height column) to machine-learning based imputation of attribute data to correct errors [95]. For example, we have made a preliminary study on undocumented attribute keys that contributors use in the OSM [5]. Since these keys are not documented in the OSM wiki, there is no way of being certain in the meaning of these attributes and they could be considered erroneous. Our approach computes the similarity between all attribute keys in OSM based on their values, co-occurrence with other attributes and geometry types. This enables us to find the most similar documented keys that can be used instead of the undocumented ones.

Apart from the objects' attributes, there are also objects' spatial characteristics that can be utilised for error detection. A combination of geometry, semantics, spatial relations, and non-spatial information about contributors has already been used to detect vandalism in OSM [113] - a specific kind of errors that would also benefit from more sophisticated spatial relationship models like the ones proposed in this thesis. Table 9.1 shows spatial characteristics that can be derived from objects' geometries. One way of dividing these spatial characteristics may be to see if they result from the geometry of a single object, or if they need multiple objects' geometries to be calculated. Here, betweenness (analysed in the first case study) is in the same category as topological relations which were analysed in the second case study, as they are qualitatively describing the relation of one spatial object to others (i.e., object is between or not in between). Additional work can be done to investigate how other qualitative and quantitative properties and relations can help detect errors in map databases. The prospects for the interactions of geometric, attribute and qualitative relationships in error detection are still wide open for future research.

TABLE 9.1: Spatial characteristics of spatial objects

	with self (properties)	**with other objects (relations)**
quantitative	length area perimeter number of nodes angularity compactness	distance intersection
qualitative	topology (e.g. self-intersection, closed line, holes in polygon) orientation	**topological relations** **betweenness** alignment

I believe that this thesis is only scratching the surface of the potential research in the field of automated error detection in map databases. With the rapid development of sensors, smart cities, and the internet of things, we will soon have all the ingredients for live maps that reflect the real world changes in real time. However, with more data collected, it will be even more challenging to efficiently and adequately process them to assure high data quality. This is why data mining and machine learning methods that can automate these processes need to be developed further, and potentially even combined with the artificial intelligence.

Bibliography

[1] Ivan Majic, Elham Naghizade, Stephan Winter, and Martin Tomko. Discovery of topological constraints on spatial object classes using a refined topological model. *Journal of Spatial Information Science*, 18(18):1–30, jun 2019. ISSN 1948660X. doi: 10.5311/JOSIS.2019.18.459. URL http://josis.org/index.php/josis/article/view/459.

[2] Ivan Majic and Elek Pafka. AwaP-IC—An Open-Source GIS Tool for Measuring Walkable Access. *Urban Science*, 3(2):48, apr 2019. ISSN 2413-8851. doi: 10.3390/urbansci3020048. URL https://www.mdpi.com/2413-8851/3/2/48.

[3] Ivan Majic, Elham Naghizade, Stephan Winter, and Martin Tomko. RIM: a ray intersection model for the analysis of the between relationship of spatial objects in a 2D plane. *International Journal of Geographical Information Science*, pages 1–26, 2020. ISSN 1365-8816. doi: 10.1080/13658816.2020.1778002. URL https://www.tandfonline.com/doi/full/10.1080/13658816.2020.1778002.

[4] Ivan Majic, Elham Naghizade, Stephan Winter, and Martin Tomko. There is no way! Ternary qualitative spatial reasoning for error detection in map data. *Transactions in GIS (accepted for publication)*, pages 1–20, 2021. doi: 10.1111/tgis.12765.

[5] Ivan Majic, Stephan Winter, and Martin Tomko. Finding Equivalent Keys in Openstreetmap: Semantic Similarity Computation Based on Extensional Definitions. In *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, GeoAI '17, pages 24–32, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5498-1. doi: 10.1145/3149808.3149813. URL http://doi.acm.org/10.1145/3149808.3149813.

[6] Jelena Gabela, Ivan Majic, Allison Kealy, Mark Hedley, and Shenghong Li. Robust vehicle localization and integrity monitoring based on spatial feature constrained PF. In *Proceedings of IEEE/ION PLANS 2020*, St. Louis, Missouri, sep 2020. URL https://www.ion.org/plans/abstracts.cfm?paperID=8276.

[7] Ivan Majic. Learning rules and detecting errors in OpenStreetMap data. Conference presentation at FOSS4G SOTM 2019, Wellington, NZ, 2019. URL https://youtu.be/SIH2_ZfQrX8.

[8] Ihab F. Ilyas and Xu Chu. Trends in Cleaning Relational Data: Consistency and Deduplication. *Foundations and Trends® in Databases*, 5(4):281–393, 2015. ISSN 1931-7883. doi: 10.1561/1900000045. URL http://www.nowpublishers.com/article/Details/DBS-045.

[9] Isabelle Bloch, Olivier Colliot, and R.M. Cesar. On the ternary spatial relation "Between". *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 36(2):312–327, apr 2006. ISSN 1083-4419. doi: 10.1109/TSMCB.2005.857095. URL http://ieeexplore.ieee.org/document/1605379/.

[10] Eliseo Clementini and Roland Billen. Modeling and computing ternary projective relations between regions. *IEEE Transactions on Knowledge and Data Engineering*, 18(6):799–814, jun 2006. ISSN 1041-4347. doi: 10.1109/TKDE.2006.102.

[11] ISO. ISO 19157:2013: Geographic information – Data quality. 2013.

[12] Christopher Barron, Pascal Neis, and Alexander Zipf. A Comprehensive Framework for Intrinsic OpenStreetMap Quality Analysis. *Transactions in GIS*, 18(6):877–895, dec 2014. ISSN 13611682. doi: 10.1111/tgis.12073. URL http://doi.wiley.com/10.1111/tgis.12073.

[13] Ziawasch Abedjan, Xu Chu, Dong Deng, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker, and Nan Tang. Detecting Data Errors : Where are we and what needs to be done ? *Proceedings of the VLDB Endowment*, 9(12):993–1004, 2016. ISSN 2150-8097. doi: 10.14778/2994509.2994518.

[14] Asher Moses. Google Maps error sparks international conflict, nov 2010. URL https://www.smh.com.au/technology/google-maps-error-sparks-international-conflict-20101108-17jl3.html.

[15] Apple Maps 'is life-threatening' to motorists lost in Australia heat, dec 2012. URL https://www.bbc.com/news/technology-20663447.

[16] Amanda Ripley. What happens when Google Maps gets it wrong - The Washington Post, mar 2010. URL https://www.washingtonpost.com/opinions/what-happens-when-google-loses-your-address-you-cease-to-exist/2020/03/09/b1885f28-622c-11ea-b3fc-7841686c5c57_story.html.

[17] Xu Chu, Ihab F Ilyas, Sanjay Krishnan, and Jiannan Wang. Data Cleaning : Overview and Emerging Challenges. In *SIGMOD '16 Proceedings of the 2016 International Conference on Management of Data*, pages 2201–2206. ACM Press, 2016. ISBN 9781450335317. doi: 10.1145/2882903.2912574.

[18] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection. *ACM Computing Surveys*, 41(3):1–58, jul 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL http://portal.acm.org/citation.cfm?doid=1541880.1541882https://dl.acm.org/doi/10.1145/1541880.1541882.

[19] Joris Maervoet, Celine Vens, Greet Vanden Berghe, Hendrik Blockeel, and Patrick De Causmaecker. Outlier detection in relational data: A case study in geographical information systems. *Expert Systems with Applications*, 39(5):4718–4728, apr 2012. ISSN 09574174. doi: 10.1016/j.eswa.2011.09.125. URL http://linkinghub.elsevier.com/retrieve/pii/S0957417411014485.

[20] Max J. Egenhofer and J Herring. Categorizing binary topological relations between regions, lines, and points in geographic databases. *The*, 9(July 2016):1–28, 1990. URL http://didattica.univaq.it/moodle/file.php/1508/9intReport_Egenhofer_1992.pdf.

[21] Wenzhong Shi, Peter F Fisher, and Michael F Goodchild, editors. *Spatial Data Quality*. Taylor & Francis, Abingdon, UK, 2002. ISBN 978-0-203-35583-1. doi: 10.4324/9780203303245. URL http://www.tandfebooks.com/action/showBook?doi=10.4324/9780203303245.

[22] Rodolphe Devillers and Robert Jeansoulin. *Fundamentals of spatial data quality*. Geographical information systems series. London ; Newport Beach, CA : © 2006 ISTE Ltd, 2006. doi: 10.1002/9780470612156.

[23] Stephen C Guptill and Joel L Morrison. *Elements of spatial data quality*. Elsevier, 1995. doi: 10.1016/C2009-0-14900-0. URL https://doi.org/10.1016/C2009-0-14900-0.

[24] H Veregin. Data quality parameters. In Paul A Longley, Michael F Goodchild, David J Maguire, and David W Rhind, editors, *Geographical information systems - principles and applications*, volume 1, chapter 12, pages 177–189. Wiley, New York, 2 edition, 1999. ISBN 0471321826.

[25] Yiliang Wan, Wenzhong Shi, Lipeng Gao, Pengfei Chen, and Yong Hua. A general framework for spatial data inspection and assessment. *Earth Science Informatics*, 8 (4):919–935, 2015. ISSN 18650481. doi: 10.1007/s12145-014-0196-9.

[26] Pepijn Van Oort. *Spatial data quality: from description to application*. PhD thesis, 2006. URL http://library.wur.nl/WebQuery/wdab/1788022.

[27] Mordechai (Muki) Haklay, Sofia Basiouka, Vyron Antoniou, and Aamer Ather. How Many Volunteers Does it Take to Map an Area Well? The Validity of Linus' Law to Volunteered Geographic Information. *The Cartographic Journal*, 47(4):315–322, 2010. ISSN 0008-7041. doi: 10.1179/000870410X12911304958827. URL http://www.tandfonline.com/doi/full/10.1179/000870410X12911304958827.

[28] Ahmed Loai Ali, Zoe Falomir, Falko Schmid, and Christian Freksa. Rule-guided human classification of volunteered geographic information. *ISPRS journal of photogrammetry and remote sensing*, 127:3–15, 2017. doi: 10.1016/j.isprsjprs.2016.06.003.

[29] Lívia Castro Degrossi, João Porto De Albuquerque, Roberto Dos Santos Rocha, and Alexander Zipf. A framework of quality assessment methods for crowdsourced geographic information: A systematic literature review. In *Proceedings of the International ISCRAM Conference*, pages 532–545, April 2017.

[30] Ahmed Loai Ali, Falko Schmid, Rami Al-Salman, and Tomi Kauppinen. Ambiguity and plausibility: managing classification quality in volunteered geographic information. In *Proceedings of the 22nd ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 143–152. ACM, 2014.

[31] Mohamed Bishr and Werner Kuhn. Trust and reputation models for quality assessment of human sensor observations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8116 LNCS:53–73, 2013. ISSN 03029743. doi: 10.1007/978-3-319-01790-7-4.

[32] Yuntian Brian Bai, Suqin Wu, Yongli Ren, Kevin Ong, Guenther Retscher, Allison Kealy, Martin Tomko, Mark Sanderson, Hongren Wu, and Kefei Zhang. A new approach for indoor customer tracking based on a single Wi-Fi connection. *IPIN 2014 - 2014 International Conference on Indoor Positioning and Indoor Navigation*, (January):239–245, 2015. doi: 10.1109/IPIN.2014.7275489.

[33] Andrea Ballatore and Michela Bertolotto. Personalizing maps. *Communications of the ACM*, 58(12):68–74, nov 2015. ISSN 0001-0782. doi: 10.1145/2756546.

[34] Michael Negnevitsky. *Artificial intelligence: A guide to intelligent systems, S*. Pearson Education Limited, Edinburgh Gate, 2 edition, 2005. ISBN 0321204662.

[35] Lukasz Ciszak. Application of clustering and association methods in data cleaning. *Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT 2008*, 3:97–103, 2008. ISSN 1896-7094. doi: 10.1109/IMCSIT.2008.4747224.

[36] Peng Gong and Lan Mu. Error Detection through Consistency Checking. *Geographic Information Sciences*, 6(2):188–193, 2000. ISSN 10824006. doi: 10.1080/10824000009480549. URL http://dx.doi.org/10.1080/10824000009480549.

[37] Sylvie Servigne, Thierry Ubeda, Alain Puricelli, and Robert Laurini. A methodology for spatial consistency improvement of geographic databases. *GeoInformatica*, 4(1): 7–34, 2000. ISSN 13846175. doi: 10.1023/A:1009824308542.

[38] M. Andrea Rodríguez, Nieves Brisaboa, Jazna Meza, and Miguel R. Luaces. Measuring consistency with respect to topological dependency constraints. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*, page 182, New York, New York, USA, 2010. ACM Press. ISBN 9781450304283. doi: 10.1145/1869790.1869818. URL http://portal.acm.org/citation.cfm?doid=1869790.1869818.

[39] Sylvain Vallires, Jean Brodeur, and Daniel Pilon. Spatial Integrity Constraints: A Tool for Improving the Internal Quality of Spatial Data. In Rodolphe Devillers and Robert Jeansoulin, editors, *Fundamentals of Spatial Data Quality*, pages 161–178. John Wiley & Sons, London, UK, 2010. ISBN 9781905209569. doi: 10.1002/9780470612156.ch9. URL http://doi.wiley.com/10.1002/9780470612156.ch9.

[40] Christoph Mülligann, Krzysztof Janowicz, Mao Ye, and Wang-Chien Lee. Analyzing the Spatial-Semantic Interaction of Points of Interest in Volunteered Geographic Information. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6899 LNCS, pages 350–370. 2011. ISBN 9783642231957. doi: 10.1007/978-3-642-23196-4_19. URL http://link.springer.com/10.1007/978-3-642-23196-4_19.

[41] Padraig Corcoran, Peter Mooney, and Michela Bertolotto. Spatial relations using high level concepts. *ISPRS international journal of geo-information*, 1(3):333–350, 2012. doi: 10.3390/ijgi1030333.

[42] Max J. Egenhofer and Robert D. Franzosa. Point-set topological spatial relations. *International journal of geographical information systems*, 5(2): 161–174, jan 1991. ISSN 0269-3798. doi: 10.1080/02693799108927841. URL https://doi.org/10.1080/02693799108927841http://www.tandfonline.com/doi/abs/10.1080/02693799108927841.

[43] Eliseo Clementini. A Conceptual Framework for Modelling Spatial Relations. *Information Technology And Control*, 48(1):5–17, mar 2019. ISSN 2335-884X. doi: 10.5755/j01.itc.48.1.22246. URL http://itc.ktu.lt/index.php/ITC/article/view/22246.

[44] A Galton. Lines of sight. In M Keane, P Cunningham, M Brady, and R Byrne, editors, *AI and Cognitive science*, pages 103–113. Dublin University Press, 1994.

[45] Juan Chen, Anthony G Cohn, Dayou Liu, Shengsheng Wang, Jihong Ouyang, and Qiangyuan Yu. A survey of qualitative spatial representations. *The Knowledge Engineering Review*, 30(1):106–136, jan 2015. ISSN 0269-8889. doi: 10.1017/S0269888913000350.

[46] D A Randell, Z Cui, and A Cohn. A spatial logic based on regions and connection. In B Nebel, C Rich, and W Swartout, editors, *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pages 165–176, San Mateo, California, 1992. Morgan Kaufmann. URL http://citeseer.ist.psu.edu/randell92spatial.html.

[47] Eliseo Clementini and Paolino Di Felice. A model for representing topological relationships between complex geometric features in spatial databases. *Information Sciences*, 90(1-4):121–136, 1996. ISSN 00200255. doi: 10.1016/0020-0255(95)00289-8.

[48] Eliseo Clementini, Paolino Felice, and Peter Oosterom. A small set of formal topological relationships suitable for end-user interaction. In *International Symposium on Spatial Databases*, pages 277–295. Springer, 1993. ISBN 978-3-540-56869-8. doi: 10.1007/3-540-56869-7_16. URL http://link.springer.com/10.1007/3-540-56869-7_16.

[49] Eliseo Clementini and Paolino Di Felice. A comparison of methods for representing topological relationships. *Information Sciences - Applications*, 3(3):149–178, 1995. ISSN 10690115. doi: 10.1016/1069-0115(94)00033-X.

[50] Jun Chen, Chengming Li, Zhilin Li, and Christopher Gold. A Voronoi-based 9-intersection model for spatial relations. *International Journal of Geographical Information Science*, 15(3):201–220, apr 2001. ISSN 1365-8816. doi: 10.1080/13658810151072831. URL http://www.tandfonline.com/doi/abs/10.1080/13658810151072831.

[51] Yohei Kurata. The 9 + -Intersection: A Universal Framework for Modeling Topological Relations. In Thomas J Cova, Harvey J Miller, Kate Beard, Andrew U Frank, and Michael F Goodchild, editors, *Geographic Information Science*, pages 181–198. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-87473-7. doi: 10.1007/978-3-540-87473-7_12. URL http://link.springer.com/10.1007/978-3-540-87473-7_12.

[52] Frank Dylla, Jae Hee Lee, Till Mossakowski, Thomas Schneider, André Van Delden, Jasper Van De Ven, and Diedrich Wolter. A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties. *ACM Computing Surveys*, 50 (1):7:1—-7:39, apr 2017. ISSN 0360-0300. doi: 10.1145/3038927.

[53] Alexander Scivos and Bernhard Nebel. The Finest of its Class: The Natural Point-Based Ternary Calculus $\mathcal{LR}$ for Qualitative Spatial Reasoning. In C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, and T. Barkowsky, editors, *Spatial Cognition IV. Reasoning, Action, Interaction. Spatial Cognition 2004. Lecture Notes in Computer Science*, pages 283–303. Springer, Berlin, Heidelberg, 2005. ISBN 978-3-540-25048-7. doi: 10.1007/978-3-540-32255-9_17. URL http://link.springer.com/10.1007/978-3-540-32255-9_17.

[54] Roland Billen and Eliseo Clementini. A Model for Ternary Projective Relations between Regions. In Elisa Bertino, Stavros Christodoulakis, Dimitris Plexousakis, Vassilis Christophides, Manolis Koubarakis, Klemens Böhm, and Elena Ferrari, editors, *Advances in Database Technology - EDBT 2004*, pages 310–328, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-24741-8.

[55] Roland Billen and Eliseo Clementini. Introducing a Reasoning System Based on Ternary Projective Relations. In *Developments in Spatial Data Handling*, pages 381–394, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-26772-0.

[56] Roland Billen and Eliseo Clementini. Semantics of Collinearity Among Regions. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3762 LNCS, pages 1066–1076. 2005. ISBN 3540297391. doi: 10.1007/11575863_129. URL http://link.springer.com/10.1007/11575863_129.

[57] Paolo Fogliaroni, Jan Oliver Wallgrün, Eliseo Clementini, Francesco Tarquini, and Diedrich Wolter. A Qualitative Approach to Localization and Navigation Based on Visibility Information. In K.S. Hornsby, C. Claramunt, M. Denis, and G. Ligozat, editors, *Spatial Information Theory. COSIT 2009. Lecture Notes in Computer Science*, volume 5756, pages 312–329. Springer, Berlin, Heidelberg,

2009. doi: 10.1007/978-3-642-03832-7_19. URL http://link.springer.com/10.1007/978-3-642-03832-7_19.

[58] David Randell and Mark Witkowski. Using occlusion calculi to interpret digital images. *Frontiers in Artificial Intelligence and Applications*, 141:432–436, 2006. ISSN 09226389.

[59] Christian Köhler. The Occlusion Calculus. In *Proceedings of the 1st Cognitive Vision Workshop*, pages 1–6, Zürich, Switzerland, 2002.

[60] David Randell, Mark Witkowski, and Murray Shanahan. From Images to Bodies: Modelling and Exploiting Spatial Occlusion and Motion Parallax. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'01, pages 57–63, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608125.

[61] Paulo E. Santos, V. Fenelon, and H. Dee. Notes on a qualitative theory of shadows. In *Cognitive Robotics 08 Workshop at ECAI, European conference on Artificial Intelligence*, pages 47–54, Patras, Greece, 2008. URL https://www.researchgate.net/publication/47783008_Notes_on_a_qualitative_theory_of_shadows.

[62] Paulo E Santos, Roberto Casati, Hannah M Dee, Carl Schultz, and Mehul Bhatt. Eclipse in Occlusion A perspectival mereotopological representation of celestial eclipses. In Bert Bredeweg, Kamal Kansou, and Matthew Klenk, editors, *Proceedings of the 29th International Workshop on Qualitative Reasoning. QR2016*, pages 93–100, New York, USA, 2016. URL https://ivi.fnwi.uva.nl/tcs/QRgroup/qr16/pdf/QR2016Proceedings.pdf.

[63] Francesco Tarquini, Giorgio De Felice, Paolo Fogliaroni, and Eliseo Clementini. A Qualitative Model for Visibility Relations. In J. Hertzberg, M. Beetz, and R. Englert, editors, *KI 2007: Advances in Artificial Intelligence. KI 2007. Lecture Notes in Computer Science*, volume 4667, pages 510–513. Springer, Berlin, Heidelberg, 2007. ISBN 9783540745648. doi: 10.1007/978-3-540-74565-5_52. URL http://link.springer.com/10.1007/978-3-540-74565-5_52.

[64] Prithwijit Guha, Amitabha Mukerjee, and K. S. Venkatesh. OCS-14: You can get occluded in fourteen ways. *IJCAI International Joint Conference on Artificial Intelligence*, (Figure 1):1665–1670, 2011. ISSN 10450823. doi: 10.5591/978-1-57735-516-8/IJCAI11-280.

[65] Gérard Ligozat and Paulo E. Santos. Spatial occlusion within an interval algebra. In *2015 AAAI Spring Symposium Series*, pages 103–106, 2015. ISBN 9781577357087. URL https://www.aaai.org/ocs/index.php/SSS/SSS15/paper/view/10276.

[66] Paulo E Santos, Gerard Ligozat, and Marjan Safi-Samghabad. An Occlusion Calculus Based on an Interval Algebra. In *4th Brazilian Conference on Intelligent Systems (BRACIS 2015)*, pages 128–133, Nadal, Brazil, nov 2015. IEEE. ISBN 978-1-5090-0016-6. doi: 10.1109/BRACIS.2015.12. URL http://ieeexplore.ieee.org/document/7424007/.

[67] Paolo Fogliaroni. Visibility Relations. In Shashi Shekhar and Hui Xiong, editors, *Encyclopedia of GIS*, volume 8, pages 1–8. Springer International Publishing, Cham, 2015. doi: 10.1007/978-3-319-23519-6_1541-1. URL http://link.springer.com/10.1007/978-3-319-23519-6_1541-1.

[68] Emo Welzl. Constructing the visibility graph for n-line segments in O(n^2) time. *Information Processing Letters*, 20(May):167–171, 1985.

[69] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. Visibility Graphs: Finding the Shortest Route. In *Computational Geometry Algorithms and Applications*, pages 323–333. 2008. ISBN 3-540-65620-0. URL http://link.springer.com/book/10.1007/978-3-540-77974-2/page/1.

[70] Michel Pocchiola and Gert Vegter. The visibility complex. *International Journal of Computational Geometry & Applications*, 06(03):279–308, sep 1996. ISSN 0218-1959. doi: 10.1142/S0218195996000204. URL https://www.worldscientific.com/doi/abs/10.1142/S0218195996000204.

[71] Stéphane Rivière. Dynamic visibility in polygonal scenes with the visibility complex. In *Proceedings of the thirteenth annual symposium on Computational geometry - SCG '97*, pages 421–423, New York, New York, USA, 1997. ACM Press. ISBN

0897918789. doi: 10.1145/262839.263057. URL http://portal.acm.org/citation.cfm?doid=262839.263057.

[72] Vlasios Kasapakis and Damianos Gavalas. Geolocative Raycasting for real-time buildings detection in pervasive games. In *2015 International Workshop on Network and Systems Support for Games (NetGames)*, pages 1–3. IEEE, dec 2015. ISBN 978-1-5090-0068-5. doi: 10.1109/NetGames.2015.7382999. URL http://ieeexplore.ieee.org/document/7382999/.

[73] Wolfgang Heidrich, Stefan Brabec, and Hans-Peter Seidel. Soft Shadow Maps for Linear Lights. In Bernard Péroche and Holly Rushmeier, editors, *Rendering Techniques 2000*, pages 269–280, Vienna, 2000. Springer Vienna. ISBN 978-3-7091-6303-0. doi: 10.1007/978-3-7091-6303-0_24. URL http://link.springer.com/10.1007/978-3-7091-6303-0_24.

[74] Jiří Bittner and Peter Wonka. Visibility in Computer Graphics. *Environment and Planning B: Planning and Design*, 30(5):729–755, oct 2003. ISSN 0265-8135. doi: 10.1068/b2957. URL http://journals.sagepub.com/doi/10.1068/b2957.

[75] Fernando Alonso. The benefits of building barrier-free: a contingent valuation of accessibility as an attribute of housing. *European Journal of Housing Policy*, 2(1): 25–44, jan 2002. ISSN 1461-6718. doi: 10.1080/14616710110120577. URL https://www.tandfonline.com/doi/full/10.1080/14616710110120577.

[76] J. Balado, L. Díaz-Vilariño, P. Arias, and M. Soilán. Automatic building accessibility diagnosis from point clouds. *Automation in Construction*, 82:103–111, oct 2017. ISSN 09265805. doi: 10.1016/j.autcon.2017.06.026. URL https://linkinghub.elsevier.com/retrieve/pii/S0926580517302029.

[77] Nikos Sakkas and Juan Pérez. Elaborating metrics for the accessibility of buildings. *Computers, Environment and Urban Systems*, 30(5):661–685, sep 2006. ISSN 01989715. doi: 10.1016/j.compenvurbsys.2005.06.002. URL https://linkinghub.elsevier.com/retrieve/pii/S0198971505000396.

[78] Suxia Liu and Xuan Zhu. An Integrated GIS Approach to Accessibility Analysis. *Transactions in GIS*, 8(1):45–62, jan 2004. ISSN 1361-1682. doi: 10.1111/

j.1467-9671.2004.00167.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9671.2004.00167.x.

[79] Saad Jubran H. AlKahtani, Jianhong (Cecilia) Xia, Bert Veenendaaland, Craig Caulfield, and Michael Hughes. Building a conceptual framework for determining individual differences of accessibility to tourist attractions. *Tourism Management Perspectives*, 16:28–42, oct 2015. ISSN 22119736. doi: 10.1016/j.tmp.2015.05.002. URL https://linkinghub.elsevier.com/retrieve/pii/S2211973615000434.

[80] Alistair Ford, Stuart Barr, Richard Dawson, and Philip James. Transport Accessibility Analysis Using GIS: Assessing Sustainable Transport in London. *ISPRS International Journal of Geo-Information*, 4(1):124–149, jan 2015. ISSN 2220-9964. doi: 10.3390/ijgi4010124. URL http://www.mdpi.com/2220-9964/4/1/124.

[81] Wei Luo and Fahui Wang. Measures of Spatial Accessibility to Health Care in a GIS Environment: Synthesis and a Case Study in the Chicago Region. *Environment and Planning B: Planning and Design*, 30(6):865–884, dec 2003. ISSN 0265-8135. doi: 10.1068/b29120. URL http://journals.sagepub.com/doi/10.1068/b29120.

[82] Gavin Tansley, Nadine Schuurman, Ofer Amram, and Natalie Yanchar. Spatial Access to Emergency Services in Low- and Middle-Income Countries: A GIS-Based Analysis. *PLOS ONE*, 10(11):e0141113, nov 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0141113. URL http://dx.plos.org/10.1371/journal.pone.0141113.

[83] Michael F. Goodchild. Citizens as sensors: The world of volunteered geography. *GeoJournal*, 69(4):211–221, nov 2007. ISSN 03432521. doi: 10.1007/s10708-007-9111-y.

[84] Stephen Maguire and Martin Tomko. Ripe for the picking? Dataset maturity assessment based on temporal dynamics of feature definitions. *International Journal of Geographical Information Science*, 31(7):1334–1358, jul 2017. ISSN 1365-8816. doi: 10.1080/13658816.2017.1287370. URL https://www.tandfonline.com/doi/full/10.1080/13658816.2017.1287370.

[85] Mordechai Haklay. How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets. *Environment and*

Planning B: Planning and Design, 37(4):682–703, 2010. ISSN 14723417. doi: 10.1068/b35097.

[86] Jean Francois Girres and Guillaume Touya. Quality Assessment of the French OpenStreetMap Dataset. *Transactions in GIS*, 14(4):435–459, 2010. ISSN 13611682. doi: 10.1111/j.1467-9671.2010.01203.x.

[87] Dennis Zielstra and Alexander Zipf. A Comparative Study of Proprietary Geodata and Volunteered Geographic Information for Germany. In *13th AGILE International Conference on Geographic Information Science*, volume 1, pages 1–15, 2010. doi: 10.1119/1.1736005. URL http://koenigstuhl.geog.uni-heidelberg.de/publications/2010/Zielstra/AGILE2010_Zielstra_Zipf_final5.pdf.

[88] Christopher Barrington-Leigh and Adam Millard-Ball. The world's user-generated road map is more than 80% complete. *PLOS ONE*, 12(8):e0180698, aug 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0180698. URL http://dx.doi.org/10.1371/journal.pone.0180698https://dx.plos.org/10.1371/journal.pone.0180698.

[89] Anita Graser, Markus Straub, and Melitta Dragaschnig. Is OSM Good Enough for Vehicle Routing? A Study Comparing Street Networks in Vienna. In Georg Gartner and Haosheng Huang, editors, *Progress in Location-Based Services 2014, Lecture Notes in Geoinformation and Cartography*, Lecture Notes in Geoinformation and Cartography, pages 3–17. Springer International Publishing Switzerland, 2015. ISBN 9783642342028. doi: 10.1007/978-3-319-11879-6_1. URL http://link.springer.com/10.1007/978-3-319-11879-6_1.

[90] Xiang Zhang and Tinghua Ai. How to Model Roads in OpenStreetMap? A Method for Evaluating the Fitness-for-Use of the Network for Navigation. In F. Harvey and Y. Leung, editors, *Advances in Spatial Data Handling and Analysis, Advances in Geographic Information Science*, volume 19, pages 143–162. Springer International Publishing Switzerland, 2015. ISBN 9783319199498. doi: 10.1007/978-3-319-19950-4_9. URL http://link.springer.com/10.1007/978-3-319-19950-4_9.

[91] Fabian Müller, Ionuț Iosifescu, and Lorenz Hurni. Assessment and Visualization of OSM Building Footprint Quality. *International Cartographic Conference*, 2015.

[92] Guillaume Touya and Carmen Brando-Escobar. Detecting Level-of-Detail Inconsistencies in Volunteered Geographic Information Data Sets. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 48(2):134–143, jun 2013. ISSN 0317-7173. doi: 10.3138/carto.48.2.1836. URL https://utpjournals.press/doi/10.3138/carto.48.2.1836.

[93] Simon Gröchenig, Richard Brunauer, and Karl Rehrl. Digging into the history of VGI data-sets: results from a worldwide study on OpenStreetMap mapping activity. *Journal of Location Based Services*, 8(3):198–210, 2014. ISSN 17489733. doi: 10.1080/17489725.2014.978403.

[94] Musfira Jilani, Padraig Corcoran, and Michela Bertolotto. Automated highway tag assessment of OpenStreetMap road networks. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL '14*, volume 04-07-Nove of *SIGSPATIAL '14*, pages 449–452, New York, New York, USA, 2014. ACM Press. ISBN 9781450331319. doi: 10.1145/2666310.2666476. URL http://dl.acm.org/citation.cfm?doid=2666310.2666476.

[95] Rajesh Chittor Sundaram, Elham Naghizade, Renata Borovica-Gajic, and Martin Tomko. Harnessing spatio-temporal patterns in data for nominal attribute imputation. *Transactions in GIS*, (00):1–32, mar 2020. ISSN 13611682. doi: 10.1111/tgis.12617. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12617http://doi.wiley.com/10.1111/tgis.12617.

[96] OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture. Technical report, Open Geospatial Consortium Inc., 2011. URL http://portal.opengeospatial.org/files/?artifact_id=25355.

[97] Daniel Hernández. Reasoning with qualitative representations. In Daniel Hernández, editor, *Qualitative Representation of Spatial Knowledge*, volume 804 of *Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*, chapter 5, pages 55–103. Springer-Verlag, Berlin/Heidelberg, 1994. doi: 10.1007/BFb0020333. URL http://link.springer.com/10.1007/BFb0020333.

[98] Eliseo Clementini. Directional relations and frames of reference. *GeoInformatica*, 17(2):235–255, apr 2013. ISSN 1384-6175. doi: 10.1007/s10707-011-0147-2. URL http://link.springer.com/10.1007/s10707-011-0147-2.

[99] Yohei Kurata and Max J Egenhofer. The 9 + -intersection for topological relations between a directed line segment and a region. In B Gottfried, editor, *Workshop on behaviour monitoring and interpretation. BMI '07*, number 42, pages 62–76. TZI-Berichte, 2007.

[100] Joshua A. Lewis and Max J. Egenhofer. Oriented Regions for Linearly Conceptualized Features. In Matt Duckham, Edzer Pebesma, Kathleen Stewart, and Andrew U. Frank, editors, *Geographic Information Science*, pages 333–348, Vienna, Austria, 2014. Springer, Cham. ISBN 978-3-319-11593-1; 978-3-319-11592-4. doi: 10.1007/978-3-319-11593-1_22.

[101] Markus Schneider and Thomas Behr. Topological relationships between complex spatial objects. *ACM Transactions on Database Systems*, 31(1):39–81, mar 2006. ISSN 03625915. doi: 10.1145/1132863.1132865. URL http://portal.acm.org/citation.cfm?doid=1132863.1132865.

[102] Marco Manetti. *Topological Structures*, pages 39–62. Springer International Publishing, Cham, 2015. ISBN 978-3-319-16958-3. doi: 10.1007/978-3-319-16958-3_3. URL https://doi.org/10.1007/978-3-319-16958-3_3.

[103] D Sheeren, S Mustière, and J.-D. Zucker. A data-mining approach for assessing consistency between multiple representations in spatial databases. *International Journal of Geographical Information Science*, 23(8):961–992, aug 2009. ISSN 1365-8816. doi: 10.1080/13658810701791949.

[104] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, jul 2009. ISSN 03064573. doi: 10.1016/j.ipm.2009.03.002. URL https://linkinghub.elsevier.com/retrieve/pii/S0306457309000259.

[105] A. Inkeri Agrawal, Rakesh and Mannila, Heikki and Srikant, Ramakrishnan and Toivonen, Hannu and Verkamo. Fast discovery of association rules. In Ramasamy Fayyad, Usama M. and Piatetsky-Shapiro, Gregory and Smyth, Padhraic and Uthurusamy, editor, *Advances in knowledge discovery and data mining*, volume 12, pages 307–328. American Association for Artificial Intelligence, 1996. ISBN 0262560976. URL http://www.cs.helsinki.fi/hannu.toivonen/pubs/advances.pdf.

[106] Stephan Mäs. Reasoning on Spatial Semantic Integrity Constraints. In Stephan Winter, Matt Duckham, Lars Kulik, and Ben Kuipers, editors, *Cosit 2007*, pages 285–302, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 3-540-74786-9, 978-3-540-74786-4. doi: 10.1007/978-3-540-74788-8_18.

[107] Francesco Tarquini and Eliseo Clementini. Spatial Relations between Classes as Integrity Constraints. *Transactions in GIS*, 12(SUPPL. 1):45–57, dec 2008. ISSN 13611682. doi: 10.1111/j.1467-9671.2008.01134.x. URL http://doi.wiley.com/10.1111/j.1467-9671.2008.01134.x.

[108] A Vandecasteele and R Devillers. IMPROVING VOLUNTEERED GEOGRAPHIC DATA QUALITY USING SEMANTIC SIMILARITY MEASUREMENTS. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-2/W1(May 2013):143–148, may 2013. ISSN 1682-1777. doi: 10.5194/isprsarchives-XL-2-W1-143-2013. URL https://doi.org/10.5194/isprsarchives-XL-2-W1-143-2013http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-2-W1/143/2013/.

[109] Jiawei Han, Yandong Cai, and Nick Cercone. Knowledge Discovery in Databases: An Attribute-Oriented Approach. In *Proceedings of the 18th International Conference on Very Large Data Bases*, pages 547 – 559. Morgan Kaufmann Publishers Inc., 1992.

[110] Christian Borgelt. Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):437–456, nov 2012. ISSN 19424787. doi: 10.1002/widm.1074. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1074http://doi.wiley.com/10.1002/widm.1074.

[111] Jordan Wood. Filter and Refine Strategy. In S. Shekhar and H. Xiong, editors, *Encyclopedia of GIS*, pages 1–1. Springer, Boston, MA, Boston, MA, 2016. ISBN 978-0-387-35973-1. doi: 10.1007/978-3-319-23519-6_415-2. URL http://link.springer.com/10.1007/978-3-319-23519-6_415-2.

[112] Omair Z. Chaudhry, William A. Mackaness, and Nicolas Regnauld. A functional perspective on map generalisation. *Computers, Environment and Urban Systems*, 33(5):349–362, sep 2009. ISSN 01989715. doi: 10.1016/j.compenvurbsys.2009.07.002. URL https://linkinghub.elsevier.com/retrieve/pii/S0198971509000490.

[113] Quy Truong, Guillaume Touya, and Cyril Runz. OSMWatchman: Learning How to Detect Vandalized Contributions in OSM Using a Random Forest Classifier. *ISPRS International Journal of Geo-Information*, 9(9):504, aug 2020. ISSN 2220-9964. doi: 10.3390/ijgi9090504. URL https://www.mdpi.com/2220-9964/9/9/504.