Distortion-Aware Spatial Data Science

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Arun Sharma

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Shashi Shekhar

August, 2025

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Shashi Shekhar, to whom I owe an immeasurable debt of appreciation. His steadfast support, intellectual guidance, and genuine mentorship have been the foundation of my academic development. Throughout my doctoral journey, he has consistently offered insightful technical feedback, encouraged rigorous thinking, and emphasized the importance of effectively communicating research to broader scientific audiences.

I am also sincerely thankful to the members of my thesis committee, Prof. Vipin Kumar, Prof. Ravi Janardan, and Prof. Ying Song, for their thoughtful feedback, constructive criticism, and valuable suggestions which significantly enhanced the quality of my thesis.

This work would not have come to fruition without the unwavering support, collaboration, and friendship: Mohit Bhatia (San), Kshitij Bhai (Dhasu Sir), Ashok Bhai (Sharmaji), Jayant Bhai (Raja), Bhushan Sir, Chinmay Pandey (ACP Pradyuman), Abhishek Chakrabarty, Prashant Agarwal, Aman Malker, Udeept Prakash, Gaurav Tiwari, Anurag Rai (Lal Badshah), Alok Srivastava (Bihari), Chetan Kaushik (Chetu), Ayush Raghuvanshi (Ballu), Ayon Dey (Bangaali), Himanshu Chauhan (Simon), Hemant Kumar (42), Prashant Singh (Dhruv Rathee), Majid Farhadloo, Mingzhou Yang, Subhankar Ghosh, Ruolei Zeng, Shuai An, and Shengya Zhang. Each one of them contributed meaningfully at different phases of my doctoral studies.

I remain indebted to my Master's advisor, Professor Varun Chandola at the University at Buffalo, who introduced me to the field and encouraged my initial explorations into research. I am equally grateful to Professors Sargur N. Srihari and Steve Y. Ko for their continued belief in my potential and their generous recommendation letters, which opened several opportunities for me.

A heartfelt thank you also goes to my primary care physician, Dr. JSK Smith/Roth, whose exceptional medical care played a critical role in supporting my well-being and enabling me to perform at my best during the final year of my Ph.D.

Lastly, I am especially grateful to Kim Koffolt, whose meticulous proofreading, thoughtful critiques, and consistent editorial support have not only refined every draft of this thesis but also significantly shaped my ability to communicate research with clarity and precision.

In closing, I am deeply grateful to all who have supported my academic and personal growth throughout this challenging yet profoundly rewarding journey. Their collective encouragement, wisdom, and kindness have been instrumental in the successful completion of this thesis and will continue to inspire my future endeavors.

Dedication

To my grandparents, parents, elder brother, and sister-in-law.

Abstract

Location data on mobile devices generates hundreds of billions of dollars in revenue annually, with applications in many sectors, including energy, health, and retail. The world's economy also heavily relies on location and time, as this data is essential to applications in banking, air travel, law enforcement, emergency services, and telecommunications. While the knowledge derived from spatial data provides essential context for understanding and interpreting spatial patterns of events and objects across space and time data, many scientific domains ignore or remain unaware of spatial data's *special* nature. First, data driven methods largely adopt a one-size-fits-all (OSFA) approach, where traditional data mining and machine learning methods are trained without accounting for the geographic properties that give rise to diverse geophysical and cultural phenomena (e.g., similar to choosing a general practitioner over a specialist when treating a complex medical condition). Moreover, spatiotemporal data samples do not follow a uniform distribution across all regions and periods; instead, different geographic areas and temporal intervals exhibit distinct distributions. While numerous spatial data mining and statistical methods exist to address these issues, they tend to underperform when there is insufficient training data or missing values. This work explores methods to handle missing or otherwise distorted spatiotemporal data as these relate to possible adversarial behavior.

The thesis is inspired by a metaphor comparing such data to a mirror: *Data alone is insufficient to distinguish between a distorted reflection of a perfect world, a perfect reflection of a distorted world, or a combination of both.* Here, the distorted reflection is examined in the context of emerging trajectory big data. Even though moving objects (e.g., ships) are required to report their geo-locations periodically, some may *cover their tracks* by not reporting geo-locations, creating a data distortion. This thesis formalizes an approach to these challenges by first creating a taxonomy that illustrates the spectrum of data distortion challenges in trajectory analysis, ranging from data removal to noise addition and outright falsification. In addition, resilient algorithms are proposed to handle such distortions, ensuring accurate detection of patterns like rendezvous while paving the way for future research to counter even more sophisticated adversarial behaviors.

Contents

Acknowledgements **i**

Dedication **iii**

Abstract **iv**

List of Tables **ix**

List of Figures **x**

1 Introduction **1**

1.1 What are Data Distortions? 2

1.2 Data Distortions in Spatial Data Science 4

1.3 Scope and Terminology 7

1.4 Contributions and Thesis Organization 10

2 Investigating Possible Rendezvous Regions: A Time Slicing Approach 15

2.1 Introduction 15

2.2 Basic Concepts and Problem Statement 18

2.2.1 Basic Concepts 18

2.2.2 Problem Formulation 19

2.3 Related Work and Limitations 19

2.4 Approach 21

2.4.1 Framework 21

2.5 Necessary and Sufficient Condition of Spatiotemporal Intersection 21

2.5.1 Baseline Algorithm . . . 23
2.5.2 Constructing a Minimum Orthogonal Bounding Region and Spatiotemporal Intersection . . . 25
2.5.3 Spatio-temporal Grid Traversal Algorithm . . . 26
2.6 Validation . . . 29
2.6.1 Experiment Design . . . 29
2.6.2 Experimental Results . . . 30
2.6.3 Interpretation of Experimental Results . . . 31
2.6.4 Case Study on Real Automatic Identification System (AIS) data . . . 32
2.7 Conclusion and Future Work . . . 33

3 Analyzing Trajectory Gaps to Find Possible Rendezvous Region 35
3.1 Introduction . . . 35
3.2 Related Work and Limitations . . . 39
3.3 Possible Rendezvous Detection Problem . . . 40
3.3.1 Framework . . . 40
3.3.2 Basic Concepts . . . 41
3.3.3 Problem Formulation . . . 42
3.4 Baseline Approach . . . 43
3.4.1 MOBR Construction and Spatiotemporal Intersection Criteria . . . 43
3.4.2 Extracting Potential Intersecting Gap Pairs . . . 45
3.4.3 Linear Temporal Scan Algorithm . . . 45
3.5 Proposed Approach . . . 49
3.5.1 Time Prioritized based Temporal Selection Pruning . . . 49
3.5.2 Spatial Filter for Temporal Selection Search Algorithm . . . 51
3.5.3 Temporal Selection Search Algorithm . . . 54
3.6 Theoretical Evaluation . . . 56
3.6.1 Correctness and Completeness . . . 58
3.6.2 Asymptotic Time Complexity Analysis . . . 59
3.7 Experimental Evaluation . . . 59
3.7.1 Experiment Design . . . 59
3.7.2 Experiment Results . . . 61

3.8 Conclusion and Future Work 65

4 Abnormal Trajectory-Gap Detection: A Physics-based Approach 66

4.1 Introduction 66
4.2 Related Work 70
4.3 Problem Definition 70
4.3.1 Basic Concepts 70
4.3.2 Problem Formulation 71
4.4 Proposed Approach 72
4.5 Execution Trace of AGD Algorithm 74
4.6 Execution Trace of Memo-AGD Algorithm 74
4.7 Asymptotic Time Complexity Analysis 75
4.8 Experimental Validation 76
4.9 Conclusion and Future Work 79

5 A Physics-based Abnormal Trajectory Gap Detection 80

5.1 Introduction 80
5.2 Abnormal Trajectory Gap Detection Problem 82
5.2.1 Basic Concept 82
5.2.2 Framework 86
5.2.3 Problem Formulation 87
5.2.4 The Choice of Problem Formulation 88
5.3 Related Work 89
5.4 Baseline Approach 91
5.5 Proposed Approach 93
5.5.1 Spatial-Time Aware Gap Detection (STAGD) 93
5.5.2 Dynamic Region Merge (DRM) Approach 99
5.6 Theoretical Evaluation 104
5.6.1 Correctness and Completeness 105
5.6.2 Asymptotic Time Complexity 106
5.7 Experimental Validation 107
5.7.1 Experiment Design 107

5.7.2 Experiment Results: . . . 109
5.7.3 Case Study . . . 114
5.8 Discussion . . . 114
5.9 Conclusion and Future Work . . . 116

6 Towards a Tighter Bound on Possible-Rendezvous Areas: Preliminary Results 118
6.1 Introduction . . . 118
6.2 Problem Formulation . . . 121
6.2.1 Framework . . . 121
6.2.2 Basic Concepts . . . 122
6.2.3 Problem Formulation . . . 123
6.3 Time Slicing Model . . . 123
6.4 Proposed Approach . . . 127
6.4.1 Constructing Spatial Sub-networks . . . 128
6.4.2 Time Slicing Gap-Aware Rendezvous Detection Algorithm (TGARD) 131
6.4.3 A Dual Convergence Approach (DC-TGARD) . . . 133
6.5 Other Related Work . . . 136
6.6 Theoretical Evaluation . . . 136
6.6.1 Correctness and Completeness . . . 136
6.6.2 Asymptotic Analysis . . . 137
6.7 Experimental Evaluation . . . 138
6.7.1 Solution Quality . . . 139
6.7.2 Computational Efficiency . . . 141
6.8 Conclusion and Future Work . . . 144

7 Conclusion and Future Directions 145
7.1 Short-Term Future Directions . . . 146
7.2 Long-Term Future Directions . . . 149

References 154

List of Tables

1.1 Thesis contribution taxonomy. 12
3.1 Application Domain and Use Cases Examples for Detecting Rendezvous Regions 37
7.1 Key Contributions and Future Directions 146

List of Figures

1.1 Illustrative examples of illicit activities in maritime space. 4
1.2 Space-Time Prism interpolation may find patterns missed by linear interpolation 5
1.3 Data Distortion Taxonomy 6
1.4 The proposed use-inspired framework for denial-based behavior 10
1.5 The overall framework for denial-based behavior 11
2.1 An illustration of rendezvous region detection (Best in color). 16
2.2 Framework for detecting possible rendezvous regions to reduce manual inspection by analyst 22
2.3 Plane sweep execution trace with a test case 24
2.4 Cone Intersection of Effective Missing Period (EMPs) 25
2.5 Execution Trace of SGT and SGT-ST 29
2.6 Effects on area-pruning effectiveness (APE) 30
2.7 Spatio-temporal intersections detected in Bering Sea (best in color). The background imagery is not taken at the same time when the vessels were traveling in January 2014. 33
2.8 A Comparison of Linear vs Space-Time Interpolation vs Time Slicing Approach 34
3.1 Time-slicing approach shrinks the area of possible rendezvous (Best in Color) 36
3.2 Framework for detecting possible rendezvous regions to reduce manual inspection by analyst 41
3.3 Creating a minimum orthogonal bounding rectangle from two cones 43
3.4 Cone Intersection of Two Effective Missing Period (EMPs) 44
3.5 Linear Temporal Scan (LTS) Algorithm 46

3.6 Extracting common time interval list 46
3.7 Execution Trace of Linear Temporal Scan (LTS) algorithm 49
3.8 Intersection of two space time prism 50
3.9 Different spatial approximation filter around static ellipses 52
3.10 Ellipse Intersection via Space-Time Prisms 52
3.11 Different Spatial Approximations for Dynamic Circular Intersection 54
3.12 Comparison of Temporal Selection Search algorithm with Baseline Approach 55
3.13 Execution Trace of Temporal Selection Search Algorithm 56
3.14 Experiment Design 60
3.15 Effect of varying parameters on Area Pruning Effectiveness 61
3.16 Effect of varying parameters on computation time 62
3.17 Evaluation Confusion Matrix 63
3.18 Effect of varying parameters on accuracy 64
3.19 Effect of varying parameters on precision 64
4.1 Mining abnormal patterns without known spatiotemporal signatures 67
4.2 An illustration of the Abnormal Gap Region Problem (Best in color). . . . 69
4.3 Examples of Abnormal Gap Measure (Left) and Degree of Overlap (Right) 71
4.4 Framework for detecting possible abnormal gaps to reduce manual inspection by analyst 73
4.5 Execution trace of the baseline AGD algorithm 74
4.6 Execution Trace of Proposed Memo-AGD Algorithm 75
4.7 Spatio-temporal interpolation is better than linear interpolation for different parameters. 77
4.8 Comparison of execution times of AGD and Memo-AGD under different parameters 78
4.9 Comparison of Linear Interpolation and Memo-AGD around Galapagos Marine Reserve 79
5.1 Example of (a) a geo-ellipse and (b) an intersection (IB) and union boundary (UB) (Best in Color) 84
5.2 Examples of abnormal gap measure (Left) and degree overlap threshold (Right) 86

5.3 Framework for detecting possible abnormal gaps to reduce manual inspection by analyst . . . 87
5.4 Plane Sweep Approach (Filter Step) . . . 91
5.5 Examples of (a) Memo-AGD Gap Enumeration and (b) Memo-AGD AGM computations . . . 92
5.6 Temporal-Merge Aware Indexing of trajectory gaps . . . 94
5.7 Spatial Merge Aware Indexing of trajectory gaps . . . 95
5.8 Spatial Merge-Aware with Maximal Union Merge Criteria Execution Trace (Step 1-2) . . . 98
5.9 Spatial merge-aware with maximal union merge criteria execution trace (Step 3) . . . 99
5.10 Proposed sliding window approach to compute AGM incrementally . . . 100
5.11 Proposed Decreasing Queue strategy to sort gaps in leaf node by decreasing order . . . 102
5.12 Proposed Doubly Linked List on Decreasing Queue for O(1) insertion . . . 103
5.13 Dynamic Region Merge Execution Trace . . . 105
5.14 Experiment Design . . . 108
5.15 Entity Relation diagram of the MarineCadastre [1] dataset . . . 109
5.16 Synthetic Data Generation . . . 109
5.17 Proposed STAGD+DRM is always more accurate than Linear Interpolation and KNN-Imputation under varying parameters. . . . 112
5.18 STAGD+DRM runs faster than Memo-AGD under different parameters . . . 113
5.19 Comparison of k-NN Imputation, Linear Interpolation and Space-Time Interpolation . . . 114
6.1 A illustration of Rendezvous Region Detection in Spatial Networks (Best in color) . . . 119
6.2 Framework for detecting possible rendezvous locations . . . 122
6.3 Time Slicing Model with Lens at time instant t . . . 124
6.4 Time slicing model with lens intersection at time instant t . . . 125
6.5 An illustration of Possible Rendezvous Nodes . . . 128
6.6 An illustration of Time Dependent Shortest Path . . . 131

6.7 An illustration of TGARD vs Dual Convergence Method 134
6.8 Experiment Design 138
6.9 DC-TGARD vs. a space–time-prism method under different parameters . . 139
6.10 Accuracy of DC-TGARD versus the space–time-prism method under varying parameters 141
6.11 DC-TGARD is more efficient than TGARD under varying parameters . . . 143
7.1 Marine traffic volume near Ukraine ports before and during disturbance phase 148
7.2 An illustrative example of prediction output comparison 149
7.3 Frequency distribution of the bearing change of a tanker "Cathay Phoenix", whose abrupt changes are highlighted in red 150
7.4 A taxonomy of deception-based distortion 151

Chapter 1

Introduction

Spatial data has tremendous value and is a necessary component in many important societal applications. In recent years, spatial technologies such as Google Maps, Waze, Uber, Lyft, Grubhub, Lime, and autonomous driving have revolutionized our everyday lives. Location data on mobile devices generates hundreds of billions of dollars in revenue annually [2] with applications in energy, health, retail, etc. The world's economy also heavily relies on location and time data from over 2 billion GPS receivers [3], as this data is essential to applications in banking, air travel, law enforcement, emergency services, and telecommunications. Meanwhile, new types of spatial data are emerging at unprecedented scale and volume. In transportation, a single vehicle can generate onboard diagnostic data at 25 GB per hour [4]. Earth observation data is being collected at increasing spatial resolution and frequency because of its value in transportation, infrastructure security, resource mapping, agriculture monitoring, etc. GPS has become a critical infrastructure in our economy for use cases ranging from precision agriculture to navigation, ride sharing, smart cities, and public health. For example, a 2019 U.S. National Academy report projects 1.6 trillion in savings for energy generation and use from earth observation data by 2035. Moreover, the government and industry have started major initiatives, such as Amazon Earth on AWS, Google Earth Engine, and Microsoft AI for Earth, to address climate change and environmental sustainability. Spatial data research is funded by National AI Institute (e.g., AI-LEAF), and multiple Harnessing the Data Revolution (HDR) Institutes have been established. Spatial data science has also transformed modern life through innovations

such as navigation systems (e.g., Google Maps), precision agriculture (e.g., GEOGLAM), and autonomous driving technologies (e.g., high-definition maps). These advances have reshaped industries and addressed global challenges. A 2022 McKinsey Global Institute report projected that achieving net-zero emissions could generate over 12 trillion dollars in annual revenue by 2030, with agriculture and transportation contributing 1.2 trillion and 2.7 trillion dollars, respectively.

Spatial data science is the discovery of useful, interesting, and non-trivial patterns from spatial datasets [5]. Spatial data is *special* with distinct characteristics (e.g., reference system, resolution, etc.) and properties (e.g., spatial autocorrelation) which violate traditional i.i.d. (independent and identically distributed data samples) assumptions of traditional data mining and statistical methods. Thus, using traditional methods may result in missed or spurious patterns which can be costly in spatial applications. However, the growing ubiquity of location-aware technologies and the scale of GPS data have increased the need for spatiotemporal analytics that can handle incomplete, distorted, or deliberately manipulated data. Hence, this thesis explores spatial data science and GeoAI to address the challenge of data distortion inspired by the mirror metaphor: *Data alone is insufficient to distinguish between a distorted reflection of a perfect world, a perfect reflection of a distorted world, or a combination of both* [6], especially when such distortion arises in the context of emerging spatial big data. Even though ships are required to report geolocations periodically, some may *cover their tracks* by not reporting geolocations, creating data distortion. The resulting trajectory gaps make it difficult to monitor illegal fishing and cargo transfers on the high seas. We formulate this setting as the detection of abnormal trajectory gaps and possible abnormal rendezvous activity between moving objects.

1.1 What are Data Distortions?

The mirror metaphor *"Data alone is insufficient to distinguish between a distorted reflection of a perfect world, a perfect reflection of a distorted world, or a combination of both."*[6] conceptualizes data as a reflection of reality. When addressing preexisting bias within data, we examine this reflection and recognize that distortions often arise. Such distortions may originate either from imperfections in the measurement process, resulting in selective amplification or suppression of aspects of reality, or from the fact that even an ideal measurement process *(a perfect mirror)* reflects the world as it currently is, including

any intrinsic societal biases, rather than portraying an idealized version of reality. The article [6] discusses biases, especially technical biases introduced through the operation of technological systems themselves, which often amplify existing societal biases. This thesis aligns with technical bias because it emerges primarily from decisions made during preprocessing stages or from issues arising after the deployment of data-intensive pipelines. Despite its acknowledged significance, limited research has addressed diagnostic methods or practical interventions to mitigate technical bias. One such technical bias is based on flawed assumptions during missing-value imputation, particularly assumptions regarding whether data is missing at random. For instance, consider a scenario in which job application forms allow applicants to indicate gender as male, female, or unspecified. Suppose that approximately equal numbers of men and women apply, but women are more likely to leave this question unanswered. Mode imputation, like assigning the most frequent gender (e.g., male), often mislabels unspecified entries, especially from women, as male. This distorts data, particularly for minority groups, as multiclass imputation tends to favor majority classes, ignoring smaller subpopulations.

In spatial data science, distortion is often studied through outlier or anomaly detection in spatial or spatiotemporal data. This thesis studies data distortion in spatiotemporal trajectories, especially in maritime space. Even though ships are required to report geolocations periodically, some may "cover their tracks" by not reporting, creating data distortion. The resulting trajectory gaps make it difficult to monitor illegal fishing and cargo transfers on the high seas. This thesis formulates the problem as detecting abnormal trajectory gaps and possible rendezvous activity. Related work from movement pattern analysis and trajectory mining has considered interpolating trajectory gaps without considering the full range of an object's movement possibilities. It has also focused on bare-bones trajectories containing only location information at each timestamp. To address these limitations, we first created a taxonomy that illustrates the spectrum of data distortion challenges in trajectory analysis, from data removal to noise addition and outright falsification, and then developed resilient algorithms to handle such distortions while preserving accurate detection of patterns such as rendezvous points.

(a) Two ships performing an illegal transfer

(b) A long trajectory gap around Galápagos Reserve

Figure 1.1: Illustrative examples of illicit activities in maritime space.

1.2 Data Distortions in Spatial Data Science

Motivation: Marine safety and regulation enforcement are important for global security concerns, such as illegal trans-shipments and fishing. Figure 1.1a shows an example of a Hong Kong ship possibly transferring oil to another ship from North Korea in the ocean, evading the United Nations [7] sanctions in October 2017. While performing these illicit activities, offenders tend to switch off their GPS transponders to hide their movement, as recent news articles [8, 9] described: "When a ship goes dark in the Persian Gulf, it may be related to dodging sanctions." Earlier in 2012, a similar illegal oil transfer to Iran possibly occurred in the Persian Gulf [10]. Transshipment is also a way to "dodge taxes" by avoiding higher tariffs while shipping goods between international subsidiaries [11].

Beyond global security concerns, many organizations and government agencies have put great effort into fighting illegal fishing. For example, in 2014, the White House released the presidential memorandum "Establishing a Comprehensive Framework to Combat Illegal, Unreported, and Unregulated Fishing and Seafood Fraud" [12]. In Figure 1.1, a Panamanian commercial fishing vessel switched off its location-reporting AIS device before entering the Galapagos Marine Reserve and then switched the device back on after 15 days. Some believe that it may hide illegal fishing [13]. "Data analysis found that of 200 Chinese vessels targeting squid off the coast of Peru, about 20 percent weren't broadcasting a signal, indicating potential foul play." [14]. Recently, a Chinese coast guard ship caused a near-collision with a Philippine patrol vessel in the South China Sea, which might have been caused by "turning off a tracking mechanism to appear invisible" [15].

Challenges: Current data-driven methods fail to capture such adversarial behavior,

making it reasonable to use a physics-based approach to model uncertainty. First, we spatially quantify a trajectory gap based on the idea that an individual object's possible movements are constrained by certain limits of space and time. We represent these constrained movement possibilities as a space-time prism [16] by considering physics-based attributes (e.g., ship's speed). This results in a more precise trajectory reconstruction technique than the classical assumption of shortest or most frequented path interpolation derived by current methods. The proposed method captures certain out-of-sample predictions often missed by the shortest or most likely path assumption.

An interpretable toy example: Examples of trajectory data with multiple attributes include maritime data with recordings of ship location, heading, draught, etc. A spurious pattern represents a significant deviation from expected values. An example is a long gap in a trajectory in a geographic area where a vast majority of trajectories do not have long gaps. The resulting spurious patterns may help detect malicious activities such as illegal cargo transfers and illegal fishing. However, traditional trajectory mining approaches tend to underperform or fail because they assume trajectory availability and precision. For instance, Figure 1.2 compares related work in which physics-based methods use the shortest path or linear interpolation during the gap, as depicted by the red line. The shortest-path approach may fail to capture possible aberrant behavior or patterns, such as a detour to a marine reserve depicted in the dark blue area inside the space-time-model-based red ellipse, because the object may wander during the unreported interval.

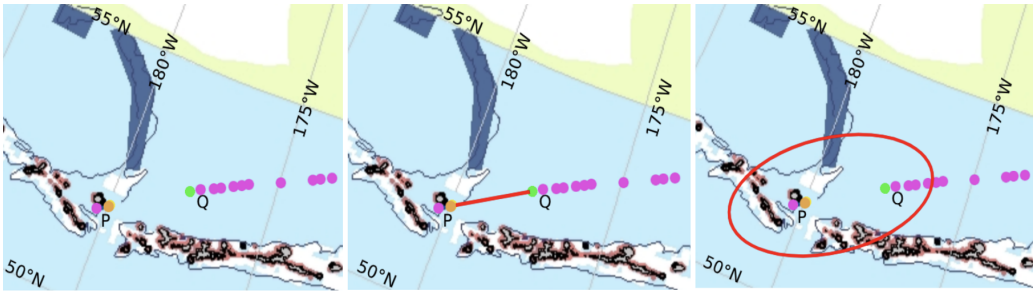


Figure 1.2: Space-Time Prism interpolation may find patterns missed by linear interpolation

In addition, the problem is computationally challenging because of the large data volume. For instance, MarineCadastre [1] is an Automatic Identification System (AIS) that

records more than 30 attributes (e.g., location, draught) for 150,000 ships every minute. Its size reaches hundreds of terabytes per year and about a petabyte per decade. Since satellites have limited capability to cover all parts of the ocean, vessels are required to carry location broadcasters that report signals via satellite to help prevent accidents and collisions. However, certain vessels intentionally switch off their location signals near marine protected habitats, creating potential abnormal activities. Monitoring the world's ships for abnormal activity is a time-intensive operation for human analysts because ships report their location-trace signals every minute, each annotated with geographic location and multiple ship attributes such as speed, direction, draft, etc. This results in terabytes of continuously updating data to analyze, and the high update rate gives rise to spatial big data. The volume, variety, and update rate of incoming maritime location-trace signals exceed the capacity of traditional spatial computing and database technologies to manage and process at reasonable cost.

Data Distortion Taxonomy: First, we created a taxonomy of data distortion categories that illustrates the spectrum of data distortion challenges in trajectory analysis, from data removal to noise addition and outright falsification. We developed resilient algorithms to handle such distortions, ensuring accurate detection of patterns like rendezvous points while paving the way for future research into countering sophisticated attack models.

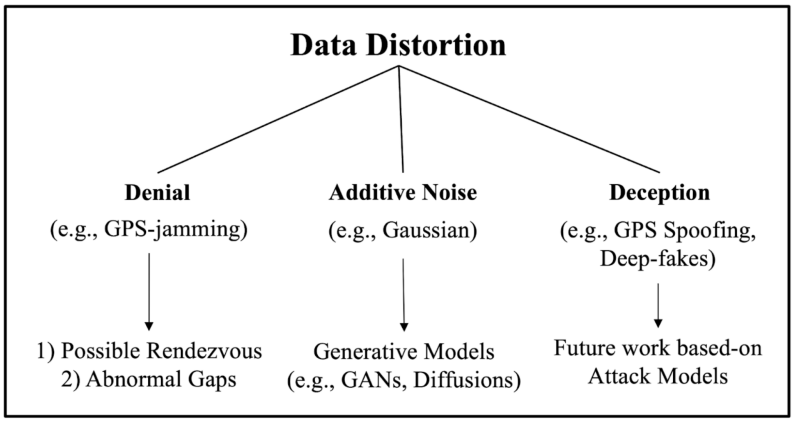


Figure 1.3: Data Distortion Taxonomy

The first category involves **denial-based** techniques (e.g., GPS jamming), which result in data loss or gaps. This category relates to the challenges of missing data in trajectory

analysis and is the focus of this thesis. The second category is **additive noise**, which involves adding noise from a probability distribution (e.g., Gaussian) to existing signals or learning such distributions with generative models. The third category, deception, involves GPS spoofing and deepfakes, where data is replaced with fabricated data that may lead to incorrect conclusions. In this thesis, we only concentrated on denial-based distortion. First, we formally defined abnormal behavior with a known abnormal signature (i.e., identifying possible rendezvous behavior) where ships have engaged in illicit activities. We then applied an abnormal gap metric based on historic location trace signals for trajectory gaps with no known signatures. Hence, this denial-based distortion can lead to two key outcomes: (i) identification of possible rendezvous points where entities might meet during these gaps and (ii) detection of abnormal gaps, which could indicate intentional switching of their location broadcasting device despite signal coverage. To identify the rendezvous region, we aim to narrow the search space so that human analysts can monitor rendezvous areas with a time-slicing approach via increasing temporal sampling. For unknown signatures, we measured the probability that a possible location of the object during the gap (unreported data time interval) had signal coverage.

Scope: In this thesis, we focus on trajectory gaps, i.e., denial-based distortion where the end user deliberately switches off its location device. Such distortions are verified by real-world case studies [13] in close interdisciplinary collaboration with domain experts. This use-inspired work seeks fundamental understanding of scientific problems while also having immediate use for society [17]. Other sophisticated distortions, such as additive noise and deception-based behavior, are discussed as short-term and long-term future work.

1.3 Scope and Terminology

The title of this thesis, *Distortion-Aware Spatial Data Science*, is deliberately broad. By distortion we mean any systematic difference between what spatial data records and what physically happened. Distortion of this kind appears throughout spatial work: in how location data is stored and indexed, in how it is queried, in the statistics computed from it, and in the models trained on it. The idea of distortion-aware reasoning is meant to carry into any of these settings, not only the ones studied here.

Within that broad program, the work developed in the most depth is narrower and more specific. We refer to it as *Distortion-Aware Spatial AI: A Physics-Informed, Neuro-Symbolic*

Approach. Spatial AI here means the learning and reasoning models, that is the neural networks, generative models, and search procedures, that predict, detect, and generate over spatial data. The problems studied are physics-informed ones: cases where a known physical model constrains what the data can plausibly be. Two such models recur throughout the thesis, the space-time prism and the kinematic bicycle model. They let us reason about where a vessel could have been while its transponder was off, and about whether a reported track is physically possible for a vehicle of a given size. The relationship between the title and the contribution is therefore one of scope. Distortion-aware spatial data science is the wider idea; distortion-aware spatial AI, physics-informed and neuro-symbolic, is the specific instrument built and tested inside it. The chapters that follow develop that instrument, and the terms used throughout are fixed below.

Spatial data science. The broad practice of managing, querying, analyzing, and modeling data tied to location and movement. It covers indexing and storage, spatial statistics, pattern mining, and machine learning, and is wider than any single model or method.

Spatial AI. The part of spatial data science concerned with learning and reasoning models: neural networks, generative models, and search or optimization procedures that predict, detect, or generate spatial outcomes. Spatial AI is a subset of spatial data science, not a synonym for it.

Distortion. A systematic difference between what the data records and what physically happened. We group distortions into three kinds. Denial is missing data, for example an Automatic Identification System (AIS) transponder switched off so that a vessel stops reporting. Additive noise is perturbed data, for example coordinate error from a map projection or a noisy sensor. Deception is fabricated data, for example a spoofed signal that places a ship somewhere it never was.

Distortion-aware. A method is distortion-aware when it represents and reasons about one or more of these distortions explicitly, rather than assuming the data is a faithful record. A distortion-aware model asks what could have been true given the gaps, the noise, or the possibility of fabrication.

Physics-informed. A method is physics-informed when it builds a known physical model into its structure or its training, instead of learning all behavior from data. The physics acts as a constraint on what the model will accept as plausible, which matters

most when data is scarce or untrustworthy.

Neuro-symbolic. A method is neuro-symbolic when it combines neural components, which are good at pattern recognition, with symbolic components, which are good at exact reasoning and verification. Graph search, constraint checking, and geometric tests sit on the symbolic side; learned encoders and generative models sit on the neural side. The reason to combine them is that the symbolic part can check, and in some cases guarantee, what the neural part proposes.

Space-time prism (STP). A construction from time geography, due to Hägerstrand. Given two known positions of a moving object and a maximum speed, the space-time prism is the set of all locations the object could have occupied in between. Projected onto the map it becomes a geo-ellipse with the two known positions as foci. We use it to bound where an object could have gone during an interval when it was not reporting.

Kinematic bicycle model (KBM). A compact model of vehicle motion that relates position, heading, speed, and steering angle, and so captures how a wheeled vehicle can actually move. We use it as a physical prior: a reported or generated trajectory that violates the kinematic bicycle model is, to that extent, implausible.

Purely data-driven models are strong when data is plentiful and trustworthy, and spatial data is often neither. Trajectories have gaps, coordinates carry projection error, and adversaries can fabricate signals outright. Under these conditions a model with no notion of physical plausibility will confidently fill a gap with a path no vehicle could drive, or accept a spoofed track because it resembles the training data. Physics gives the model a way to say what is possible: the space-time prism bounds where an object could be, and the kinematic bicycle model bounds how it could have moved. Symbolic reasoning then gives the model a way to check its own output, through a geometric intersection test, a constraint on speed and turning, or a search over a road graph. The neural part still recognizes patterns and proposes candidates, but it no longer has the last word. This combination is the thread that runs through the thesis: the detection methods for trajectory gaps and possible rendezvous use exactly this pairing of physical bounds and symbolic checks, and the generative work discussed as future work extends it to producing data that respects the same physical constraints.

It helps to name the two ways the neural and symbolic parts can be arranged, following the taxonomy of Kautz [18] as developed for spatial computing by Yang [19]. In a Symbolic[Neuro] design the symbolic procedure is on the outside and calls neural parts from within, as when a routing or search algorithm operates over learned cost estimates [20]. In a Neuro[Symbolic] design the neural model is on the outside and carries symbolic or physical knowledge inside it, as when a generative model is constrained by a motion law during training. The detection methods in this thesis are symbolic on the outside: a space-time prism or a road-network bound defines the feasible region, and the pruning logic in MOBR, STAGD, and TGARD checks candidates against it, with learned components in a supporting role. The physics-informed generative direction points the other way. A diffusion model that respects a kinematic prior, as in our work on physics-informed trajectory diffusion [21] and in the closely related geo-lucid conditional diffusion model [22], is a Neuro[Symbolic] instance, since the network proposes trajectories and the physics keeps them feasible. Reading the thesis along this axis connects the detection chapters that follow to the generative work described later as future directions, since both are points on one neuro-symbolic spectrum.

1.4 Contributions and Thesis Organization

The overall research methodology is inspired by Stokes' "Pasteur's Quadrant" framework [17], where scientists motivated purely by curiosity about fundamental laws resemble Bohr, while government researchers in the spirit of Edison prioritize practical, deployable solutions for society. University researchers, however, occupy the hybrid *"Pasteur"* quadrant: like Louis Pasteur, they pursue theoretical advances and real-world applications in tandem. Accordingly, the work presented here is embedded in a collaborative network, ensuring that its contributions remain both intellectually rigorous and pragmatically useful.

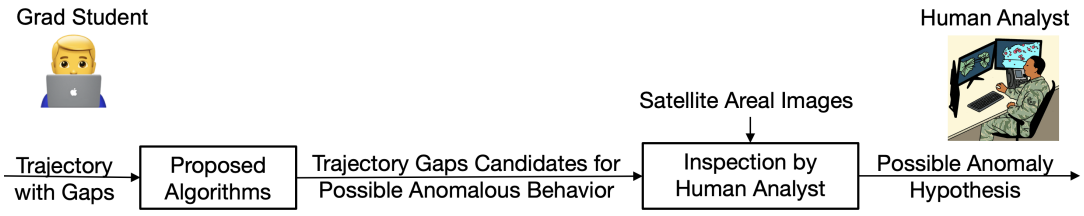


Figure 1.4: The proposed use-inspired framework for denial-based behavior

The proposed framework in Figure 1.4 is based on a filter-and-refine approach, where trajectory gaps are first preprocessed by the proposed algorithms for early-stage filtering. Our aim is to quantify the search space within the trajectory gap under physical laws and constraints while also reducing geometric area to minimize the monitoring region. The reduced area is then sent to human analysts for manual inspection, reducing post-processing effort for timely ground-truth verification.

Contribution: This thesis proposes a three-stage framework for analyzing denial-based distortion in trajectory data with gaps, as described in Figure 1.5. First, we model trajectory gaps using physics-based attributes such as speed, while incorporating geometric properties such as ship dimensions and physical laws. Next, we develop querying methods to detect aberration patterns with known spatiotemporal signatures, drawing on theories from physics and behavioral science. Finally, we investigate mining techniques to uncover novel aberration patterns without predefined signatures, using statistical models informed by query results. The first goal is to create scalable methods to narrow down geographic areas of possible rendezvous, reducing the effort required for manual verification using auxiliary data such as satellite imagery. The second goal is to extend this approach into network space, utilizing known rendezvous patterns observed in civilian contexts. Third, we explore algorithms to detect abnormal gaps in trajectories, where a ship fails to report its location while others do so regularly.

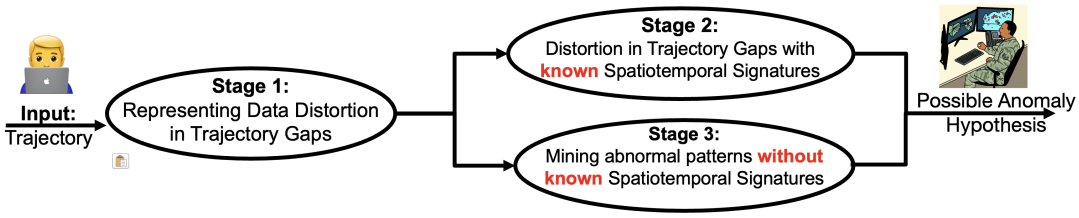


Figure 1.5: The overall framework for denial-based behavior

Organization: This thesis presents algorithms and modeling frameworks for detecting abnormal trajectory gaps and potential rendezvous events in spatiotemporal data. In Chapter 2, we address scenarios with known co-occurrence signatures to detect maritime rendezvous using geometric and temporal overlap. Chapter 3 focuses on improving detection efficiency through optimized spatial pruning techniques. Chapter 4 shifts attention to individual trajectory gaps lacking known signatures, introducing a physics-guided abnormal

gap measure. Chapter 5 expands upon this with a hierarchical and dynamic spatial indexing method for scalable detection. Chapter 6 generalizes these ideas to spatial network settings, enabling detection of coordinated behavior in constrained environments. Chapter 7 concludes the thesis and provides both short-term and long-term future work. In Table 1.1, we offer a structural overview of the thesis and preview its key contributions, which span from theoretical insights to scalable algorithms designed for real-world mobility analysis under uncertainty.

Table 1.1: Thesis contribution taxonomy.

Data Distortion			
	Denial-based Distortion (e.g., Subtraction)	**Additive Noise Distortion (e.g., Addition)**	**Deception-based Distortion (e.g., Substitution)**
Anomaly Detection with Known Spatiotemporal (ST) Signatures	Analyzing Trajectory Gaps for Possible Rendezvous Region **(Chapter 2, 3)**, Towards tighter Bounds for Possible Rendezvous on Spatial Networks **(Chapter 6)**		
Anomaly Detection without known ST Signatures	Physics-based Abnormal Trajectory Gap Detection **(Chapter 4, 5)**		
Anomaly Detection with Classification & Prediction			

- Chapter 2 introduces the problem of detecting spatiotemporal rendezvous in maritime trajectories with missing or manipulated AIS data. It proposes a baseline approach

that formalizes spatiotemporal co-occurrence between trajectory gaps using space-time prisms and minimum orthogonal bounding regions (MOBRs). The chapter develops a scalable rendezvous detection algorithm based on time-slicing and pruning via geometric conditions, and evaluates its effectiveness on real-world ship data. Results show that the approach significantly reduces the candidate rendezvous regions for identifying suspicious meeting events in vast maritime spaces.

- Chapter 3 improves upon the baseline method by introducing ellipse-based gap modeling to better capture motion uncertainty during AIS signal outages. It presents multiple spatial and temporal pruning techniques that eliminate unnecessary computations through tighter geometric bounds. Theoretical analysis is provided to guarantee correctness and completeness of the method. Experiments on synthetic and real datasets demonstrate up to 10x speedup and greater interpretability of detected rendezvous events compared to prior methods.
- Chapter 4 presents Memoized Abnormal Gap Detection (Memo-AGD), a method designed to identify abnormal gaps in individual trajectories where no co-occurrence signatures are available. It introduces a lookup-based spatial structure for caching prior gap interactions, reducing redundant intersection checks. Memo-AGD incorporates a novel abnormal gap measure (AGM) to quantify spatiotemporal deviation under physical constraints, and shows improved precision over baseline methods.
- Chapter 5 extends Memo-AGD by proposing Spatio-Temporal-Aware Gap Detection (STAGD) and Dynamic Region Merge (DRM). These methods combine hierarchical spatial indexing and AGM-based scoring to incrementally merge trajectory gaps while preserving spatial consistency. The chapter introduces an early termination criterion to avoid unnecessary updates and a doubly linked structure to support efficient cell-based recomputation. Extensive experiments show that STAGD+DRM achieves improved anomaly detection accuracy while significantly reducing computational cost.
- Chapter 6 generalizes the anomaly detection problem to spatial networks, presenting TGARD and DC-TGARD for discovering collusive behavior (e.g., illicit meetings) in graph-based mobility scenarios. It introduces dual convergence logic to prune both spatial and temporal candidates, and employs periodicity-aware pruning based on signal repetition patterns. Evaluations on real-world riverine and road networks show

DC-TGARD's superior performance in reducing false positives while maintaining recall, illustrating its adaptability to broader surveillance settings.

- Chapter 7 summarizes the thesis findings and gives an overview of related directions with short-term and long-term future work.

Chapter 2

Investigating Possible Rendezvous Regions: A Time Slicing Approach

2.1 Introduction

Given multiple trajectories which have gaps due to weak signals, instrument malfunction or malicious interference we find possible possible times and places where moving objects (e.g. ships) rendezvous or meetup. Figure 2.1 shows an example of the input and output. For simplicity, we are using one-dimensional geographical space along with the dimension of time. Object 1 is shown in blue and Object 2 is shown in red. The gaps are shown in a dotted form and lie between P3, P4 for blue and P4, P5 for red. Object 1 has a maximum speed of 1 and Object 2 has a maximum speed of 2. The output shows the candidate active volume (CAV) for each object. A CAV is the region in a gap that represents all the possible locations of the object during a missing time interval. The intersection of the two CAVs is the possible rendezvous region termed as a spatio-temporal intersection (STI).

Analysis of gaps in trajectories has many societal applications related to maritime safety, homeland security, epidemiology, and public safety. For example, maritime safety and regulation enforcement are important for global security for concerns such as illegal oil transfer and trans-shipments. Such activities can be restricted and managed by identifying frequent missing signals from GPS trajectories of oil vessels with gaps. however, the trajectories can be spread over a large geographical space and manual inspection for gaps

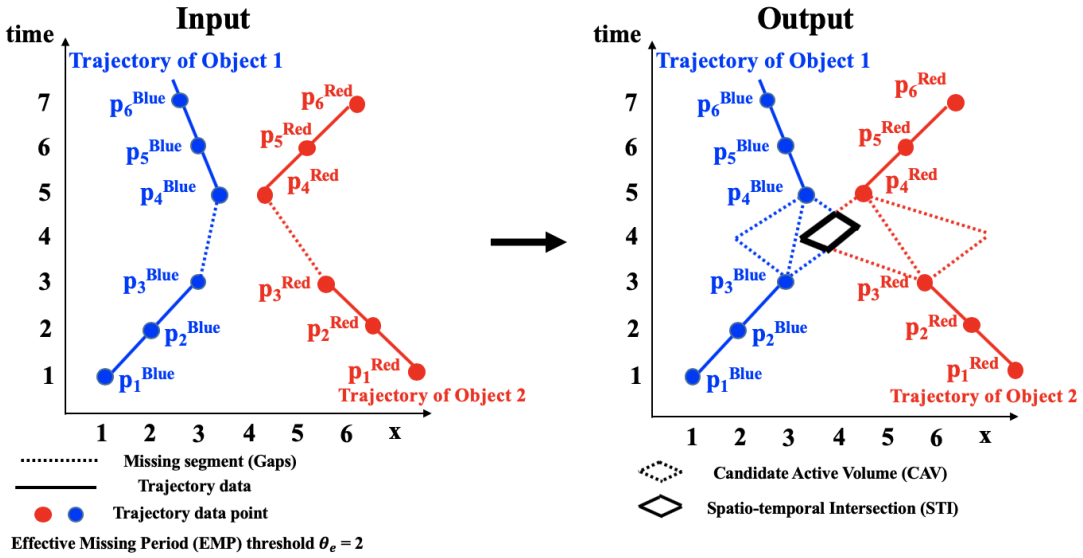


Figure 2.1: An illustration of rendezvous region detection (Best in color).

to detect rendezvous can be time intensive. Computational methods that detect possible rendezvous regions can substantially reduce the preliminary work for human analysts.

The problem is computationally challenging because the gaps may cause traditional trajectory mining approaches [23] to underperform or fail where they assume the availability and preciseness of trajectories. A second challenge is that the data has a large volume and is spread over considerable geographical space. For example, MarineCadastre [1] is an automatic identification system (AIS) dataset that contains records for more than 30 attributes (e.g., location, draught) for 150,000 ships taken every minute during the years 2009 to 2017. Its total size is about 600 GB and covers all the waters around the US.

The literature on movement pattern analysis [24] and trajectory mining [23, 25] interpolates gaps in trajectories without considering the full range of an object's movement possibilities. The approach provides an approximate solution and may miss possible rendezvous points. Areal interpolation of the gaps has been done through various prism models (e.g., space-time prism [16], kinetic prism [26]). A space-time prism overlap for two or more objects can be thought of as a potential rendezvous region or meeting place for moving objects. It may be computationally modeled as a spatio-temporal intersection of trajectories and can alternatively be called a spatio-temporal co-occurence.

In this chapter, we study the computational cost of determining potential rendezvous regions. The baseline method uses a plane sweep [27] technique to identify the potential rendezvous regions. To improve the efficiency of plane sweep, we propose partitioning the space into space-time grids. The grids are used to cover all the possible paths the object can take. The grid overlapping two or more objects is a rendezvous region. The proposed framework ensures completeness by finding all the possible rendezvous regions, and ensures correctness because the rendezvous point is bound within the region. We further reduce the geographical search space through time slicing techniques. We use plane-sweep as the baseline to compare our results. Experimental results show that the proposed approach gives tighter bounds. Further, results from time-slicing techniques improve as we increase the time-slicing factor or use a finer time scale.

Contributions:

– We formally define the problem of rendezvous region detection for spatio-temporal trajectories with gaps.

– We propose a space-time partitioning approach to detect rendezvous regions. The approach is further refined to give more accurate approximation using time-slicing techniques.

– We propose and use a new evaluation metric, area pruning effectiveness (APE), to compare the methodologies.

– We compare the proposed approach with the plane sweep based baseline on various relevant evaluation parameters (e.g., study area) and metrics. Results (Section 2.6.2 show that the proposed approach has better APE values.

– We provide a case-study on ship trajectories from the Bering Sea to show the effectiveness of the proposed approach on a real-world dataset. We find that the proposed approach gives better results on the study area.

Scope: In this thesis, we do not study kinetic prisms [26]. Further, the proposed framework has multiple phases and we limit this work to the filter phase. The refinement phase requires input from a human analyst and is not addressed in this work. Furthermore, the calibration of cost model parameters is outside the scope of this work. In addition, we do not model rendezvous areas of trajectories without gaps which are involved in intersection. Finally, we do not address the issue of positional accuracy while modeling the trajectories.

Organization: The chapter is organized as follows: Section 2.2 introduces basic concepts

and the problem statement. Section 2.4 describes the proposed framework and approach used in our work. Experiment design, results, and a brief discussion on the computation cost of our approach are reported in Section 2.6. Section 2.3 reviews the related work (in more detail). Finally, Section 2.7 concludes this work and briefly lists future work.

2.2 Basic Concepts and Problem Statement

2.2.1 Basic Concepts

This section reviews several key concepts in the rendezvous detection problem and presents a formal problem formulation.

Definition 2.2.1. A **study area** is a two-dimensional rectangular area where the input data are located. It usually complies with the (latitude, longitude) coordinate system.

Definition 2.2.2. A **spatial trajectory** is a trace generated by a moving object in a geographic space, that is usually interpreted as a series of chronologically sorted points, for instance, $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, where each point (p_i) is associated with a geospatial coordinate set (x, y) and a time stamp (t).

Definition 2.2.3. An **object maximum speed ($Smax$)** is the maximum speed of an object based on the domain knowledge.

For maritime data, S_{max} can be identified from publicly available vessel databases [1]. For vehicles, humans, or animals, we can use the maximum physically allowed speed.

Definition 2.2.4. An **effective missing period (EMP)** is a time period when the signal is missing for longer than a user-specified EMP threshold (θ_e) .

As shown in Figure 2.1 the EMPs for Object 1 and Object 2 is between timestamps 3 and 4, Here, we assume $\theta_e = 2$.

Definition 2.2.5. A **candidate active volume (CAV)** is the spatio-temporal volume where an object is possibly located during an EMP [28, 29, 30, 16, 31]. A CAV is based on a space time prism using conical shape and is derived from an EMP.

2.2.2 Problem Formulation

The problem to identify optimized rendezvous patterns in a spatio-temporal domain is formulated as follows:

Input:

1. A study area S,
2. A set of $|N|$ trajectories $T = t_1 \dots t_{|N|}$, each associated with an object,
3. An object maximum speed (S_{max}) for each object,
4. An effective missing period threshold (θ_e).

Output: Approximate geometry intersection of two gaps.

Constraints: Minimal Filter Storage Cost.

Objective: Improve area pruning effectiveness (APE)

For example Figure 2.1 illustrates the one-dimensional representation of two gaps involving a spatiotemporal intersection given the study area (one-dimensional), two trajectories, two object maximum speeds, and $\theta_e = 2$. The output is the STI represented by the triangle shown the Figure.

Area pruning effectiveness (APE) is the ratio of the total study area and minimum bounded area inside the filtered region (i.e. approximate CAV).

$$area\ pruning\ effectiveness\ (APE) = \frac{total\ study\ area}{area\ bounded\ inside\ the\ region} \tag{2.1}$$

If the value of APE is higher, the solution quality is better since the minimum bounded area enclosed within a CAV will be lower.

2.3 Related Work and Limitations

Due to recent advancements in location-acquisition services, and mobile computing research, an extensive amount of trajectory data is available which serve different research purposes such as pattern recognition, anomaly detection, etc. The work in [23] provides a comprehensive survey of trajectory data mining and discussed reconstruction techniques which illustrate a variety of frameworks for modeling uncertainty and noise in trajectory data. However, all the techniques are based on assumptions related to linear interpolation or shortest path discovery. These approaches are not designed to detect patterns when

the trajectories of the moving objects are missing (e.g., due to weak signaling) in which case the objects possibly move far from the shortest path. Trajectory data mining has also motivated interdisciplinary research in other fields such as geography and ecology. In [24], the authors provide a unified taxonomy of moving objects concerning their movement patterns by classifying them into generic and behavioral patterns, which encompass patterns such as co-locations, co-occurrence, etc.

Movement behavior patterns such as evasive patterns are used to detect potential anomalies. Analyzing maritime trajectory data with gaps is a particular case of an evasive pattern. A recent survey, [32] provides a panorama of existing techniques to identify anomalous patterns in maritime trajectories by classifying them as data-driven, signature-based, and hybrid methods. Many frameworks [33, 34, 35, 36] have been proposed for analyzing evasive patterns in maritime trajectories. For instance, the authors in [36] proposed a method for determining if the vessel is anomalous by considering longitude, latitude, speed, and direction for each trajectory point and providing a three-division distance that can detect anomalous navigational behaviors. Stop and move [37] is an another conceptual model which analyzes anomalous behavior based on DBSCAN [38], and speed and direction [39], when it comes to ship trajectories data. These techniques do not apply to our work, which is focused on interpreting gaps in trajectory data, rather than detecting anomalous behaviour.

More realistic solutions for modeling gaps in trajectory data are contextual models such as space-time prisms [16, 40] that construct an areal interpolation of the gaps using coordinates and maximum speed of the objects. More recently, the kinetic prism model [26] provides a better estimation by considering other physical parameters such as uncertainty and acceleration. However, applying these models can be computationally expensive. One way to address the cost is through spatial indexing. Many spatial indexing techniques such as 3D R-Trees [41] or many others as described in [41], [42], [43] could be used to index trajectories efficiently. Other spatial indexing techniques, such as Hilbert Curve [42] have also been used to provide a computational speedup. The literature related to space-time prisms addresses computational speedup by using an alibi query for checking whether two space-time prisms intersect theoretically [44] [45]. In this work, we introduce the use of space-time prisms using grid-based indexing for detecting possible rendezvous patterns over maritime trajectories.

2.4 Approach

We begin with an overview of our framework. Then we describe the baseline algorithm (Plane sweep), a naive Spatio-temporal Grid Traversal (SGT), and the proposed Spatio-temporal Grid Traversal algorithm with time slicing (SGT-TS) in detail with their corresponding execution trace.

2.4.1 Framework

Our aim is to identify possible rendezvous regions on a given set of trajectories through a two-phase *Filter* and *Refine* approach. Figure 2.2 shows a representation of the proposed framework. The framework includes three algorithms: a baseline plane sweep algorithm [27], a naive space-time grid (SGT) algorithm, and a space-time grid algorithm with time slicing (SGT-TS). Plane sweep [27] is a basic computational geometry concept for finding intersections (e.g., line segments, polygons). We used the plane sweep algorithm to extract a minimum orthogonal bounding region (MOBR). SGT partition the study area into 3-dimensional (3D) grid cells (x,y, and time), where we approximate two endpoints based on the maximum speed for each trajectory, and in total four endpoints including the starting, and ending points of a gap segment that illustrates a possible rendezvous region. SGT-TS adds a time-slicing technique for selecting a more accurate region that helps reduce data redundancy and storage cost. The output from the filter phase is given to the refinement phase where we can find the exact geometry of the cone intersect using accurate modeling of the space-time prism of each object. The exact geometry can then be used by human analysts for ground truth verification via satellite imagery.

2.5 Necessary and Sufficient Condition of Spatiotemporal Intersection

Lemma: Two beads must be intersected if and only if

$$r_{start} + r_{end} \leq dis(start, end) \tag{2.2}$$

where index $start = \{start^{EMP_1}, start^{EMP_2}\}$, index $end = \{end^{EMP_1}, end^{EMP_2}\}$, and $disstart, end$ is the Euclidean distance between points P_{start} and P_{end}.

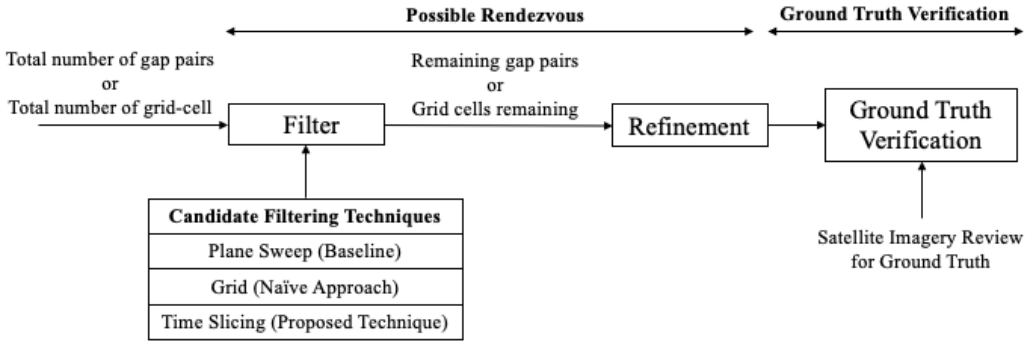


Figure 2.2: Framework for detecting possible rendezvous regions to reduce manual inspection by analyst

Proof: For the necessary condition there must be at least one timestamp when the sections of these two beads intersect i.e. if two beads have an overlapping time range, at least one-time stamp (start-point or the end point) of one of the gap segments must be between the time range of the other. To prove this, we take two data gaps, EMP_1 and EMP_2 having time range (t^1_{start}, t^1_{end}) and (t^2_{start}, t^2_{end}) respectively and check if the difference between (t^1_{start}, t^2_{end}) or (t^1_{end}, t^2_{start}) ≥ 0. If true, then the two EMPs satisfy the necessary condition of a two EMP intersect.

For the sufficient condition, if there is one timestamp that is between the overlap of the time gaps, the two beads must intersect. In order to satisfy this condition, we use the radius information of two cones from a different gap and check whether their respective radii will overlap with each other. According to the condition whether two circles overlap, the sum of their radii must be smaller than the distance between their respective centers. As stated in Equation 4, the sum of the radius of the cone from start point r^1_{start} of EMP_1 and end point r^2_{end} of EMP_2 must be less than the distance between their respective radii centers $dist(s, e)$. Using known t and S_{max}, when $S^1_{max} \geq S^2_{max}$, Equation 4 is further derived into:

$$t \geq \frac{d_{s,e} + t_s \times S^1_{max} + t_e \times S^2_{max}}{S^1_{max} + S^2_{max}} \rightarrow \text{where } s = start^{EMP_1},\ e = start^{EMP_2} \tag{2.3}$$

$$t \geq \frac{d_{s,e} + t_s \times S^1_{max} - t_e \times S^2_{max}}{S^1_{max} - S^2_{max}} \rightarrow \text{where } s = start^{EMP_1},\ e = end^{EMP_2} \tag{2.4}$$

$$t \leq \frac{d_{s,e} + t_s \times S^2_{max} - t_e \times S^1_{max}}{S^2_{max} - S^1_{max}} \rightarrow \text{where } s = end^{EMP_1},\ e = start^{EMP_2} \tag{2.5}$$

$$t \leq \frac{d_{s,e} + t_s \times S^1_{max} - t_e \times S^2_{max}}{-S^1_{max} - S^2_{max}} \rightarrow \text{where } s = end^{EMP_1},\ e = end^{EMP_2} \tag{2.6}$$

In the case that $S^1_{max} < S^2_{max}$, the conditions are derived by swapping the variables.

If the conditions are all satisfied, we know these two EMPs intersect. In contrast, if any of the conditions is not satisfied, the two EMPs do not intersect.

2.5.1 Baseline Algorithm

Gaps in a trajectory can be analyzed by computing the minimum orthogonal bounding region (MOBR) over the set of gaps for an individual trajectory. We use the plane sweep algorithm [27] for extracting MOBRs. It is a filter and refine approach [46] where the given study area is projected into a lower-dimensional space. In the filtering phase, all gaps are sorted based on x or y coordinate. Ordering on one dimension reduces the storage and I/O cost, and further allows the computation of intersections in a single pass. In the refinement phase, the gaps are extracted based on the start time and end time of their respective effective missing periods (EMPs). The segments are further approximated using MOBRs over each candidate active volume (CAV). The following describes the algorithm.

Step 1: Sort the endpoints of all the effective missing periods (EMPs) First, we sort the endpoints of all EMPs based on one of the coordinates. An endpoint is represented by three coordinates, namely x, y, and time, and either x or y can be the sorting coordinate. For consistency, We use x throughout this chapter.

Step 2: A plane orthogonal to the x-axis sweeps along the sorted EMPs The second step conducts the sweeping. Imagine there is a plane parallel to the y-t plane and orthogonal to the x-axis sweeping from the low to the high end along x-axis. The sweeping plane stops at both start and end endpoints of each EMP. Note that "start" and "end" refer to the order of sweeping, which is irrelevant to the temporal dimension. An Observed Object List is maintained CAVs being currently crossed by the sweeping plane.

When stopping at the start of an EMP, the algorithm first determines if the gap is larger than the given EMP threshold (θ_e). If it is, a new CAV is constructed for that object

along with an approximate MOBR around the new CAV discussed later in Section 2.5.2. The CAV along with its MOBR is then saved inside the observed list and a check is done to see if any other CAV inside the list is intersecting with the given CAV. If it is, a common MOBR around the pair of intersecting CAVs is added to the observed list as well as the output. The sufficient and necessary condition for the spatiotemporal intersection of two CAVs is explained in 2.5.2. On the other hand, when stopping at the end of an EMP, the algorithm removes all the STIs that involve the EMP from the list. Note that each of these CAVs has different corresponding time periods indicating when the possible rendezvous may happen. We introduce how to compute the time period in the following Section 2.5.2.

Line	Observed Object List
a (start of 1)	1
b (start of 2)	1, 2, <1, 2>
c (start of 3)	1, 2, <1, 2>, 3, <1, 3>, <2, 3>
d (start of 4)	1, 2, <1, 2>, 3, <1, 3>, <2, 3>, 4, <3, 4>, <2,4>, <1,4>
e (end of 1)	2, 3, <2, 3>, 4, <3, 4>, <2,4>
f (end of 2)	3, 4, <3, 4>
g (end of 4)	3
h (end of 3)	Empty

Outputs: <1, 2>, <1, 3>, <2, 3>, <2, 4>, <3, 4>, <1, 4>

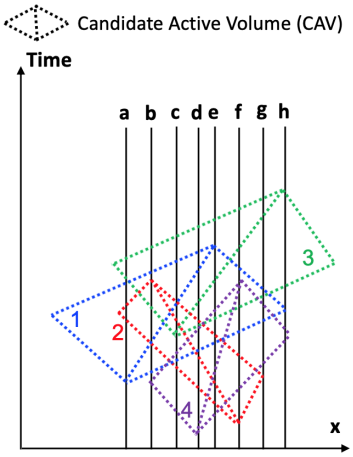


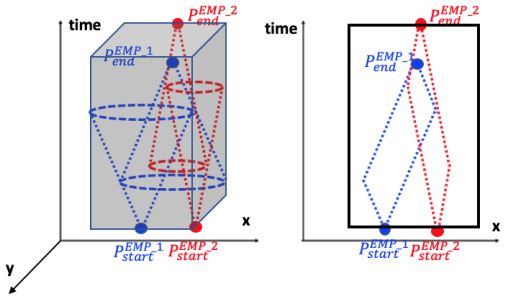
Figure 2.3: Plane sweep execution trace with a test case

An execution trace of Plane Sweep: Figure 2.3 shows a dataset containing EMPs and their corresponding CAVs from four objects. For illustration, we simplify the study area into one-dimensional geographical space. A vertical line sweeps from left to right and stops at the endpoints (from a to h) of each EMP. The table on the left shows the elements in the observed object list after each stop. For example, when stopping at Line d (start of 4), the algorithm determines whether the incoming EMP $< 4 >$ intersects with any element in the observed object list, namely $< 1 >$, $< 2 >$, and $< 3 >$. Hence, $< 1, 4 >$, $< 2, 4 >$ and $< 3, 4 >$ are added to the observed list and the output list. When stopping at Line e (end of 1), the algorithm removes all the elements in the list involving EMP $< 1 >$ which includes $< 1, 2 >$, $< 1, 3 >$, and $< 1, 4 >$. The last stop is at Line h (end of 4). The

Observed Object List becomes empty and the final output STIs include: $< 1, 2 >$, $< 1, 3 >$, $< 2, 3 >$, $< 2, 4 >$, $< 3, 4 >$, and $< 1, 4 >$.

2.5.2 Constructing a Minimum Orthogonal Bounding Region and Spatiotemporal Intersection

In section 2.5.1, we explained the intuition behind the baseline algorithm along with its corresponding execution trace. Now we explain in more detail the creation of Candidate Active Volumes and new Minimum Orthogonal Bounding Regions. A loop goes over all the EMPs with gaps and checks if the given EMP is greater than the missing threshold. If it is, then a new CAV is constructed bounded by the coordinates attained through Maximum Speed and a check is done to see if any other CAV in the observed list intersects with the new given CAV. If it does, a new common MOBR is constructed around the intersection of the two CAVs using their maximum and minimum coordinates. The sufficient and necessary condition of whether two CAVs intersect is discussed later in this section.



	Bead 1 (blue)	Bead 2 (red)
Start	$P_{start}^{EMP\_1} = (x_{start}^{EMP\_1}, y_{start}^{EMP\_1}, t_{start}^{EMP\_1})$	$P_{start}^{EMP\_2} = (x_{start}^{EMP\_2}, y_{start}^{EMP\_2}, t_{start}^{EMP\_2})$
End	$P_{end}^{EMP\_1} = (x_{end}^{EMP\_1}, y_{end}^{EMP\_1}, t_{end}^{EMP\_1})$	$P_{end}^{EMP\_2} = (x_{end}^{EMP\_2}, y_{end}^{EMP\_2}, t_{end}^{EMP\_2})$
Object max speed	S_{max}^{1}	S_{max}^{2}

Figure 2.4: Cone Intersection of Effective Missing Period (EMPs)

As defined in Section 2.5.1, an effective missing period (EMP) is the intersection of two cones (i.e., a bead) vertexed at the endpoints of a CAV [47]. Therefore, in order to construct a new CAV during each iteration of this loop, we first need to determine the coordinates of the end points attained via maximum speed along with their respective radius. As shown in Figure 2.4, we first construct a CAV using maximum speed and radius where the radius of each cone at time t is the product of the object max speed S_{max} and the time difference from the start point of the EMP. For example, the radius of the cone vertexed at $P_{End}^{EMP_1}$ at time t: $r_{end}^{P1} = (t_{end}^{P2} - t) \times S_{max}^{1}$. Then, we find the end points by calculating the distance covered at maximum speed from each start point to its respective

endpoint. This operation is done for both CAVs and creates a common MOBR using the extreme coordinates bounded by each CAV.

To compute the intersection between two EMPs during each iteration of this loop, we need to determine the intersections between four cones (i.e. two beads). We first check if these two beads have an overlapping time range. If not, then the beads are guaranteed to be not intersecting. Otherwise, the following geometric property is used to determine the intersection. Figure 2.4 shows two beads generated from two EMPs. EMP_1 starts at $P_{start}^{EMP_1}$ and ends at $P_{end}^{EMP_1}$, while EMP_2 starts at $P_{start}^{EMP_2}$ and ends at $P_{end}^{EMP_2}$. Each point is represented by three coordinates. For example, start point $P_{start}^{EMP_1} = (x_{start}^{EMP_1}, y_{start}^{EMP_1}, t_{start}^{EMP_1})$ is presented by two spatial coordinates $x_{start}^{EMP_1}$ and $y_{start}^{EMP_1}$ as well as the temporal coordinate $t_{start}^{EMP_1}$. The radius of each cone at time t is the product of the object max speed S_{max} and the time difference from the start point of the EMP. For example, the radius of the cone vertexed at $P_{end}^{EMP_1}$ at time t: $r_{end}^{EMP_1} = (t_{end}^{EMP_1} - t) \times S_{max}^2$. Now, we formulate the sufficient and necessary condition for two beads intersect as follows:

$$r_{start} + r_{end} \leq dis(start, end), \tag{2.7}$$

where index $start = \{start^{EMP_1}, start^{EMP_2}\}$, index $end = \{end^{EMP_1}, end^{EMP_2}\}$, and $dis_{start,end}$ is the Euclidean distance between points P_{start} and P_{end}. Appendex A includes a detailed proof for Equation 2.

2.5.3 Spatio-temporal Grid Traversal Algorithm

Computing possible rendezvous regions is challenging due to the high computational cost over a large set of trajectories. The plane sweep algorithm provides an axis parallel MOBR which reduces the search space for finding the possible rendezvous regions for the refinement phase. However, this baseline approach proves to be inefficient when pruning a set of gap pairs both having higher positional displacement and EMPs. This may result in the construction of a common MOBR with size similar to the entire study area even if the actual rendezvous region between the gap pairs is relatively small compared to their respective MOBRs. Thus, we propose a Spatio-temporal Grid Traversal algorithm that aims to identify possible rendezvous regions using location, time and maximum speed.

Spatio-temporal Grid Traversal (SGT) is based on the idea of a 3D filtering technique by leveraging spatiotemporal properties and additional attributes of the space-time prism model to get a better geometric approximation of a bounded region. SGT starts by creating

a spatiotemporal grid and applying the baseline approach for constructing CAVs of incoming gaps followed by the a common minimum object bounded rectangle. However, inside each MOBR, we compute the linear bounds of the cones generated from the start point and endpoint of the individual gaps and determine which grid cells reside inside the linear bounds of each CAV by checking each of the cell's corner points. Then we check the common cells residing in both CAVs. This operation is performed for every slice in the third dimension. Algorithm 1 shows the pseudo-code for SGT.

Step 1: Create common minimum orthogonal bounding rectangles (MOBRs): First, we create a spatiotemporal grid having the size of the given study area and a specified spatial and temporal resolution. Then, we apply the plane sweep algorithm to get common MOBR around gap pairs and save them in a common MOBR.

Step 2: Create linear bounds within each common MOBR: For each common MOBR inside the CMOBR List, we index its endpoints inside the ST grid along with the start points and end points of the individual gaps to their respective nearest cell. Next we derive CAV linear bounds from the start point and end point of each gap based on the object's maximum speed. The linear bounds can be defined as the geometric interpretation of the slant height of the cone derived from the maximum speed. The maximum speed provides the slope i.e. the angle between the slant height and the parallel y-axis (time).

Step 3: Filter remaining grid cells qualified within CAVs: The filtering step linearly checks whether each cell inside the MOBR resides in the given CAV linear bounds. The linear bounds can be further divided into lower bounds and the upper bounds which are derived from the start point and end point of the gaps respectively. In order to satisfy this condition, at least one of the corner points should be positioned higher than the lower bound but lower than the upper bound for each individual CAV. During filtering, we further refine the intersection by concurrently checking if any of those cells reside in both the CAVs. If they do, we filter out the remaining cells into the Output list.

Time Slicing: Time slicing is an intermediate filtering phase which bounds each cone slice by a rectangle which is tighter than the corresponding slice of the space-time grid, thereby achieving higher efficiency. Hence, increasing the number of slices greater than the spatial resolution extent results in finer pruning that filters out extra region between CAV bounds and MOBR. Algorithm 2 provides a modification of step 2 in Algorithm 1 where

Algorithm 1 Spatio-temporal Grid Traversal (SGT)

1: $Spatial\ Resolution \leftarrow M$
2: $Temporal\ Resolution \leftarrow T$
3: Create Spatiotemporal Grid
4: $CMOBR\ list \leftarrow \emptyset$
5: $Output \leftarrow \emptyset$
6: Apply Plane Sweep Algorithm for Commom MOBRs
7: $CMOBR\ list \leftarrow Common\ MOBRs$
8: **for each:** $CMOBR_i\ in\ CMOBR\ list$ **do**
9: $Index\ CMOBR_i\ over\ Spatial\ Temporal\ Grid$
10: $Calculate\ Linear\ Bounds\ of CAVs$
11: **for each:** $Cell_i\ inside\ Common\ MOBR$ **do**
12: **if** $Cell_i\ resides\ in\ both\ CAVs$ **then**
13: $Cell_i \rightarrow Output\{\}$
14: **end if**
15: **end for**
16: **end for**

we increase the temporal resolution greater than its respective spatial resolution.

Algorithm 2 Spatio-temporal Grid Traversal with Time Slicing (SGT-TS)

1: $Spatial\ Resolution \leftarrow M$
2: $TemporalResolution \leftarrow t > T$
3: Run Algorithm 1 (line: 3 to 16)

Execution trace of SGT and SGT-TS: Figure 2.5 shows the execution trace of Algorithms 1 and 2 over a 2D grid with 64 cells taking the x-axis as longitude (or latitude) and the y axis as time. For a given pair of CAVs, we first create a common MOBR and approximate its end points to their respective nearest cells. Figure 2.5(a) shows a 16x4 example grid with a CAV surrounded by its MOBR approximated by a grid where each grid cell is marked in yellow. During the filtering phase, SGT checks whether the corners of each cell qualify to be included in the given CAV. Figure 2.5 (b) shows the resulting grid after filtering where cells marked in red do not qualify and yellow cells represent the cells residing in the CAV. After getting all the cells inside the MOBRs, we check whether each cell resides in both the CAV pairs and output them as the final ouptut. Figure 2.5 (c) shows the blue cells which take part in the intersection of two CAVs. For SGT with time slicing,

we increase in temporal resolution providing a better filter as compared to the original SGT. Figure 2.5 (d) shows the final grid cells remaining in yellow and discarded cells in red. As compared to SGT, SGT-TS gives more refined results since more extra space has been discarded. Figure 2.5 (e) shows a greater number of intersecting cells, represented in blue, and indicating a better approximation of the intersection area.

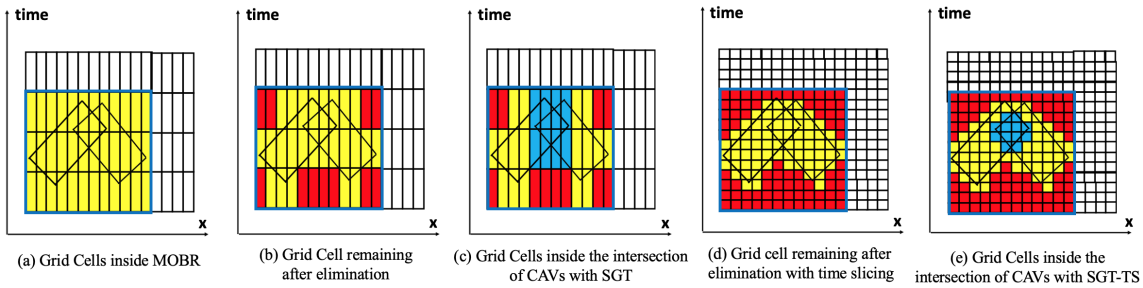


Figure 2.5: Execution Trace of SGT and SGT-ST

2.6 Validation

2.6.1 Experiment Design

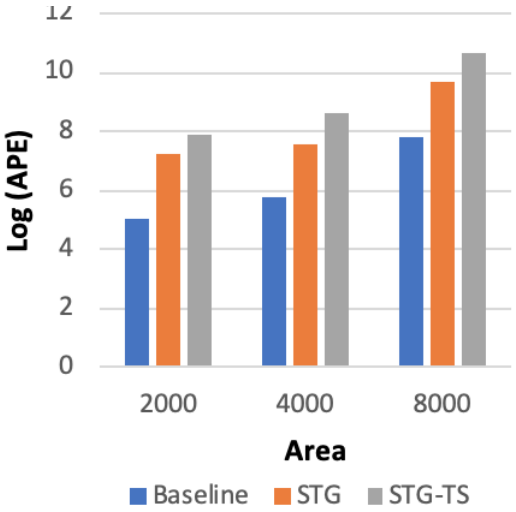
Dataset: The dataset used in the experiments was MarineCadastre [1] which contains records of more than 30 attributes (e.g., Maritime Mobile Service Identity, longitude, latitude, speed over ground (SOG), course over ground (COG) etc.) for 150,000 objects (i.e. ships) taken every minute from 2009 to 2017. The dataset is based on the WGS 1984 coordinates system with a geographical extent of 180W to 66W degrees in longitude and 90S to 90N degrees in latitude covering waters around the US.

Experimental goal: The goal was to evaluate the performance of the proposed baseline, SGT and SGT-TS under different parameters. Our research questions were:

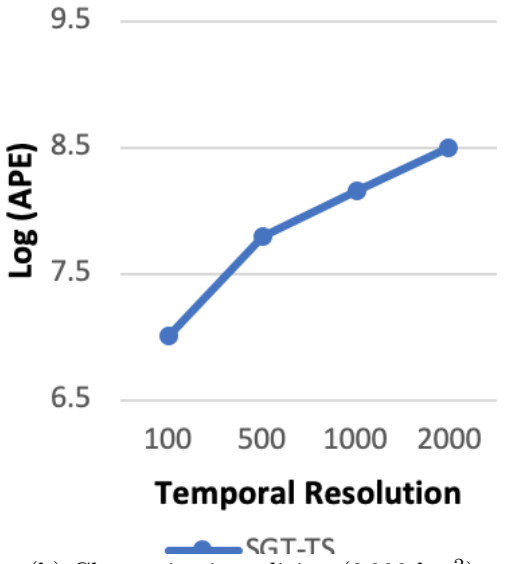
(1) How does the size of the study area affect the APE ? and (2) How does an increase in the number of objects affect the APE on a given study area ?

Our evaluation metric was area pruning effectiveness (APE) which is the ratio of the total study area and minimum bounded area inside the filtered region (i.e. approximate CAV) as discussed in Section 2.2.

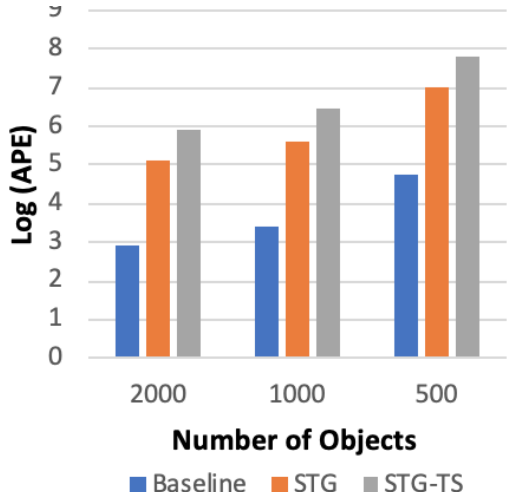
Computing Resources: All the experiments were conducted using Python and performed on an Intel Core i5 2.5GHz CPU and 16GB memory.



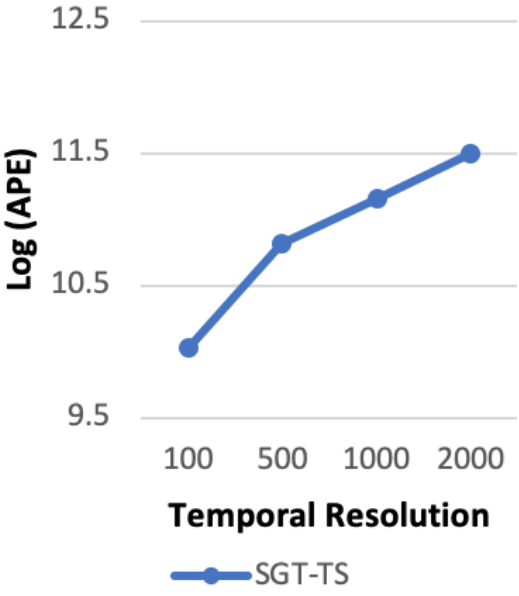
(a) Comparison based on study area



(b) Change in time slicing (8 000 km^2)



(c) Comparison based on object number



(d) Change in time slicing (300 objects)

Figure 2.6: Effects on area-pruning effectiveness (APE)

2.6.2 Experimental Results

Effect of the size of study area: In this experiment, we tested three study area sizes: $2000km^2$, $4000km^2$ and $8000km^2$. The number of GPS points varied for each study area

as 1.5×10^5, 3×10^5, and 6×10^5. The density of GPS points remained consistent for the different areas. The results in Figure 2.6a show that SGT and SGT-TS are always more accurate than the baseline, especially as the study area increases. Figure 2.6b shows the APE values for SGT-TS with increasing time-slicing factor. Again, the SGT-TS algorithm outperforms the baseline with the APE improving as the time-slicing factor is increased.

Effect of the number of objects: In this experiment, we set the study area size to $8000km^2$ and varied the number of object pairs (i.e. ships) from 500 to $2,000$. We also varied the number of GPS points respectively from 2.5×10^4 to 4.5×10^4. This was realized by picking same study area from the original dataset for each varied number of object pairs resulting in different density of GPS points. The results in Figure 2.6c show that the increase in average pruning effectiveness (APE) is significantly greater in SGT and SGT with time slicing as compared to plane sweep for different number of objects. Figure 2.6d shows the further improvement of the APE as we increase the time-slicing factor.

2.6.3 Interpretation of Experimental Results

Scanning an entire study area can be exponentially expensive in terms of computational cost and human effort. The filter phase provides approximate regions within a trajectory gap for filtering possible rendezvous regions. However, the refinement phase can be expensive due to uncertainty in modeling the exact geometry of cones. Exact geometry involves inclusion of many real world physics based parameters (e.g. speed, acceleration) [26] which add complexity due to the need to solve quadratic equations. Hence, our main intuition is to reduce the total refinement cost in terms of computation of each cell per unit area by providing a tighter and more accurate filter in the filtering phase. Equation 2.8 shows the relationship of filter and refinement in terms of computational cost.

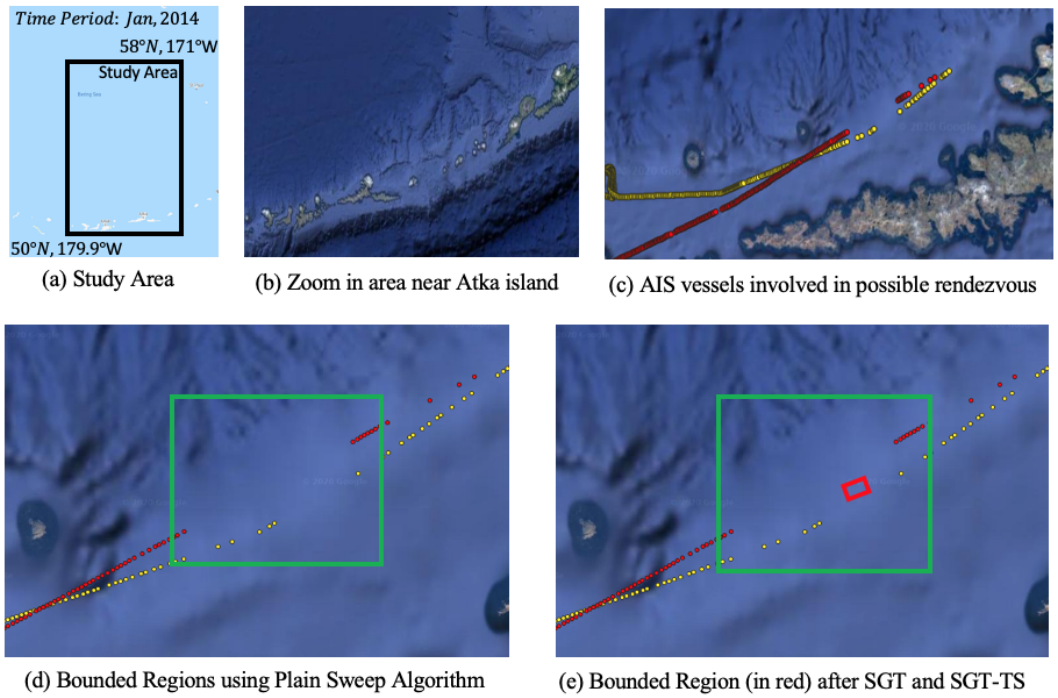
$$C_f + C_r = C_t, \tag{2.8}$$

where C_f is the cost of filtering, C_r is the cost of refinement and C_t is the total cost to prune the given study area. The cost of refinement C_r decreases as we increase the filtering efficiency which often requires high computational cost in terms of prepossessing, model refinement, etc. However, if we are not considering any filtering then C_r will be equal to C_t.

2.6.4 Case Study on Real Automatic Identification System (AIS) data

We conducted a case study on data from MarineCadastre, a popular real world AIS dataset [1] to find possible rendezvous regions using the algorithms proposed in the chapter. The approaches were applied on a study area ranging from $179.9W$ to $171W$ degrees in longitude and from $50N$ to $58N$ degrees in latitude in the Bering Sea, shown in Figure 2.7 (a). The dataset contained ˜1.4×10^6 GPS readings from 72 ships that traveled during January, 2014. The EMP threshold (θ_e) was set to 30 minutes by which only the top 0.5% longest missing periods were considered as EMPs. In this case study, we focus on various different trajectories that formed two STI clusters, one near Idak, and Atka Isalnds (bottom of the figure), and one near St. Paul Island (top of the figure). These clusters accord with reports by Marine Traffic Agency [48] that near an Island, AIS systems tend to switch back and forth between terrestrial-AIS and satellite-AIS. Ships moving across the boundary of the effective zone may put out weak and unstable signals. The STI clusters we identified, which are near islands, likely represent such areas of weak signaling. Figure 2.7 (b) shows the zoomed in region near Atka Island where we selected to study our case. Figure 2.7 (c) shows the voyage of two vessels in that region categorized by a unique identifier (MMSI) each represented by a unique color indicating the start and end points of the EMPs. Figure 2.7(d) shows the output of the plane sweep algorithm, which construct a common MOBR around the CAV's (in green). Figure 2.7 (e) shows the common region (shown in red) after applying SGT and SGT with time-slicing inside the MOBR where both the CAVs intersect, providing a significantly smaller region and better area effective pruning (APE) than the baseline.

Figures 2.8(a)-(c) depict a trajectory analysis for two entities, likely vehicles or robots, focusing on rendezvous planning. We focused on a pair of trajectories located in the same general vicinity, and both trajectories (Trajectory 1 in red, Trajectory 2 in yellow) had missing data. Figure 2.8(a) shows that applying traditional linear interpolation to the missing data produces two straight line trajectories without indicating possible rendezvous activity between the two vessels. Figure 2.8(b) introduces overlapping geo-ellipses (spatiotemporal prisms) along the trajectories, highlighting a "Two Circle Intersect Area" that suggests a potential rendezvous zone where the entities might meet, factoring in time and space uncertainties. Figure 2.8(c) image refines this further by pruning the intersection into a smaller "Actual Area" (green region within the overlap), reducing the



(a) Study Area (b) Zoom in area near Atka island (c) AIS vessels involved in possible rendezvous

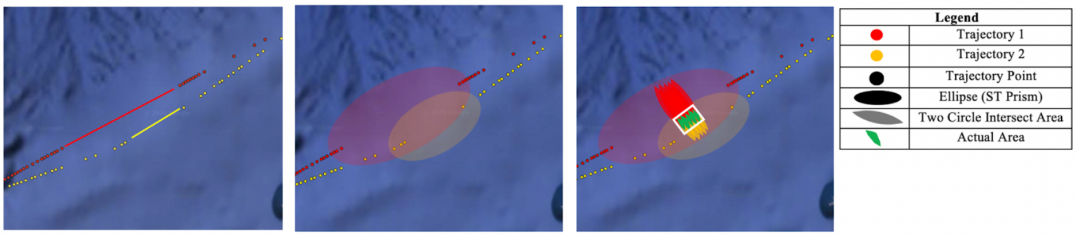
(d) Bounded Regions using Plain Sweep Algorithm (e) Bounded Region (in red) after SGT and SGT-TS

Figure 2.7: Spatio-temporal intersections detected in Bering Sea (best in color). The background imagery is not taken at the same time when the vessels were traveling in January 2014.

rendezvous possibilities to a more precise area, which lowers post-processing efforts for human analysts by providing a clearer, more focused region for coordination.

2.7 Conclusion and Future Work

In this chapter, we introduced the problem of rendezvous detection in trajectory data with gaps. We proposed a baseline algorithm based on a plane sweep approach which first sorts and does a linear scan over a set of gaps and then provides a minimum orthogonal bounding region around the gaps. We proposed a Spatio-temporal Grid Traversal that provides tighter MOBRs, which in turn provides a more approximate shape of the candidate active volume. We further add efficient pruning based on time-slicing by adding a finer temporal resolution that gives a more accurate approximation bounded by the intersection of two cones. The results show relatively better area density ratio in SGT with time slicing as compared to



(a) Linear interpolation (b) Space-Time Interpolation (c) Time Slicing Approach

Figure 2.8: A Comparison of Linear vs Space-Time Interpolation vs Time Slicing Approach

SGT with significantly better APE when compared to the baseline.

Future Work: We plan to further implement the refinement phase where we refine the process of finding the exact geometry of spatiotemporal intersection since it is hard to find the exact geometry of the cone intersect due to its complexity. Computing approximate regions is very expensive in terms of time complexity and modeling them in regional space is challenging. Hence, we plan to further address the computational cost for extracting gaps and extend the proposed work in regional space as described in [44] [45]. We will also study if there an empirical threshold or ratio beyond which the gap may be too large to be meaningfully estimated via space-time prism. In addition, we will create synthetic dataset and then remove data points to make the data coarse to evaluate precision and recall with known rendezvous regions. Finally, analyzing more interesting rendezvous patterns which involve multiple space-time prisms intersection takes place.

Chapter 3

Analyzing Trajectory Gaps to Find Possible Rendezvous Region

3.1 Introduction

In the earlier chapter, we investigated the problem of detecting possible rendezvous between moving objects during the gaps since it may facilitate illegal cargo transfer on the high seas. Unfortunately, the traditional space-time prism method provides a very loose upper bound on possible rendezvous locations during trajectory gaps, resulting in an unnecessarily large manual post-processing effort to cross-check candidate locations with satellite imagery. We further designed a novel time-slicing method to reduce the post-processing effort and provide a tighter upper bound using finer temporal reasoning to explore each time stamp within the trajectory gap. However, the time-slicing method is computationally more expensive, taking hours even for ship trajectories near U.S. waters. In this chapter, we will investigate faster algorithms to reduce the computational cost by designing novel spatial filters based on space-time prism ellipses and time-sliced circle pairs. We provided a mathematical analysis of correctness, completeness, and time complexity. The comparative analysis experiments with synthetic and real maritime trajectory data showed that the proposed approach substantially tightened the geo-area of possible rendezvous and reduced computation time.

The state-of-the-art uses either shortest-path or space-time prism interpolation during

the gap. The shortest-path approach may fail to detect the possible rendezvous if the ship deviates from the shortest path during the gap (the time period of unreported location). Space-time prism approaches provide a loose bound on the geographic area of possible rendezvous. To overcome the limitation of related work, we investigated a time-slicing approach to tighten the geographic area of possible rendezvous regions, as illustrated in Figure 3.1 for two-dimensional Euclidean space. Figure 3.1 (a) shows the input trajectory derived from Object 1 (small solid blue circles) and Object 2 (small solid red circles) traveling from left to right with the data gap during the time-interval t=4 to t=8, which is depicted using small hollow circles representing traditional approach of assessing possible locations via shortest-path interpolation. Assuming a maximum speed of 1 unit/second for simplicity, Figure 3.1 (b) shows the baseline estimation of the area (highlighted in purple) of possible rendezvous, determined by the intersection of space-time prism-based ellipses (shown in light blue and light red) bounding possible locations of object 1 and object 2 during the gap. Figure 3.1 (c) shows the output of the proposed time-slicing method in green. Notice that the green area (Figure 3.1 (c)) is much smaller than the purple area (Figure 3.1 (b)), and this illustrates the superiority of the proposed time-slicing method over the classical space-time prism method.

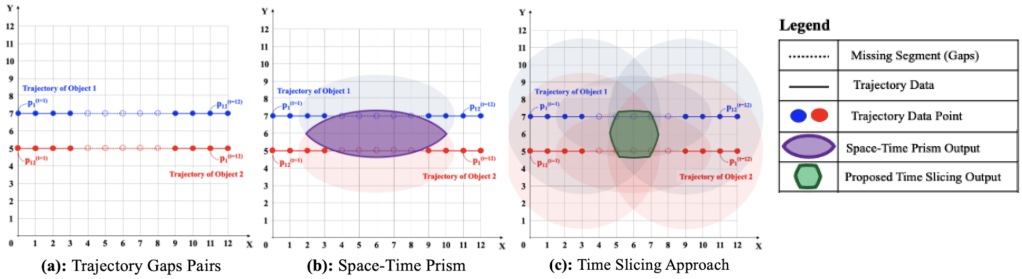


Figure 3.1: Time-slicing approach shrinks the area of possible rendezvous (Best in Color)

Moving objects such as ships often report trajectories with spatial and temporal gaps due to weak GPS signals, instrument malfunction, or malicious interference. Societal applications for analyzing trajectory gaps are related to maritime safety, homeland security, epidemiology, and public safety. For instance, global security concerns such as trans-shipments, illegal oil transfer, illegal fishing etc. are important for maritime safety and regulatory enforcement. Such activities can be restricted and managed by identifying

Table 3.1: Application Domain and Use Cases Examples for Detecting Rendezvous Regions

Application Domain	Example Use Cases
Astrophysics	Manual Tracking of astronomical objects (e.g. comets) which are no longer visible or temporarily lost in space.
Marine Biology	Tracking marine animals (e.g. whales, white sharks) trajectory movements in deep sea where signals tend to get lost for several minutes.
Contact Tracing	Identifying potential areas where people may have had contact with an infected person.
Mapping Coverage Gaps	Understanding potential areas of communication gaps or under capacity.

frequent missing signals from GPS trajectories of oil or fishing vessels. Other use cases from various domain are listed in Table 3.1. However, detecting potential rendezvous of objects such as ships is a **time-intensive process** because it often requires manual inspection of gaps in trajectories spread over a large geographic space. Preliminary work done by human analysts can be substantially reduced with computational methods that narrows the search space to those regions where rendezvous could have occurred.

Traditional trajectory mining approaches use linear interpolation to characterize ship location during the gap and this can lead to many missed patterns. Figure 3.1 illustrate the missing patterns via traditional trajectory mining approaches. If we assume the missing segments of Object 1 and Object 2 in the input are linear interpolations, we forced to conclude the objects never meetup. By contrast, our framework (shown in the output) based on space-time prism models which identifies a region where the objects could have potentially conducted a rendezvous. This results in missing patterns in real world scenarios (e.g. illegal fishing, illegal cargo transfer etc.). Another computational challenge is the large data volume distributed over considerable geographical space. For example, MarineCadastre [1], an automatic identification system (AIS) dataset, contains records with more than 30 attributes (e.g., location, draught) for 150,000 ships taken every minute during the years 2009 to 2017. Its total size is about 600 GB and covers all the waters around the US.

Many techniques implementing spatiotemporal joins (especially on trajectories) based on similarity measures and spatial indexes. For instance, the TS-Join [49] is applied in a

spatial network and considers both spatial and temporal proximity while Time Relaxed Join [50] uses an effective technique to group objects moving in a similar pattern at different times. Other similarity joins [51, 52] perform time-interval threshold techniques such as sampling, sliding window etc. Query processing techniques based on 3D R-Trees have been proposed, such as Spatiotemporal R-Trees and Trajectory-Bundle tree (TB-Tree) [42]. However, the current literature has not addressed trajectory gap uncertainty or efficient pruning strategies inside the spatial join filter.

In this chapter, we analyze the computational cost of determining potential rendezvous regions. The baseline method [53] uses a Linear Temporal Scan (LTS) Time Slicing algorithm inside a space-time grid and further reduces the geographical search space for rendezvous regions through additional time slicing techniques. To improve the time complexity of Linear Temporal Scan, we propose a Temporal Selection Search (TSS) which narrows down the temporal range closer to the actual intersection time range. Then we apply a Static Ellipse Intersection filter and Dynamic Circle Intersection filter to further reduce the computational time. The proposed framework ensures completeness by finding all the possible rendezvous regions such as (i) trajectories that actually met (ii) trajectories which came close but did not meet (iii) trajectories that did not meet. While precision is low due to the large number of false positives (i.e. trajectories which came close but did not meet), recall is high since we do not lose any information.

Contributions :

- Overall, this chapter investigates computational trajectory-gap handling techniques to improve accuracy (e.g. reduce false positives). Experiments on synthetic data show that the accuracy improved by 50 percent as compared to the baseline approach.
- We propose an efficient temporal pruning approach in TSS algorithm providing efficient temporal pruning data gaps modelled based on a space-time prism model. Experiments on real world data show a significant improvement in computations over the baseline approach.
- We also propose spatial filters in a TSS algorithm: (i) Static Ellipse Intersection, (ii) Dynamic Circle Intersection. Static Ellipse intersection provides tighter filtering for more accurately detecting rendezvous and reduces spatial pruning as compared to the baseline approach. Dynamic Circle Intersection further reduces the filtering space for

more efficient pruning during the time slicing operation.

- We evaluate the proposed algorithms theoretically for Correctness, Completeness and Time Complexity of baseline and proposed algorithm.
- This chapter provides experimental evaluation of the proposed algorithm (TSS) based on area pruning effectiveness, computation time, accuracy and precision. The experiments are conducted while tuning different parameters such as Effective Missing Period and GPS readings.

Scope : In this work, we do not study kinetic prisms [26]. The proposed framework has multiple phases but we limit this work to the filter phase. The refinement phase requires input from a human analyst and is not addressed here. Furthermore, the calibration of the cost model parameters is outside the scope of this work. In addition, we do not model rendezvous areas of trajectories without gaps which are involved in intersections. Finally, we do not address the issue of positional accuracy while modeling the trajectories.

Organization : The chapter is organized as follows: Section 3.2 reviews the related work in more detail. Section 3.3 introduces the framework, basic concepts and the problem statement. Section 3.4 describes the baseline approach along with certain key concepts from our previous work [53]. We introduce the proposed Temporal Selection Search algorithm along with key concepts in Section 3.5. Section 3.6 provides theoretical proofs for correctness, completeness and time complexity. Experiment design and results are discussed in Section 3.7. Finally, Section 3.8 concludes the work and briefly lists future work.

3.2 Related Work and Limitations

An extensive amount of trajectory data is being generated via location-acquisition services and serves different research purposes such as anomaly detection, pattern recognition, etc. The survey in [23] gives broad overview of trajectory data mining techniques related to pre-processing, data management, uncertainty, pattern mining etc. Spatiotemporal joins are the most common type of queries for trajectory query processing in real world applications such as mobility analysis, rendezvous studies etc. In addition, there are trajectory similarity joins based on similarity functions such as Edit Distance [54], Longest Common Sub-sequence (LCSS) [55] and DTW [56]. A recent similarity join technique, TS-Join [49], uses spatial and temporal proximity via a two phase divide and conquer algorithm. Strain-Join [57] is

a signature-based similarity join framework, which generates trajectory wise high quality signatures via a similarity metric and then filters out non similar trajectory pairs. Other techniques include a sliding window approach [50, 58] for enumerating candidate pairs and time interval threshold based elimination [51, 52]. Clue-aware trajectory similarity [59] is used over trajectories with silent duration for clustering and aggregation based on behavioural patterns. Regarding query processing for spatiotemporal joins, there are methods based on 3D R-Trees such as Spatiotemporal R-Tree and TB-Tree [42].

Some frameworks [60, 61] employ reconstruction techniques for modeling uncertainty in gaps (or missing signals) in trajectory data. However, many of the techniques assume shortest path discovery or linear interpolation. These approaches are not designed to detect patterns when the trajectories of the moving objects are missing (e.g., due to weak signaling) since the objects may move far from the shortest path. More realistic solutions for modeling uncertainty in gaps are based on geometric and contextual models such as cylindrical [47] and space-time prism models [16, 40] that construct an areal interpolation of the gaps using coordinates and maximum speed of the objects. More recently, a kinetic prism model [26] showed improved estimation by considering other physical parameters such as uncertainty and acceleration. However, applying these models can be computationally expensive. Researchers in [45] also presented the idea of intersecting two uncertainty regions using geometric properties of space-time prism. An alibi query proved to be an effective method for checking whether two space-time prisms intersect. However, their method is limited to theoretical simulations and not used in real world applications such as analyzing generic and behavioral patterns [24] etc. In this chapter, we leverage the geometric properties of space-time prism [45] for real world applications and perform time slicing for better geometric approximation.

3.3 Possible Rendezvous Detection Problem

3.3.1 Framework

Our aim is to identify possible rendezvous regions on a given set of trajectories through a two-phase *Filter* and *Refine* approach. Figure 3.2 shows a representation of the proposed framework. The trajectories are first preprocessed with a given effective missing period

(EMP) threshold θ, whose output is a number of gap pairs greater than θ. These gap pairs then act as input to the framework as shown in Figure 3.2.

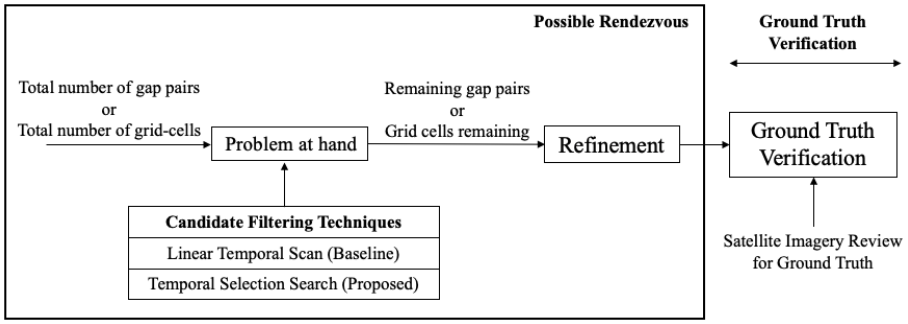


Figure 3.2: Framework for detecting possible rendezvous regions to reduce manual inspection by analyst

The framework includes two algorithms: a baseline Linear Temporal Scan [53], and the proposed Temporal Selection Search (TSS) algorithm. Linear Temporal Scan is a basic time slicing approach based on the union of two minimum orthogonal bounding rectangles. Temporal Selection Search (TSS) expedites the appropriate time range search by calculating the start point and end point of the trajectories where an actual intersection may have occurred. This TSS algorithm further reduces the search space extent by introducing the intersection of two minimum orthogonal bounding rectangles (MOBRs) using 3D R-Tree and two minimum bounding rectangles (MBRs). The output from the filter phase is then refined by finding the exact geometry of the cone intersect and accurately modeling the space time prism of each object. The exact geometry can then be used by human analysts for ground truth verification via satellite imagery.

3.3.2 Basic Concepts

In this section, several key concepts in the rendezvous detection problem are reviewed which help us to understand the problem formulation in next subsection.

Definition 3.3.1. A **study area** is a two-dimensional rectangular complying (latitude, longitude) coordinate system and represents the location of the given input data.

Definition 3.3.2. A **spatial trajectory** is a trace of chronologically sorted points in series generated by a moving object in a geographic space.

For example, $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$ represents a spatial trajectory, $p_i = (x_i, y_i, t_i)$ where each point (p_i) is associated with a geographic coordinate set (x, y) and a time stamp (t).

Definition 3.3.3. An **object maximum speed ($Smax$)** is the maximum speed an object can attain based on the domain knowledge. The variable S_{max} can be identified from publicly available maritime vessel databases [1]. For vehicles, humans, or animals, we can use the maximum physically allowed speed.

Definition 3.3.4. An **effective missing period (EMP)** is a time period when a GPS signal is missing for longer than a user-specified EMP threshold (θ_e). Figure 2.1 shows that the EMPs for Object 1 and Object 2 are between timestamps 3 and 4 with $\theta_e = 2$.

Definition 3.3.5. A **candidate active volume (CAV)** is the spatiotemporal volume based on a space time prism which provides the spatiotemporal bounds for all possible locations of an object during an EMP[28, 29, 30, 16, 31].

3.3.3 Problem Formulation

The problem to optimally identify rendezvous patterns in a spatio-temporal domain is formulated as follows:

Input:

1. A study area S
2. A set of $|N|$ trajectories $T = t_1 \ldots t_{|N|}$, each associated with an object,
3. An object maximum speed (S_{max}) for each object,
4. An effective missing period threshold (θ_e).

Output: Approximate geometry intersection of two gaps.

Constraints: (a) Trajectories have gaps, (b) Maximum acceleration not available.

Objective: Improve area pruning effectiveness (APE) and computational efficiency.

Figure 2.1 shows a one-dimensional representation of two trajectory gaps involving a spatiotemporal intersection via overlap of two Candidate Avoidance Volumes.

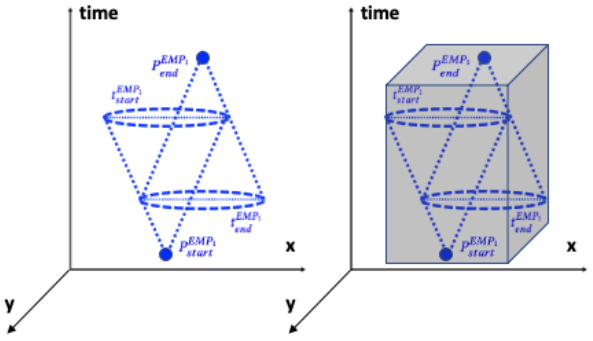
Area pruning effectiveness (APE) is the ratio of the total study area and minimum bounded area inside the filtered region (i.e. approximate CAV).

$$area\ pruning\ effectiveness\ (APE) = \frac{total\ study\ area}{area\ bounded\ inside\ the\ region} \tag{3.1}$$

where, **area bounded inside the region** $\neq \emptyset$. Further, the higher the value of APE, the better the solution quality since the minimum bounded area enclosed within a CAV will be smaller.

3.4 Baseline Approach

Our previous work [53] was based on a time slicing approach for effective pruning of pairs of gaps and improving area pruning efficiency. In this chapter, we take this work as a baseline and propose new spatial and temporal optimizations. The following summary of the baseline method lays the groundwork for our proposed new work that will be explained later in Section 3.5. Section 3.4.1 discuss MOBR creation and Spatiotemporal Intersection Criteria using space-time prism [16] and further approximation via Common MOBR. Extracting potential intersecting gaps (i.e. Candidate Pairs) via Spatiotemporal Intersection Criteria is discussed in Section 3.4.2 and the Linear Temporal Scan algorithm in Section 3.4.3.



	Bead 1 (blue)
Start	$P_{start}^{EMP_1} = (x_{start}^{EMP_1}, y_{start}^{EMP_1}, t_{start}^{EMP_1})$
End	$P_{end}^{EMP_1} = (x_{end}^{EMP_1}, y_{end}^{EMP_1}, t_{end}^{EMP_1})$
Object max speed	$S_{max}^{EMP_1}$

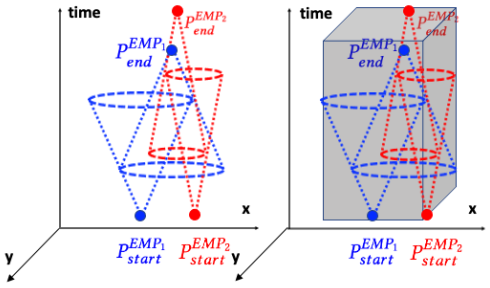
Figure 3.3: Creating a minimum orthogonal bounding rectangle from two cones

3.4.1 MOBR Construction and Spatiotemporal Intersection Criteria

A minimum orthogonal bounding rectangle (MOBR) acts as a base filter for the Linear Temporal Scan (LTS) Algorithm and the Spatiotemporal Intersection Criteria to check if two EMPs intersect. It is generated as follows:

Minimum Orthogonal Bounding Rectangle: Section 3.3.2 defined an effective missing period (EMP) as the intersection of two cones (i.e., a bead) vertexed at the endpoints of a CAV. Therefore, in order to construct a new CAV, we first need to determine

the coordinates of the end points attained via maximum speed and the radius attained during that time elapse. These parameters help us to find the endpoints to create a minimum orthogonal bounding rectangle over a given CAV. For instance, in Figure 3.3, we construct a CAV from point $P_{end}^{EMP_1}$ at any time instance t based on radius r, i.e. from the product of $S_{max}^{EMP_1}$ and time elapsed $(t_{end}^{EMP_1} - t)$. A similar operation is performed at P_{start}^{EMP1} at any time instance with radius $r_{start}^{EMP_1} = (t - t_{start}^{EMP_1}) \times S_{max}^{EMP1}$. To construct a MOBR, we define the radius at time $t = t_{end}^{EMP_1}$ from $P_{start}^{EMP_1}$ or at $t = t_{start}^{EMP_1}$ from $P_{end}^{EMP_1}$. Then, we find the endpoints by calculating the relative radial shift in coordinates from the start and end points. This operation is done for both CAVs. Finally, we take the overall maximum and minimum coordinates from the two CAV's respective endpoints and create a common MOBR (Figure 3.4).



	Bead 1 (blue)	Bead 2 (red)
Start	$P_{start}^{EMP_1} = (x_{start}^{EMP_1}, y_{start}^{EMP_1}, t_{start}^{EMP_1})$	$P_{start}^{EMP_2} = (x_{start}^{EMP_2}, y_{start}^{EMP_2}, t_{start}^{EMP_2})$
End	$P_{end}^{EMP_1} = (x_{end}^{EMP_1}, y_{end}^{EMP_1}, t_{end}^{EMP_1})$	$P_{end}^{EMP_2} = (x_{end}^{EMP_2}, y_{end}^{EMP_2}, t_{end}^{EMP_2})$
Object max speed	$S_{max}^{EMP_1}$	$S_{max}^{EMP_2}$

Figure 3.4: Cone Intersection of Two Effective Missing Period (EMPs)

Spatiotemporal Intersection Criteria: To compute the intersection between two EMPs, we need to determine the intersections between the four cones (i.e. beads) generated from the endpoints of the EMPs. We first check whether any two beads have an overlapping time range. If non do, then the two EMPs are guaranteed to be not intersecting. To be intersecting, there must be at least one timestamp when sections of the two beads intersect i.e. if two beads have an overlapping time range, at least one-time stamp (start-point or end point) of one of the gap segments must be within the time range of the other. To illustrate this, we take two data gaps, EMP_1 and EMP_2 having time range $t_{start}^{EMP_1}$, $t_{end}^{EMP_1}$ and $t_{start}^{EMP_2}$, $t_{end}^{EMP_2}$ respectively and check whether the difference between $(t_{start}^{EMP_1}, t_{end}^{EMP_2})$ or $(t_{end}^{EMP_1}, t_{start}^{EMP_2}) \geq 0$. If it is, then the two EMPs satisfy the necessary condition of a two EMP intersect.

For the sufficient condition, we use the radius information of two cones from a different EMP and check whether their respective radii will overlap with each other. For two circles to overlap, the sum of their radii must be greater than the distance between their respective centers. For instance, the sum of the radius of the cone from start point $r_{start}^{EMP_1}$ of EMP_1 and end point $r_{end}^{EMP_2}$ of EMP_2 must be greater than the distance between their respective radii centers $d(start, end)$. Equation 3.2 provides the sufficient condition for two EMPs to be considered intersecting:

$$r_{start} + r_{end} \geq d(start, end), \tag{3.2}$$

where $start = \{P_{start}^{EMP_1}, P_{start}^{EMP_2}\}$, $end = \{P_{end}^{EMP_1}, P_{end}^{EMP_2}\}$, and $d(start, end)$ is the Euclidean distance between points start and end points. Here we consider the radius generated at the endpoints of the EMPs for MOBR construction.

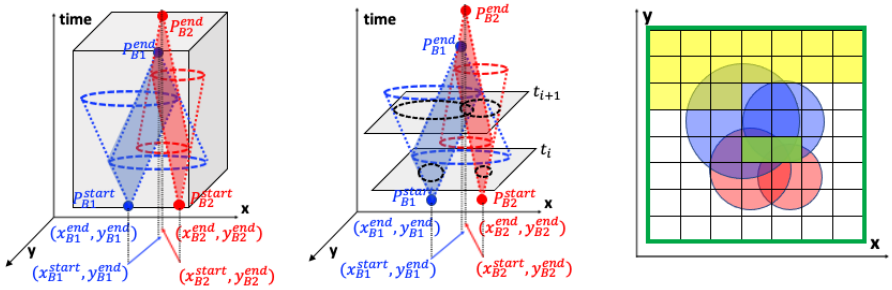
3.4.2 Extracting Potential Intersecting Gap Pairs

In our preliminary work [53], the extraction of gaps was based on a filter and refine approach where a given set of gaps are first sorted to produce potential candidate pairs CP. The idea is to first reduce I/O storage cost in the filter phase and minimizing the computation in the refinement phase. The sufficient and necessary condition for the spatiotemporal intersection of two CAVs and the creation of a minimum orthogonal bounding rectangle is explained in Section 3.4. The final output is then saved in a candidate pair list that will further refined in the time slicing operation for the Linear Temporal Scan algorithm.

3.4.3 Linear Temporal Scan Algorithm

The Linear Temporal Scan (LTS) is based on the idea of a 3D filtering technique that leverages spatiotemporal properties and additional attributes of the ST prism model for a better geometric approximation of a bounded region. This results in a more optimal filter as compared to a minimum orthogonal bounded rectangle area that is relatively larger than the actual intersection area.

LTS starts by extracting $|N|$ potential candidate pairs and saving them in a candidate pair list. Next, it creates a spatiotemporal grid with a specified spatial and temporal threshold. A common temporal range is selected and candidate pairs with a common MOBR are approximated as described in Section 3.4.1. Then, we perform time slicing



(a) Construction of Common MOBR (b) 2-Dimensional Time Slicing using LTS algorithm (c) Resulting grid cells with Static Spatial Filters at timestamp t.

Figure 3.5: Linear Temporal Scan (LTS) Algorithm

operations by generating circular disks D from the common start point to the common end point from both $EMPs$ and check if any grid cell $grid(x, y)$ resides in a given circular disk. If it does, then the grid cell $grid(x, y)$ is saved in an output list. The pseudo-code of LTS algorithm is given in Algorithm 3.

Step 1: Extract Potential Candidate Pairs (MOBRs): First, we extract potential candidate pairs where two CAVs prove to be intersecting as per the necessary and sufficient conditions discussed in Section 3.4.1. The candidate pairs are then saved in a candidate pair list ranging from CP_1 to CP_n using the methodology mentioned in Section 3.4.2. More details related to gap extraction are given in our previous work [53].

Step 2: Create a Spatiotemporal Grid Second, we create a spatiotemporal grid with the size of the given study area and a spatial and temporal resolution m and n respectively.

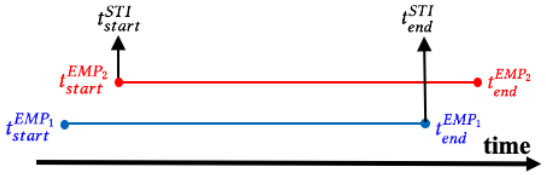


Figure 3.6: Extracting common time interval list

Step 3: Extract Time Interval List We formalize a common time range as $[t^{STI}_{start}, t^{STI}_{end}]$ by $[\max(t^{EMP1}_{start}, t^{EMP2}_{start}), \min(t^{EMP2}_{end}, t^{EMP2}_{end})]$ as illustrated in Figure. 3.6. This operation is followed by indexing of the time interval points and populating the temporal units within t^{STI}_{start}, t^{STI}_{end} in a Time Slice List.

Step 4: Perform Time Slicing: Before the time slicing operation, a spatial approximation is performed over each EMP of a candidate pair (since each CP_i is the union of two $EMPs$) based on a minimum orthogonal bounding rectangle (MOBR) using maximum speed and time elapsed from within [t_{start}^{STI}, t_{end}^{STI}] (shown in Figure 3.5(a)). Then, spatial grid cells $grid(x, y)$ are saved in a Static Spatial Grid. Finally, a time slicing operation is performed for each temporal unit defined by temporal resolution m and start point t_{start}^{STI} and end point t_{end}^{STI}. For each time slice (or temporal unit), we construct Circular Disks $D_{start}^{EMP_1}$, $D_{end}^{EMP_1}$, $D_{start}^{EMP_2}$ and $D_{end}^{EMP_2}$ whose centers are the terminal points of the EMPs and radius as the product of maximum speed $S_{max}^{EMP_1}$ $S_{max}^{EMP_2}$ and time elapsed (as discussed in Section 3.4.1). Then for each $grid(x, y)$ inside a predefined Static Grid Filter, we check whether the given $grid(x, y)$ resides in the intersection of the dynamically constructed Circular Disk (i.e. $D_{start}^{EMP_1} \cap D_{end}^{EMP_1} \cap D_{start}^{EMP_2} \cap D_{end}^{EMP_2}$). If it does, the given $grid(x, y)$ is saved in a $TimeSliceGridList$ which is constructed at each time slice operation. The $TimeSliceGridList$ is also saved in a $LocalGridList$ for each candidate pair. This list is eventually saved in a GlobalGridList representing all the possible rendezvous locations of the given study area.

Execution trace of LTS Algorithm: Figure 3.7 shows the execution trace of Step 4 of Algorithm 3 in a 2-dimensional representation with the intersection of two CAVs over a 12x10 grid. The x-axis represents longitude and the y-axis, time. Prior to time slicing, we first decide the common temporal time range [t_{start}^{STI},t_{end}^{STI}] where STI stands for Spatiotemporal Intersection and t_{start}^{STI},t_{end}^{STI} are the actual times of the intersection from the start and end points followed by a common MOBR approximation. Grid cells $grid(x, y)$ residing in none of the CAVs are shown in red. Cells residing in one of the CAVs are shown in yellow while cells that reside in both CAVs are colored blue. As shown in Figure 3.7 (a), no grid cell $grid(x, y)$ resides in any of the CAVs at time $t = 1$, whereas in a subsequent iteration, some cells do reside in one of the CAVs at t = 2 (shown in Figure 3.7(b)). Figure 3.7 (c) and (d) show the common grid cells inside both CAVs in blue from time $t = 5$ to $t = 8$. These common grid cells serve as the intersection of 4 circular discs D_{start}^{EMP1}, D_{start}^{EMP1}, D_{start}^{EMP2}, D_{start}^{EMP2} in 3 dimensional space as shown in Figure 3.5(c) where the green cells are the common intersection of the 4 circular discs (i.e. blue cells in Figure 3.7 (c), (d) and (e)).

Algorithm 3 Linear Temporal Scan (LTS) Algorithm

Input:
- A Study Area S
- A set of $|N|$ Gap Pairs with given EMPs and Maximum Speed
- Spatial Resolution $\leftarrow m$
- Temporal Resolution $\leftarrow n$

Output:
- GlobalGridList $\in$ Total Intersection Area of Candidate Pair List

procedure :
  Initialize GlobalGridList $\leftarrow \emptyset$
  Step 1: Extract Potential Candidate Pairs
  Candidate Pair List $\leftarrow [CP_1, CP_2, ..., CP_n]$
  Step 2: Create Spatiotemporal Grid
  Spatial grid $\leftarrow \{grid(0,0), ...grid(m,m)\}$
  Temporal grid $\leftarrow \{grid(x,y,0), ...grid(x,y,n)\}$
  for each: $CP_i \in$ Candidate Pair List **do**
    Initialize LocalGridList $\leftarrow \emptyset$
    $CP_i \in \{EMP_1 \cup EMP_2\}$
    Step 3: Extract Time Interval List
    $[t_{start}^{STI}, t_{start}^{STI}] = [\max(t_{start}^{EMP1}, t_{start}^{EMP2}), \min(t_{end}^{EMP1}, t_{end}^{EMP2})]$
    Time Slice List $\rightarrow [t_{start}^{STI}, t_{end}^{STI}]$
    Step 4: Perform Time Slicing
    Static Filter $\leftarrow$ Common MOBR Filter (EMP_1,EMP_2,$S_{max}^{EMP_1}$,$S_{max}^{EMP_2}$)
    Static Spatial Grid $\leftarrow grid(x,y) \in$ Static Filter
    for each: Time Slice $\in$ Time Slice List **do**
      Initialize $TimeSliceGridList_i \leftarrow \emptyset$
      Construct Circular Discs D_{start}^{EMP1}, D_{start}^{EMP1}, D_{start}^{EMP2}, D_{start}^{EMP2}
      for each: $grid(x,y)$ in Static Spatial Grid **do**
        if $grid(x,y) \in [D_{start}^{EMP1} \cap D_{end}^{EMP1} \cap D_{start}^{EMP2} \cap D_{end}^{EMP2}]$ **then**
          $TimeSliceGridList_i \leftarrow grid(x,y)$
        end if
      end for
      $LocalGridList_i \leftarrow TimeSliceGridList_i$
    end for
    GlobalGridList $\leftarrow LocalGridList_i$
  end for
end procedure

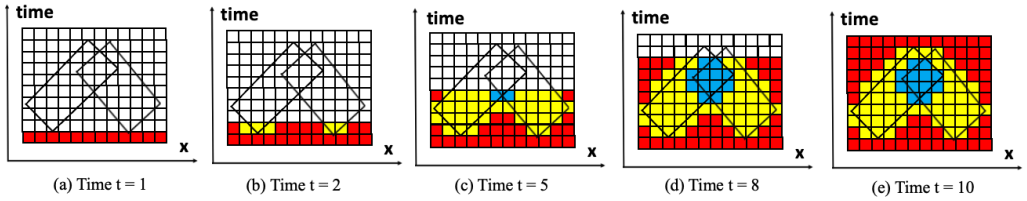


Figure 3.7: Execution Trace of Linear Temporal Scan (LTS) algorithm

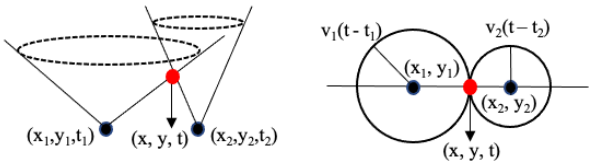
3.5 Proposed Approach

Linear Temporal Scan (LTS) provides fine pruning to help a human analyst to scan minimal approximate areas for refinement. However, the baseline LTS technique proves to be inefficient in terms of computational time when we increase spatial and temporal resolution. In our new work, we propose a Temporal Selection Search (TSS), a temporal optimization method based on the concept of a space-time prism intersection [45] which helps us to get prior temporal information on when two space time prisms actually met. Section 3.5.1 explains the basic intuition and concept for finding a smaller actual time range as compared to the baseline approach. In addition, spatial optimizations using 3D R-Trees and minimum bounding rectangle intersection along with Dynamic Circle Intersection are also proposed. Section 3.5.2 describes spatial optimization across each temporal unit based on (i) a Static Ellipse Intersection Filter and (ii) a Dynamic Circle Intersection Filter which significantly reduces the spatial bounds, resulting in tighter pruning. Finally Section 3.5.3 formalizes the Time Prioritize Approach and proposed spatial optimization.

3.5.1 Time Prioritized based Temporal Selection Pruning

The time interval of a Linear Temporal Scan from t_{start}^{STI} to t_{end}^{STI} is based on a common time selection as discussed in Section 3.4.3. However, selecting a common time may not be efficient in terms of overall computational time especially when the temporal range in t_{start}^{STI} to t_{end}^{STI} is not significant. Therefore, we incorporate the concept of finding a temporal range using space-time prism intersection [45] in Step 3 of Algorithm 3 when selecting the appropriate range of t_{start}^{STI} to t_{end}^{STI}.

For instance in Fig 3.8 (a), (x_1, y_1, t_1) and (x_2, y_2, t_2) are the start points of two space-time prisms whose intersection can be imagined as the two dimensional intersection of 2 circles at point (x, y, t) (as shown in 3.8 (b)). However, in a realistic scenario, the actual



(a) Intersection of space-time prisms in **3D** at time **t**

(b) Intersection of space-time prisms in **2D** at time **t**

Figure 3.8: Intersection of two space time prism

intersection point or initial contact (x, y, t) happens much later than either prisms start points (x_1, y_1, t_1) and (x_2, y_2, t_2). Hence, prior knowledge of the time interval when a prism actually intersect can also contribute toward saving multiple time slicing operation resulting in early termination.

Section 3.4.1 briefly summarized the sufficient condition for a Spatiotemporal Intersection by comparing the distance of two points with their respective individual radii using the equation:

$$r_{start} + r_{end} \geq d(start, end), \tag{3.3}$$

where $start = \{P_{start}^{EMP_1}, P_{start}^{EMP_2}\}$, $end = \{P_{end}^{EMP_1}, P_{end}^{EMP_2}\}$, and $d(start, end)$ is the Euclidean distance between points start and end points.

In order to satisfy this condition, we use the radius information of two cones from a different gap and check whether their respective radii will overlap with each other. Two circles are considered overlapping if the sum of their radii is smaller than the distance between their respective centers. For finding the initial contact, we check whether the two circles touch each other (e.g. Figure 3.8) by checking if the sum of their radii is **equal** to the distance between their respective centers. This motivates a search for a new start time t_{start}^{STI} and end time t_{end}^{STI} interval by deriving the terms and solving Equation 3.3:

$$t \geq \frac{d(start, end) + t_{start} \times S_{max}^{EMP_1} + t_{end} \times S_{max}^{EMP_2}}{S_{max}^{EMP_1} + S_{max}^{EMP_2}} \tag{3.4}$$

where $start = P_{start}^{EMP_1}$ and $end = P_{start}^{EMP_2}$

$$t \leq \frac{-d(start, end) + t_{start} \times S_{max}^{EMP_1} + t_{end} \times S_{max}^{EMP_2}}{S_{max}^{EMP_1} + S_{max}^{EMP_2}} \tag{3.5}$$

where $start = P_{end}^{EMP_1}$ and $end = P_{end}^{EMP_2}$

$$t_{start}^{STI} = \frac{d(start, end) + t_{start} \times S_{max}^{EMP_1} + t_{end} \times S_{max}^{EMP_2}}{S_{max}^{EMP_1} + S_{max}^{EMP_2}},$$
$$t_{end}^{STI} = \frac{-d(start, end) + t_{start} \times S_{max}^{EMP_1} + t_{end} \times S_{max}^{EMP_2}}{S_{max}^{EMP_1} + S_{max}^{EMP_2}}. \tag{3.6}$$

$$t_{start}^{STI} \leq t \leq t_{end}^{STI} \tag{3.7}$$

Hence, if the two space-time prisms are found to intersect, then t_{start}^{STI} and t_{end}^{STI} acts as a new time interval in Step 3 of Algorithm 3 when two space time prisms actually intersect.

3.5.2 Spatial Filter for Temporal Selection Search Algorithm

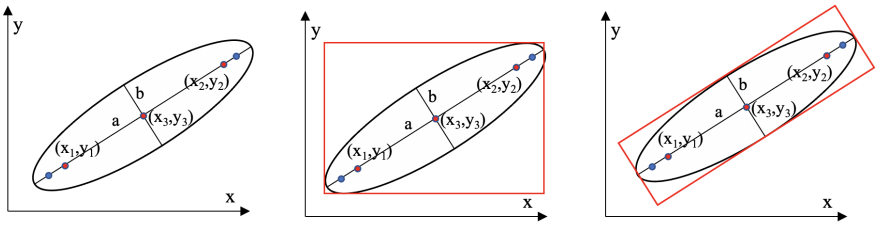
The spatial approximation of space-time prisms via a common MOBR was proposed in the baseline approach can be further optimized by leveraging the geometric properties of the prisms. The main intuition is to minimize the $grid(x, y)$ comparison with $[D_{start}^{EMP1} \cap D_{end}^{EMP1} \cap D_{start}^{EMP2} \cap D_{end}^{EMP2}]$ in Step 4 of the Algorithm 3. Hence we propose two spatial optimization techniques, (i) Static Static Ellipse Intersection (ii) Dynamic Circle Intersection.

Static Ellipse Intersection: Given the start and end points of the EMP, a space-time prism can be projected onto an $x - y$ spatial grid [16] via an ellipse which spatially delimits the extent of a moving object's mobility with maximum speed v_{max}. In Figure 3.9 (a), start point (x_1,y_1,t_1) and end point (x_2,y_2,t_2) with $t_1 < t_2$ are bordered by an ellipse with focii (x_1, y_1) and (x_2, y_2). The equation for the ellipse represented in Figure 3.9 can be stated as :

$$\sqrt{(x - x_1)^2 + (y - y_1)^2} + \sqrt{(x - x_2)^2 + (y - y_2)^2} \leq (t_2 - t_1) \times v_{max} \tag{3.8}$$

where (x_1, y_1) and (x_2, y_2) are the focii and (x, y) is any given point on the ellipse. The major axis a is the product of the total distance travelled with given time interval $t_2 - t_1$ and maximum speed v_{max} divided by 2 (i.e. $a = \frac{v_{max} \times (t_2 - t_1)}{2}$). Minor axis length b can be derived by subtracting $v_{max}(t_2 - t_1)$ with the Euclidean distance from (x_1, y_1) and (x_2, y_2). Hence, defining a basic spatial approximation such as a MOBR or MBR around ellipses

proves to give us a much tighter bound, resulting in fewer cells as compared to the common MOBR used for the baseline in Section 3.4.1.



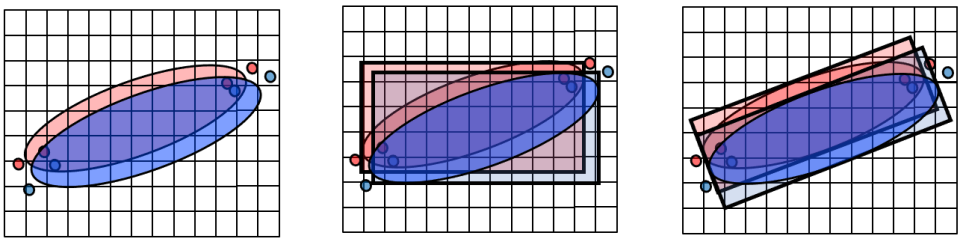
(a) Elliptical representation of ST Prism for a given segment

(b) Spatial MOBR creation using 3D R-Tree for a given segment

(c) Spatial MBR creation using major-minor axis for a given segment

Figure 3.9: Different spatial approximation filter around static ellipses

Figure 3.9 (b) shows an approximating ellipse with a minimum orthogonal bounding rectangle via a 3D R-Tree. The approximation is done by calculating the end points of the major axis using midpoint (x_3, y_3), a and slope $m_1 = \frac{(y_2 - y_1)}{(x_2 - x_1)}$. Using major axis endpoints as diagonals and time elapsed, we generate a 3D Minimum Orthogonal Bounding Rectangle with the help of 3D R-Trees. Figure 3.9 (c) shows an approximating ellipse with a Minimum Bounding Rectangle (MBR). The Minimum Bounding Rectangle (MBR) can be constructed using midpoint (x_3, y_3), the end points of the major axis and the perpendicular slope to the major axis which was calculated while constructing the MOBR (i.e. $-[\frac{1}{m_1}]$). This provides even better spatial approximation as compared to 3D R-Trees intersection with an even fewer number of grid cell comparisons.



(a) Ellipse Creation for Trajectory Gaps

(b) MOBR Approximation via 3D R-Tree

(c) MBR Approximation

Figure 3.10: Ellipse Intersection via Space-Time Prisms

For a rendezvous of two or more objects, calculating the intersection of such a spatial

approximation (e.g. Figure 3.10 (a)) is more computationally efficient than performing an approximation based on a common MOBR (as described in Section 3.4.1). For instance, the intersection of the two ellipses shown in Figure 3.10 (b) shows that the intersection of two MOBRs approximated via the intersection of ellipses using a 3D R-Tree provides a better approximation compared to a common MOBR. Further in Figure 3.10 (c), the intersection of the newly approximated MBR of the two ellipses provides an even better bound, resulting in even fewer grid cells compared to 3D R-Trees and Baseline. This helps identify rendezvous regions beyond the base filter with fewer $grid(x, y)$ cells and provides more accurate rendezvous regions than the MOBR, which will further reduce false positives.

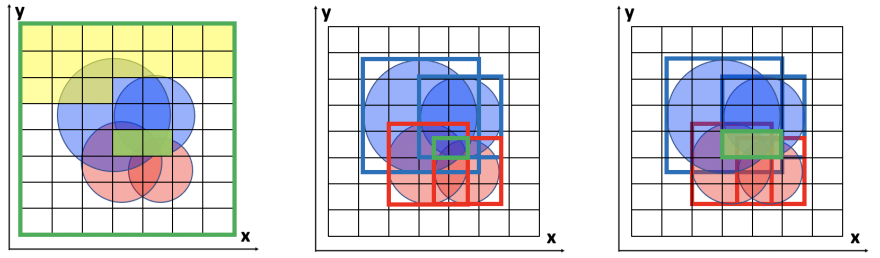
Dynamic Circle Intersection: As noted in Section 3.4, given start point (x_1, y_1, t_1) and end point (x_2, y_2, t_2) with speed v, the cones at each temporal unit t $(t_i < t < t_{i+1})$ are represented as two dimensional circles. Such information provides dynamic spatial bounds throughout the temporal life-cycle of a space time prism. Hence, the uncertainty of a space-time prism of a given CAV can be summarized by 3 polynomial inequalities [16]:

$$t_i \leq t \leq t_{i+1} \tag{3.9}$$

$$(x - x_i)^2 + (y - y_i)^2 \leq (t - t_i)^2 v^2 \tag{3.10}$$

$$(x - x_{i+1})^2 + (y - y_{i+1})^2 \leq (t_{i+1} - t)^2 v^2 \tag{3.11}$$

where, Inequality (3.9) defines the time range defined by the effective missing period EMP of a CAV and Inequality (3.10) and Inequality (3.11) provide circle equations within the ST Prism from the start point and end point respectively. In the case of finding a Rendezvous area, we perform a spatiotemporal intersection join of two space time prisms as discussed in Section 3.4. Hence, each grid cell needs to qualify under Inequality (3.10) and (3.11) for both space time prisms for the rendezvous area. However, the baseline approach proves expensive since creation of a common MOBR depends on the maximum radius generated from the endpoints for both CAVs for a large number of $grid(x, y)$ comparisons. This motivates us to reduce the spatial bounds at each time stamp based on the intersection of 4 circles (i.e. Inequalities (3.10) and (3.11) of both ST Prisms) resulting in a significant decrease in the number of comparisons.



(a) Resulting grid cells with Static Spatial Filters at timestamp **t** .

(b) Intersection outline with Dynamic Spatial Filters at timestamp **t** .

(c) Resulting grid cells with Dynamic Spatial Filters at timestamp **t** .

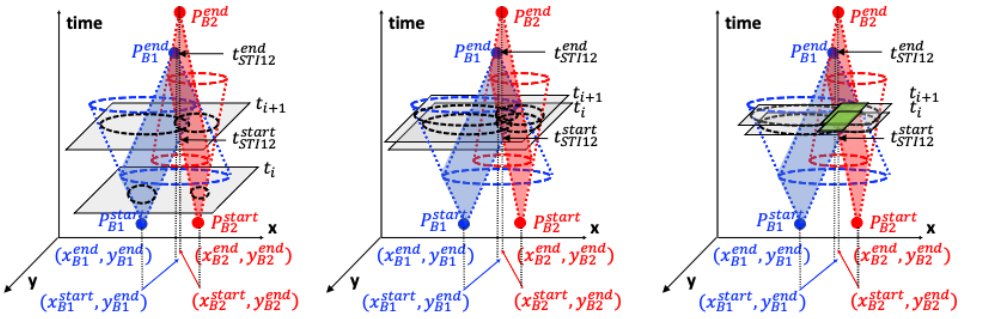
Figure 3.11: Different Spatial Approximations for Dynamic Circular Intersection

For instance, Figure 3.11 shows a comparison operation to determine whether each grid cell $grid(x, y)$ resides in all four circles at a given time stamp t. Figure 3.11 (a) shows each cell $grid(x, y)$ (yellow) and the actual grid cells residing in all 4 circles (green). Figure 3.11 (b) and Figure 3.11 (c) show intermediate operation of spatial approximation by Dynamic Circle Intersection where Figure 3.11 (b) shows the intersection of MOBRs (approximated around the each circle) resulting in smaller MOBR which is a significantly smaller rectangular area than given by the LTS time slicing operation. The MOBR bounds are then indexed on a static filter (e.g. 3D R-Tree or Ellipse intersection). This results in a shorter comparison operation when the resulting intersection rectangle is indexed to the nearest grid cells as shown in Figure 3.11 (c).

3.5.3 Temporal Selection Search Algorithm

Temporal Selection Search (TSS) extends the idea of the Linear Temporal Scan (LTS) by leveraging the spatiotemporal and geometric properties of space-time prism models for optimal geometric approximation. Temporal Selection Search starts with a similar approach as the baseline i.e. finding potential intersection candidates and constructing a space-time grid. A temporal range is selected using the time prioritizer approach (discussed in Section 3.5.1) to minimize the number of linear temporal scan operations. Finally, a spatial optimization method such as Dynamic Circle Intersection Filtering (discussed in Section 3.5.2) is performed to further minimize the number of $grid(x, y)$ comparisons, resulting in a lower time cost.

Figure 3.12 shows a comparison of the proposed Temporal Selection Search algorithm with the baseline Linear Temporal Scan approach. In the baseline approach, the



(a) Time Slicing with Linear Temporal Scan Algorithm

(b) Time Slicing with Temporal Selection Scan Algorithm

(c) Spatial Optimization and Time Slicing with Temporal Selection Scan Algorithm

Figure 3.12: Comparison of Temporal Selection Search algorithm with Baseline Approach

time slicing operation is done on timestamps in time interval range [max(t_{start}^{EMP1},t_{start}^{EMP2}), min(t_{end}^{EMP2},t_{end}^{EMP2})] involving time slicing operations (e.g. Figure 3.12(a)) where no actual intersection of circular discs occurs. Temporal Selection Search provides a better pruning strategy by using a Time Prioritizer (Equations 3.4 and 3.5) to get t_{start}^{STI}, t_{end}^{STI}, significantly reducing the time slicing operation as shown in Figure 3.12 (b). In addition, spatial filtering is also applied using the Static Elliptical and Dynamic Circle Intersection Filters providing a significantly tighter filter than baseline technique (shown in Figure 3.12 (c)). Algorithm 4 shows the pseudo-code for TSS.

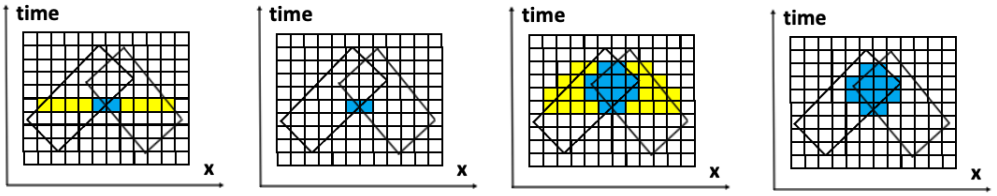
Step 1 and Step 2 are similar to Algorithm 3.

Step 3: Extract Time Interval List: In this step, the Time Prioritizer extracts a relatively smaller time interval range, i.e. from $t_{start}^{EMP_1}$ to $t_{end}^{EMP_2}$. Then the time interval points are indexed via a linear search [t_{start}^{STI},t_{end}^{STI}] and enumerated in the Time Slice List.

Step 4: Perform Time Slicing: This final step is similar to the baseline approach except for some extra operations related to spatial approximation and intersection. In Step 11, we approximate the proposed space-time prism with static elliptical bounds with a MBR by leveraging geometric properties of the ellipses. This involves construction of tighter bound ellipse with physics based parameters e.g. speed, time elapsed for each missing segment in the pair and will result in a smaller search space i.e. fewer grid cell comparisons (discussed in Section 3.5.2). The proposed static ellipse filter then serves as the base filter for the time slicing operation. During time slicing, we construct 4 disks for each time slice and

check that their intersection (i.e. $D_{start}^{EMP1} \cap D_{end}^{EMP1} \cap D_{start}^{EMP2} \cap D_{end}^{EMP2}$) is not empty. If it isn't, then we construct a spatial approximation such as a bounding box around each disk and perform an intersection to get Intersection MOBR Bounds. These bounds are then indexed in the base filter and the resulting grid cells are saved as a Dynamic Spatial Grid variable. Finally we check if any $grid(x, y)$ resides in the intersection of the original 4 discs and qualifying grid cells are saved inside $TimeSliceGridList_i$, which is further saved in $LocalGridList$ for each candidate pair. A GlobalGridList then saves the $LocalGridList$, representing the rendezvous regions of the entire study area.

Execution trace of Temporal Selection Search Algorithm: Figure 3.13 shows the execution trace of Step 4 for the Algorithm 4 representation with the intersection of two CAVs over a 12x10 grid where the x-axis represents longitude and the y-axis, time. Before proceeding to time slicing, a time range $[t_{start}^{STI}, t_{start}^{STI}]$ is decided by the Time Prioritizer algorithm, providing a smaller estimation as compared to baseline. In the figure, grid cells residing in only one of the CAVs are colored yellow while cells residing in both CAVs are blue. The time slicing operation starts at time t = 5 (as shown in Figure 3.13 (a)) with the static spatial filter and a common circle intersection area (yellow) in any of the CAVs. Figure 3.13 (b) shows the circle intersection filter providing tighter bounds compared to the static filter, resulting in filtering minimal number of cells. A similar operation is shown in Figure 3.13 (c) and (d) for time $t = 8$, where the time slicing operation terminates.



(a) Time **t** = 5 with Static Ellipse Intersection Filter

(b) Time **t** = 5 with Dynamic Circle Intersection Filter

(c) Time **t** = 8 with Static Ellipse Intersection Filter

(d) Time **t** = 8 with Dynamic Circle Intersection Filter

Figure 3.13: Execution Trace of Temporal Selection Search Algorithm

3.6 Theoretical Evaluation

In this section, we present a theoretical evaluation of the proposed Temporal Selection Search (TSS) algorithm and show that it is complete but not always correct due to the large presence of false positives in the results. We also provide a brief time complexity

Algorithm 4 Temporal Selection Search Algorithm (TSS)

Input:
- A Study Area S
- A set of $|N|$ Gap Pairs with given EMPs and Maximum Speed
- Spatial Resolution $\leftarrow m$
- Temporal Resolution $\leftarrow n$

Output:
- GlobalGridList $\in$ Intersection Area of $CPairList$

procedure :
  Initialize GlobalGridList $\leftarrow \emptyset$
  Step 1 and 2 are same as Algorithm 1
  for each: $CP_i \in CPairList$ **do**
    Initialize $LocalGridList_i \leftarrow \emptyset$
    $CP_i \in \{EMP_1 \cup EMP_2\}$
    Step 3: Extract Time Interval List
    $[t^{STI}_{start}, t^{STI}_{end}]$ = Time Prioritizer (EMP_1,EMP_2,$S^{EMP_1}_{max}$,$S^{EMP_2}_{max}$)
    Time Slice List $\rightarrow [t^{STI}_{start}, t^{STI}_{end}]$
    Step 4: Perform Time Slicing
    Static Filter $\leftarrow$ Optimized Spatial Filter (EMP_1,EMP_2,$S^{EMP_1}_{max}$,$S^{EMP_2}_{max}$)
    Static Spatial Grid $\leftarrow grid(x, y) \in$ Static Filter
    for each: Time Slice $\in$ Time Slice List **do**
      Initialize $TimeSliceGridList_i \leftarrow \emptyset$
      Construct Circular Discs D^{EMP1}_{start}, D^{EMP1}_{end}, D^{EMP2}_{start}, D^{EMP2}_{end}
      if $[D^{EMP1}_{start} \cap D^{EMP1}_{end} \cap D^{EMP2}_{start} \cap D^{EMP2}_{end}] \neq \emptyset$ **then**
        Disk MOBR List $\leftarrow$ SpatialApproximation (D^{EMP1}_{start}, D^{EMP1}_{end}, D^{EMP2}_{start}, D^{EMP2}_{end})
        Intersection MOBR Bounds $\leftarrow$ MOBRIntersection (Disk MOBR List)
        Index Intersection MOBR Bounds with Static Spatial Grid
        Dynamic Spatial Grid $\leftarrow grid(x, y) \in$ Indexed Intersection MOBR
        for each: $grid(x, y)$ in Dynamic Spatial Grid **do**
          if $grid(x, y) \in [D^{EMP1}_{start} \cap D^{EMP1}_{end} \cap D^{EMP2}_{start} \cap D^{EMP2}_{end}]$ **then**
            $TimeSliceGridList_i \leftarrow grid(x, y)$
          end if
        end for
        $LocalGridList_i \leftarrow TimeSliceGridList_i$
      end if
    end for
    GlobalGridList $\leftarrow LocalGridList_i$
  end for
end procedure

analysis for extracting gaps in Section 3.6.2.

3.6.1 Correctness and Completeness

Lemma 3.6.1. Given a finite set of trajectories, Temporal Selection Search terminates in finite time.

Proof. Given a finite set of $|K|$ trajectories, each trajectory may have a maximum N number of finite points. Therefore, we have a maximum bound of $|K| \times |N|$ finite trajectory points to process for finite candidate pair generation. Hence, for a finite number of trajectories and gap pairs, a finite number of operations will be performed for the Temporal Selection Search algorithm.

□

Lemma 3.6.2. Temporal Selection Search is complete.

Proof. A search algorithm is considered complete if it returns all true positive cases. This may include some cases with false positives but no false negatives. The temporal bounds of TSS are based on Effective Missing Periods of both objects is the subset of the time interval range $[t^{STI}_{start}, t^{STI}_{end}]$ generated at Step 12 of LTS algorithm. Further, the spatial bounds of a rendezvous region at Step 18 will be a subset of the proposed base-filter (MBR) for each time slice from $[t^{STI}_{start}, t^{STI}_{end}]$. Hence TSS will consider all possible rendezvous cases are follows:

(i) When a rendezvous actually occurs: TSS return all true positives (i.e. identifies positive rendezvous when actual rendezvous take place). This includes cases when objects appear to cross paths or remain very close to each other, generating a guaranteed intersection of spatial filters, which results in a rendezvous region being returned.

(ii) When objects try to approach but do not meet: TSS may return false positives (i.e. identifies positive rendezvous when no actual rendezvous take place). This may occur when objects are so close, they appear to meet, which may result in the intersection of spatial filters and subsequent generation of a rendezvous region.

(iii) When objects do not meet: TSS may return false positives or true negative (i.e. identify a negative rendezvous when no actual rendezvous take place). This may occur when objects are far apart or have smaller overlapping time range, since, in certain cases, spatial filters will do interact, generating a rendezvous region.

In all three cases, rendezvous regions are created on top of spatial filters which act as a super-set of the maximum extent of the spatial bounds at each time slice of an actual rendezvous region. This also leads to a large number of false positives, which may result in low precision or accuracy but due to no false positive cases, recall will remain perfect. Hence TSS is complete but not always correct. □

3.6.2 Asymptotic Time Complexity Analysis

Extracting Candidate Gap Pairs: Given $|K|$ unique trajectories each associated with $|N|$ trajectory points, there are $|K| \times |N-1|$ total number of gap pairs. For generating candidate pairs, $\binom{|K|}{2}$ trajectories must be selected for potential candidate pairs where each such pair will have $|N-1| \times |N-1|$ comparisons. Hence, the total number of comparisons will be $\binom{|K|}{2} \times |N-1| \times |N-1|$ which result in asymptotic worst case of $O(|K|^2 \times |N|^2)$.

3.7 Experimental Evaluation

In this section, we evaluated the proposed Temporal Selection Search algorithm by comparing it against the baseline Linear Temporal Scan. Section 3.7.1 provides brief overview of Experiment Design along with Comparative Study, Sensitivity Analysis and dataset description. Section 3.7.2 shows comparison of baseline and proposed method over each performance metrics.

3.7.1 Experiment Design

The goal is to evaluate and compare the proposed Temporal Selection Search approach against the baseline Linear Temporal Scan under different parameters. The performance metrics were area pruning effectiveness (APE), computation time, accuracy and precision. Figure 3.14 shows detailed Experiment Design.

Comparative Study: We compared Temporal Selection Search (TSS) and Linear Temporal Scan (LTS) via experiments under fixed set of parameters such as Effective Missing Period and number of GPS points. The experiments are based on performance metrics such as Area Pruning Effectiveness (APE), Computation Time, Accuracy and Precision. Then we varied these parameters to perform sensitivity analysis where TSS clearly outperforms the baseline LTS.

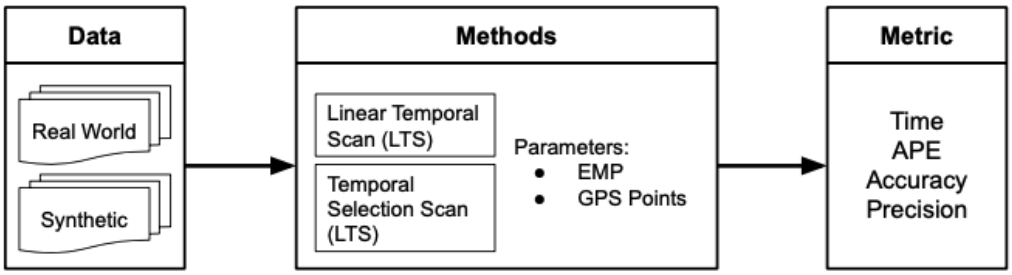


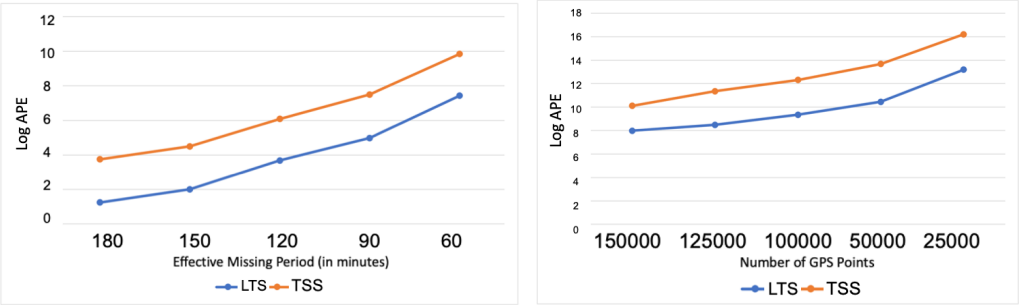
Figure 3.14: Experiment Design

Sensitivity Analysis: The parameters were chosen to answer two questions: For a given study area, we asked:
(1) What is the effect of the Effective Missing Period (EMP) threshold ?
(2) What is the effect of the number of GPS Points ?

Synthetic Dataset: We created a synthetic dataset using a set of real world trajectories from MarineCadastre [1] and considered those trajectory pairs which spatially cross each other. Then we removed overlapping trajectory points to create a hypothetical rendezvous empty space and temporally synchronized the trajectory pairs in order to extract a time interval list as discussed in Section 3.4.3. The experiments were done on over the same number of trajectories with manual labeling for ground truth verification. The manual labelling was based on actual rendezvous for ground truth comparison.

Real World Dataset: We used real world data from the MarineCadastre [1] dataset. The dataset contains many attributes (e.g., Longitude, Latitude, Speed Over Ground (SOG) etc.) for 150,000 objects from 2009 to 2017. The dataset has a geographical extent of 180W to 66W degrees in longitude and 90S to 90N degrees in latitude (based on the WGS 1984 coordinates system). In experiments, we used the attributes Longitude, Latitude, Time, Speed Over Ground (SOG) and Maritime Mobile Service Identity (MMSI). Maximum speed was calculated by taking the average SOG from both endpoints of the EMP.

Computing Resources: All the experiments were conducted using Python and



(a) Decreasing Effective Missing Period (b) Decreasing Number of GPS Points

Figure 3.15: Effect of varying parameters on Area Pruning Effectiveness

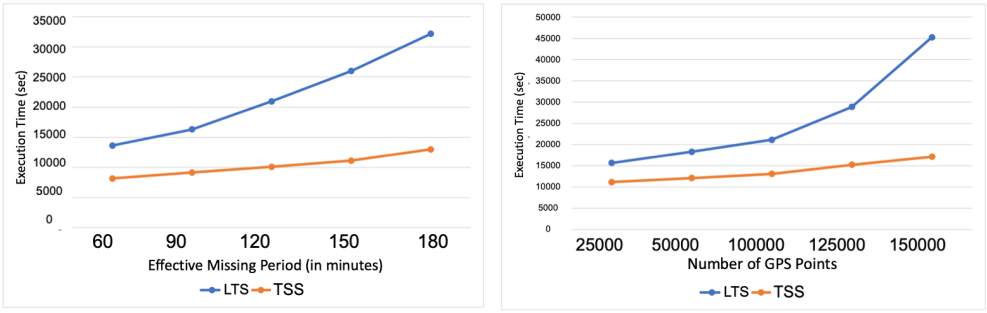
performed on an Intel Core i5 2.5GHz CPU and 16GB memory.

3.7.2 Experiment Results

In this section, we perform a comparative study and sensitivity analysis over performance metrics such as Area Pruning Effectiveness and Computation Time, Accuracy and Precision. In sensitivity analysis, we vary parameters such as Effective Missing Period from 60 minutes to 180 minutes and number of GPS points from 2.5×10^4 to 1.5×10^5 with EMP at 60 minutes.

(1) Area Pruning Effectiveness : Area pruning effectiveness (APE) is the ratio of the total study area and minimum bounded area inside the filtered region. The formulae is presented in Section 3.3.3. This indicates that if APE is higher, the enclosed minimum bounded area will be lower resulting in better solution quality. The main intuition of APE is to reduce the refinement cost by reducing computation time of each cell per unit area [53]. The comparison is based on LTS and TSS using spatial filters 3D R-Trees and MBR respectively where ellipse bounded MBR provides tighter bounding area as compared to 3D R-Trees. The spatial approximations of 3D R-Trees and ellipse bounded MBR is performed at Step 15 of Algorithm 3 and Step 11 of Algorithm 4 respectively. The details of the spatial approximation steps for TSS can be found in Step 4 (Perform Time Slicing) in Section 3.5.3.

Trends: The results of the experiments are shown in Figure 3.15. As can be seen,



(a) Increasing Effective Missing Period (b) Increasing Number of GPS Points

Figure 3.16: Effect of varying parameters on computation time

in all cases, TSS achieves higher area pruning effectiveness than LTS. The results also reveal how APE is affected by different parameters. Figure 3.15 (a) and (b) shows a reverse relationship between APE and the other two parameters, i.e., APE increases with decreasing EMP and number of GPS points. This means the shorter the Effective Missing Period, the less likely an overlap will occur, resulting in a smaller average rendezvous area and higher Area Pruning Effectiveness. Similarly, the larger the number of GPS Points, the greater the number of potential gap pairs, resulting in a larger average rendezvous area and smaller Area Pruning Effectiveness.

(2) Computation Time : Computation Time is the total time taken by the baseline Linear Temporal Scan (LTS) and proposed Temporal Selection Search (TSS) algorithms. Our main purpose is to provide an efficient algorithm in terms of computation time in filter phase. The unit for time measurement is in seconds.

Trends: Using the same set values and parameter increases as the APE experiments, we compared the run times for the baseline LTS and proposed TSS methods. The results are shown in Figure 3.16. As can be seen, increasing the Effective Missing Period (Figure 3.16a), and number of GPS points (Figure 3.16b) leads to longer computation times. This is because longer EMP increases the overlap between candidate pairs, and a greater number of GPS points means more candidate pairs to analyze. Nevertheless, in both cases, the proposed Temporal Selection Search method outperforms the baseline LTS.

We also performed evaluations based on accuracy and precision for ground truth verification. The experiments are based on evaluating correctness of the baseline and proposed algorithm for detecting rendezvous using spatial interaction of their respective spatial approximation filters (i.e. 3D R-Tree and MBR of LTS and TSS respectively). The outcomes for both baseline and proposed algorithm are then compared with ground truth which are based on actual rendezvous. The experiments are performed in synthetic dataset where we manually label the ground truth for actual rendezvous as described in Section 3.7.1. Accuracy and precision can be described as the function of True Positive (TP). True Negative (TN), False Positive (FP), and False Negative (FN). The confusion matrix is shown in Figure 3.17.

		Predicted Class	
Actual Class		**Positive**	**Negative**
	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

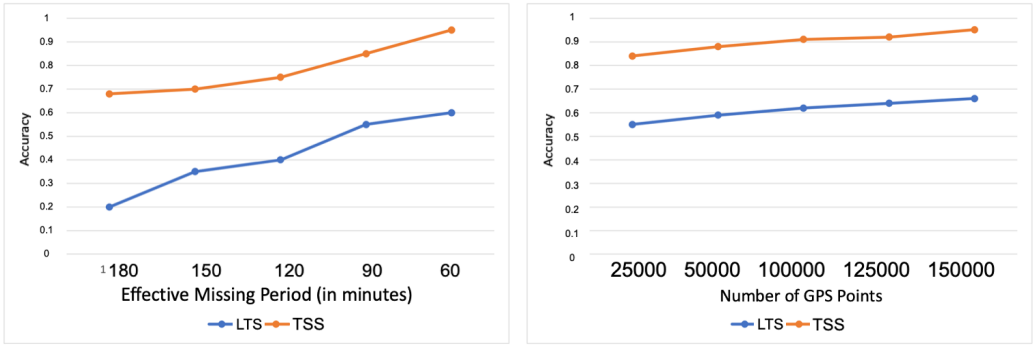
Figure 3.17: Evaluation Confusion Matrix

(3) Accuracy: Accuracy is the ratio of observation which are correctly predicted to the total number of observations i.e. it provides how accurately the algorithms can predict the actual rendezvous which may or may not take place. Based on Figure 3.17, the equation is given below:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.12}$$

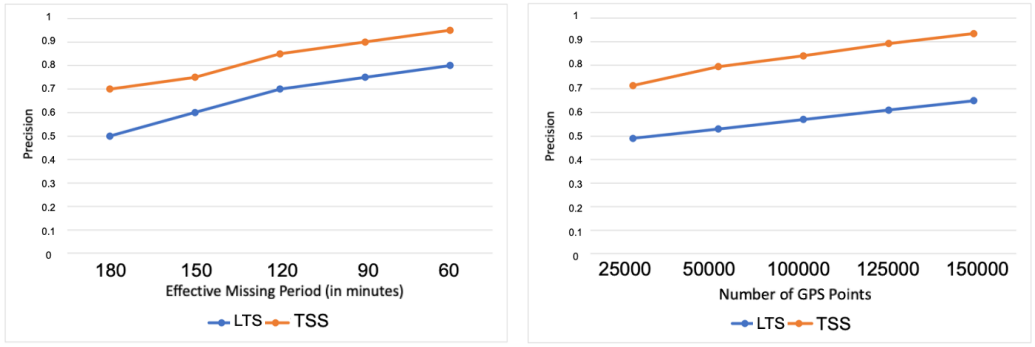
Trends: As, we can see in Figure 3.18a, the proposed TSS approach is more accurate than the baseline LTS approach with substantial improvement as we decrease the Effective Missing Period. This is due to the larger size of the MOBR in LTS as compared to MBR in TSS which results in high potential interaction among EMPs. This results in high number of potential candidate pairs thus resulting in high number of false positive results. Similarly, Figure 3.18b shows the proposed approach outperforms the baseline as we increase the number of GPS Points with fewer number of False Positives.

(4) Precision: Precision is the ratio of observations which are correctly predicted to the total number of predicted positive observations. This includes False Positives instances



(a) Decreasing Effective Missing Period (b) Increasing Number of GPS Points

Figure 3.18: Effect of varying parameters on accuracy



(a) Decreasing Effective Missing Period (b) Increasing Number of GPS Points

Figure 3.19: Effect of varying parameters on precision

(i.e. when algorithm falsely predict the positive rendezvous). Hence, the lower is the number of false positives instances, the higher is the precision. Based on Figure 3.17, the equation is given below:

$$Precision = \frac{TP}{TP + FP} \tag{3.13}$$

Trends: Similar trends are observed using the same parameters in Figure 3.19 showing that the proposed approach outperforms the baseline approach as we decrease the Effective Missing Period (Figure 3.19a) or increase the GPS points (Figure 3.19b). This is due to the fewer number of false positives generated in the proposed approach due to it's tighter minimum bounding rectangle as compared to the 3DR Tree used in the baseline.

3.8 Conclusion and Future Work

Our baseline algorithm is a Linear Temporal Scan [53] based on an approximating common minimum orthogonal bounding rectangle with a time slicing operation for better area pruning effectiveness. We proposed a Temporal Selection Search algorithm that uses a time prioritizer to further reduce the time slicing operation. In addition, we perform spatial optimization that provides tighter bounds than baseline filters. The results show relatively better area pruning effectiveness (APE), computation time, accuracy, and precision for the proposed Temporal Selection Search compared to the baseline approach.

Future Work: Computing approximate regions is expensive in terms of time complexity, and modeling them in regional space is challenging. We are exploring a grid indexing technique that is more efficient than linear search. Also, we plan to apply the proposed techniques in other application domains. We also plan to provide ground-truth verification for evaluation metrics on a real-world dataset. In addition, we plan to provide a detailed theoretical evaluation for the baseline and proposed algorithm. Finally, we plan to further refine the process of finding the exact geometry of a spatiotemporal intersection since it is hard to find the exact geometry of the cone intersection due to its complexity.

Chapter 4

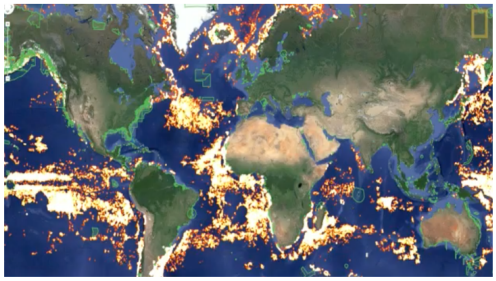
Abnormal Trajectory-Gap Detection: A Physics-based Approach

4.1 Introduction

In the previous two chapters, we focused on identifying abnormal gaps characterized by known spatiotemporal signatures, such as rendezvous events. This task was effectively addressed by performing spatiotemporal joins combined with geometrically informed trajectory gap estimation. However, when analyzing a single trajectory in isolation, determining the abnormality of a gap becomes challenging due to the absence of explicit spatiotemporal signatures for classification. Consequently, this chapter will concentrate on identifying abnormal trajectory gaps that lack predefined spatiotemporal signatures.

Since 2005, ships have been mandated to periodically report their geolocations to ensure collision avoidance. This requirement has generated extensive ship-trajectory datasets (as shown in Figure 4.1b, offering potential reductions in manual labor needed to review remote sensing imagery for monitoring and deterring illicit maritime activities, such as illegal fishing or unauthorized cargo exchanges in expansive oceanic regions. Nonetheless, adversaries can exploit gaps in trajectory data by disabling their automatic identification systems (AIS). For example, as illustrated in Figure 4.1a, a Panamanian commercial fishing

(a) Galapagos Case Study



[Source: How Illegal Fishing Is Being Tracked From Space, Natl. Geographic, 3/12/18*].*

(b) AIS Trajectory Data

Figure 4.1: Mining abnormal patterns without known spatiotemporal signatures

vessel disabled its AIS prior to entering the Galapagos Marine Reserve, only reactivating it 15 days later, likely to obscure illicit fishing activities before returning to Ecuadorian waters. To detect such anomalies, we propose leveraging historical trajectory data as a form of signal coverage map to validate hypotheses of abnormal behavior.

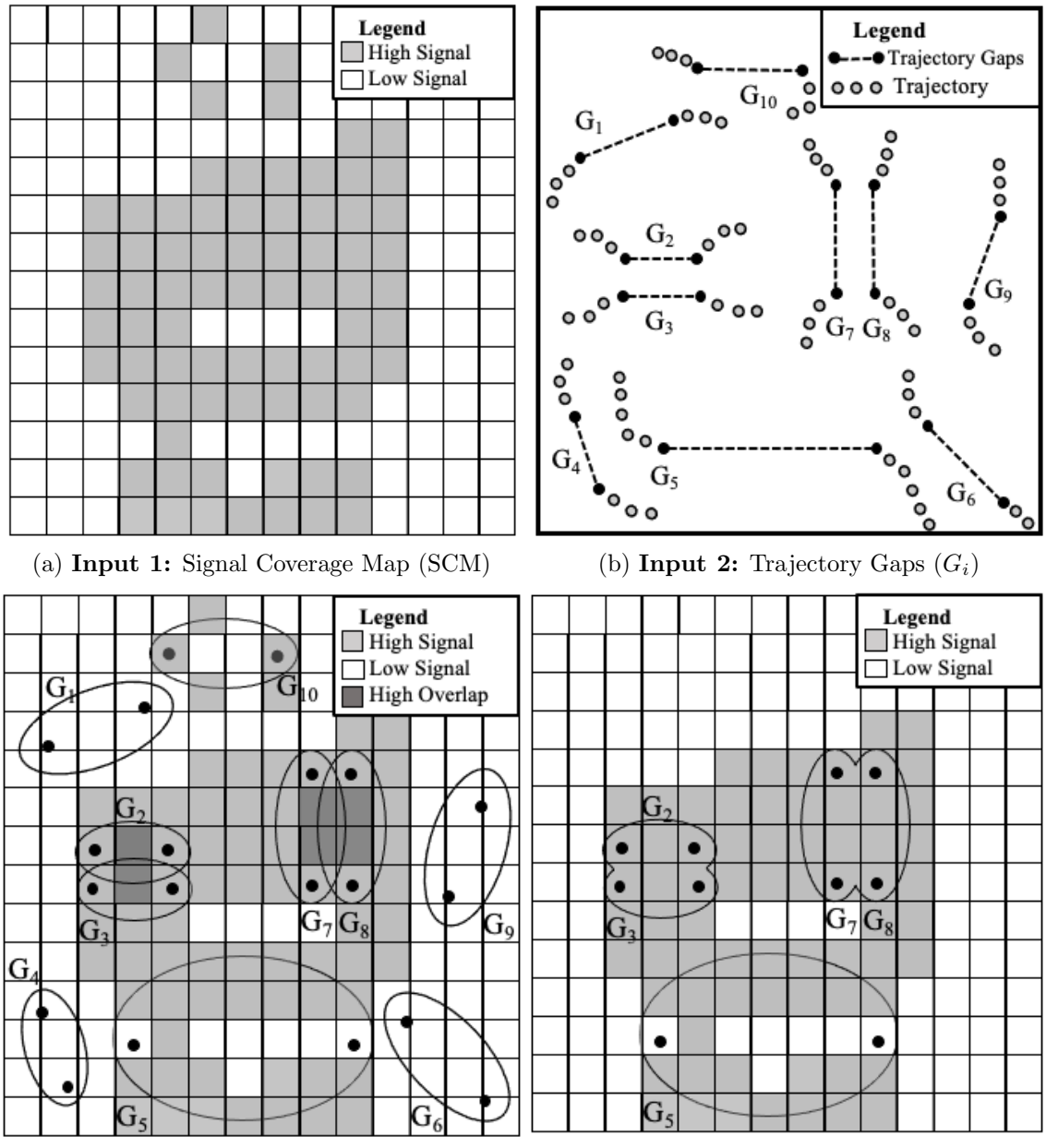
Given multiple trajectory gaps and a signal coverage map based on historic object activity, we find possible abnormal gaps in activity where moving objects (e.g., ships) may have behaved abnormally such as not reporting their locations in an area where other ships historically did report their location. Figure 4.2 shows the problem's *input* which includes a map of the signal coverage area (grey cells) for a set of derived historic trajectories (shown in Figure 4.2a) and trajectory gaps G_1, ... , G_{10} (shown in Figure 4.2b). Figure 4.2 also shows the output where gaps G_1, G_4 G_6 and G_9 are entirely outside the signal coverage map indicating weak signal coverage. In contrast, the rest of the gaps are overlapping the signal coverage map. The absence of location reporting in an area known to have signal coverage may be interpreted as intentional behavior by a ship that temporarily switched off its location broadcasting device. Figure 4.2c shows *intermediate output* stage where trajectory gaps (G_1, ... , G_{10}) are modelled in the form of geo-ellipses [60, 16] along with their intersections with the signal coverage map. Figure 4.2c also shows two-gap pairs, G_2, G_3 and G_7, G_8, that have intersecting regions (i.e., dark grey cells), which suggests two ships in a rendezvous potentially engaging in illegal activity. The gaps entirely outside the signal coverage area have been filtered out. Figure 4.2d shows the *final output* with gap pairs G_2, G_3 and G_7, G_8 are merged to their overlapping regions and gap G_{10} is filtered

out since it did not meet the user defined *priority threshold*.

Analyzing trajectory gaps has many societal applications in maritime safety, homeland security, epidemiology, and public safety. Other use cases include tracing comets, tracking marine animals and contact tracing. In this chapter, we focus on understanding the potential benefits of signal coverage mapping to improve maritime safety and regulatory enforcement. A signal coverage map helps to reduce false positives by providing historical activity traces for a region, against which abnormal behavior can be detected within a trajectory gap. In addition, current methods assume shortest path in a trajectory gap which leads to many missed patterns since moving objects do not always travel in a straight path. In contrast, our approach is based on a space-time prism, which can accommodate greater movement possibilities around a signal gap where an object of interest could have potentially deviated from the predefined (or known) linear path. We also provide a way to reduce computational cost over a large geographical space.

Contributions: We define an Abnormal Gap Measure (AGM) for modeling abnormal gaps and propose an abnormal gap detection (AGD) algorithm to handle multiple gaps. In addition, we propose a memoized abnormal gap detection (Memo-AGD) algorithm to further improve computational efficiency.We show experimentally that our methods are efficient and have good solution quality. We also present a real-world case study to validate our approach.

Scope: This work is limited to space-time prisms for computing abnormal gaps and methods such as kinetic prisms [26] are not studied. This chapter did not consider acceleration in space-time prism due to data limitations. The use of signal coverage maps based on aerial imagery datasets (e.g., satellite imagery) and it's it's falls outside the scope of this chapter. Finally, signal coverage map's stability, its effect based on weather or other environmental conditions, and radio outages are out of the scope of this work.
Organization: The rest of the chapter is organized as follows: Section 4.2 summarizes related work briefly followed by Section 4.3 introduces key concepts and formally defines the abnormal gap detection problem. Section 4.4 describes the proposed algorithms AGD and refined Memo-AGD. Experiments and Results are presented in Section 4.8. Finally, Section 4.9 concludes this work and briefly lists the future work.



(a) **Input 1:** Signal Coverage Map (SCM)

(b) **Input 2:** Trajectory Gaps (G_i)

(c) **Intermediate Output**: Abnormal Gaps

(d) **Final Output:** Summarized Abnormal Gaps

Figure 4.2: An illustration of the Abnormal Gap Region Problem (Best in color).

4.2 Related Work

Surveys [62, 23] provide a broad classification of anomaly detection methods in trajectory mining. Riveiro et al. [32] provide a broad review of maritime anomaly detection but do not cover trajectory gaps. Works that do consider trajectory gaps [63, 64, 33] assume shortest path methods. For instance, the proposed framework in [63] extracts maritime movement patterns assuming shortest path within trajectory gaps.

There are some realistic frameworks [61, 47, 65] that employ reconstruction techniques for modeling uncertainty in trajectory gaps based on space-time prism models [16, 60]. However, they are limited to theoretical simulations and little attention is given to real-world applications [24]. For instance, Winter [65] provides a probabilistic interpretation of a space-time prism. However, the interpretation lacked real-world validation. A recent work [66] considers trajectory gaps using a space-time prism but does not consider abnormal behavior within gaps. This chapter uses a space-time prism model to capture abnormal gaps based on historical data, which allows us to distinguish abnormal gaps as possible anomaly hypotheses.

4.3 Problem Definition

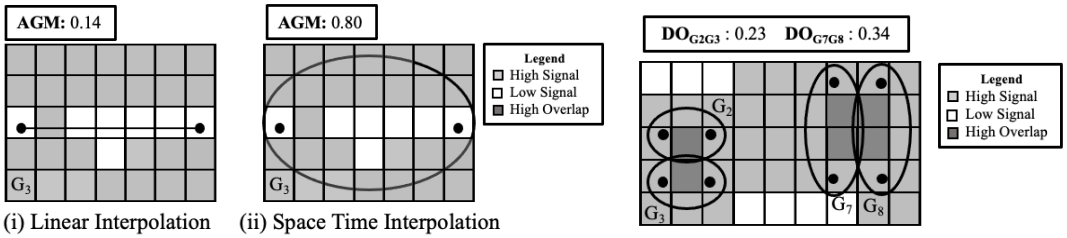
4.3.1 Basic Concepts

Definition 4.3.1. A **signal coverage map (SCM)** is a discretized grid space where cells are color-coded to represent regions with regular historically reported location signals.

The maps are generated by first computing the total reported ship movement in the area using a set of historic location-traces P_i (i.e., trajectories) within some time interval and then checking if the total is above a given threshold (say θ). Figure 4.2a shows a synthetic example of signal coverage map based on historical ship movement with binary color-coding. The grey cells in the figure have movement above the threshold (i.e., High Signal), whereas the white cells have low or no reported historical movement (i.e., Low Signal).

Definition 4.3.2. An **abnormal gap measure (AGM)** for a gap G_i is the probability that a possible location of the object during the gap (unreported data time interval) has signal coverage. A higher value of AGM indicates anomalous behavior since it means an object is not reporting its location despite having the location signal coverage in the past.

The probability is computed using interpolated grid cells (GC_{int}) and regions with high historical movement (GC_m). We first compute the overlap between GC_{int} and GC_m, and then normalize the overlap with GC_{int}. Figure 4.3a provides two examples of computing the abnormal gap measure between two points using cells color-coded based on SCM. The first shows the AGM for linear interpolation, where GC_{int} is the set of cells crossed or touched by a line between the points (i.e., 7) which overlaps with only 1 cell with a high historically reported movement. Thus, the AGM is 0.14 ($\frac{1}{7}$). The second shows the AGM based on (GC_{int}) for space-time interpolation to accommodate additional (movement) possibilities. As shown, GC_{int} is the number of cells crossed or touched by the ellipse (i.e., 35) which overlaps with 28 cells with a high historically reported movement. Thus, the AGM is 0.80 ($\frac{28}{35}$).



(a) Abnormal Gap Measure (b) Degree of Overlap

Figure 4.3: Examples of Abnormal Gap Measure (Left) and Degree of Overlap (Right)

Definition 4.3.3. Given a pair of gaps (G_A, G_B), the **degree of overlap ($DO_{G_AG_B}$)** for two pairs is the minimum ratio between the common interpolated cells to the interpolated cells.

Figure 4.3b shows two gap pairs G_2, G_3 and G_7, G_8. The degree of overlap for the first pair is $\mathrm{DO}_{G2,G3} = 0.23(min[\frac{2}{9}, \frac{2}{9}])$ and for the second pair $\mathrm{DO}_{G7,G8} = 0.33(min[\frac{4}{12}, \frac{4}{12}])$.

4.3.2 Problem Formulation

Input: (1) Trajectory Gaps: Missing location signal(s) between two consecutive points. Fig. 4.2b represent trajectory gaps G_1 to G_{10} which later modelled as geo-ellipses (Fig. 4.2c-4.2d).

(2) Signal Coverage Map (SCM): The primary motivation behind a signal coverage map (Figure 4.2a) is to improve the accuracy of detecting specific trajectory gaps where a

moving object (e.g., ships) may have behaved abnormally. If we consider the entire study area, we may increase the rate of false positives since not reporting locations from moving objects may be due to weak signal coverage in certain regions. Hence, these maps provide a way to narrow down the search space to specific gaps which need further investigation.

(3) Priority Threshold : A threshold value used for extracting abnormal gap based on user's preference. For instance, gaps with AGM scores *above* a priority threshold are extracted and prioritized for further investigation (e.g., G_2, G_3, G_5, G_7, G_8 in Figure 4.2d).

Output: Summarized abnormal trajectory gaps (as shown in Figure 4.2d). Here, we first filter out trajectory gaps that are within areas without signal coverage and then use the priority threshold to prioritize gaps with a high AGM score. In addition, we also coalesce the gaps for two or more geo-ellipse intersections, which usually occur in dense regions, and reduce additional scanning of the overlapped area (e.g., Figure 4.2c) while considering one gap at a time. This results in higher post-processing costs by the human analysts, which can be greatly reduced by merging the common intersection region.

Objective: Our objective is solution quality and computational efficiency. Solution quality can be achieved by reducing the false positive rate and using optimal AGM values to detect abnormal gaps. To enhance computational efficiency, we focus on optimizing gap enumeration during the process of forming clusters of gaps that interact spatially and temporally.

Constraints: The space-time prism do not consider acceleration and deceleration [26].

4.4 Proposed Approach

Framework: Our aim is to identify possible abnormal gaps on a given set of trajectory gaps and signal coverage area through a three-phase *Filter* and *Refine* approach. The trajectory signals are first preprocessed to filter out all the trajectory gaps. Then we model gaps as geo-ellipses and apply the proposed algorithms to effectively optimize the spatial interactions by coalescing pair of gaps and reducing redundant linear scans on the overlap as shown in Figure 4.2c and 4.3b. Then, the output is a summary of significant abnormal gaps (Figure 4.2d) which helps a human analyst for ground truth verification via satellite imagery.

(1) Abnormal Gap Detection (AGD) Algorithm: Here we describe the abnormal gap detection (AGD) algorithm used for enumerating gaps such that each gap is modeled

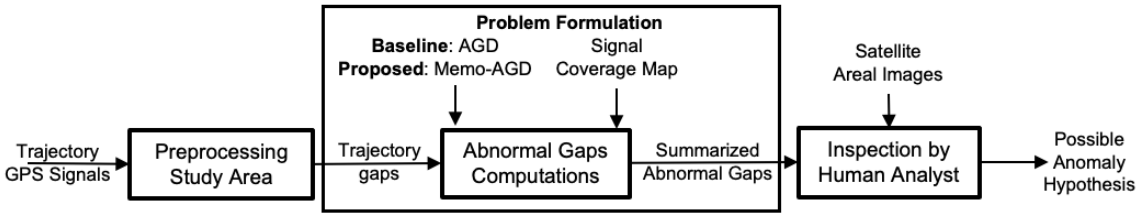


Figure 4.4: Framework for detecting possible abnormal gaps to reduce manual inspection by analyst

via a geo-ellipse and an AGM score is computed with a coalescing operation (Figure 4.2d). First, we sort each gap G_i by time and then check which gap pairs intersect in space and time. We further calculate their Degree of Overlap (DO) and check if DO $\geq \lambda$. Second, we save each qualified gap either as a single (¡G_i¿) or as a subset of gap pairs (e.g., ¡G_i,G_j¿). Finally, we merge all subsets of trajectory gaps and compute their AGM scores, which are later extracted via a priority threshold. An execution trace is given in Section 4.5.

(2) Memoized Abnormal Gap Detection (Memo-AGD) Algorithm: Since the AGD approach enumerates an exponential number of candidates, we use additional variables such as G_i^{Obs} and G_i^{LU}, where G_i^{Obs} keeps track of the total current elements in an Observed List and G_i^{LU} provides a *lookup table* which allow us to store information which was already involved in a prior intersection with G_i. Such *memoization* avoids unnecessary gap enumeration.

Algorithm 5 Memoized Abnormal Gap Detection (Memo-AGD) Algorithm

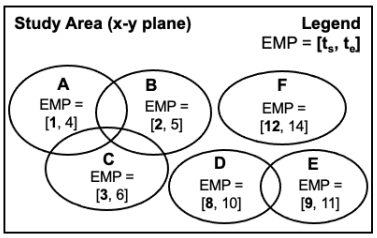
Input: Trajectory Gaps (G_i), Signal Coverage Map (SCM), and DO Threshold λ
Output: Summarized Abnormal Trajectory Gaps

1: **procedure** MEMO-AGD
2: **Step 1:** Initialize Lookup Table G_i^{LU} and copy Observed List to G_i^{Obs}
3: **Step 2:** Check spatiotemporal overlap and avoid gap enumeration
4: **while** $G_i^{Obs} \neq \emptyset$ for any $G_j \in G_i^{Obs}$ **do**
5: **while** $G_i \cap G_j \neq \emptyset$ and DO $\geq \lambda$ **do**
6: Save or update the derived shape from $G_i \cap G_j$ to G_i^{LU} and G_j^{LU}
7: Add $\langle G_i, G_j \rangle$ to G_i^{LU} and G_j^{LU} and update $G_i^{Obs} \leftarrow G_i^{Obs} - G_j^{LU}$
8: **end while**
9: If $G_i \cap G_j = \emptyset$ then update $G_i^{Obs} \leftarrow G_i^{Obs} - G_j^{LU}$
10: **end while**
11: **Step 3:** Compute AGM Score for all elements in Lookup Table G_i^{LU}
12: **end procedure**

First, we initialize G_i^{LU} and copy the current observed list into G_i^{Obs}. After verifying spatiotemporal overlap and ensuring that DO $\geq \lambda$, we update and store the resultant shape derived from $G_i \cap G_j$, and add the subset $\langle G_i, G_j \rangle$ to both G_i^{LU} and G_j^{LU}. For example, a new gap G_k only needs to compare once with the already computed $G_i \cap G_j$ (stored as $\langle G_i, G_j \rangle$ in G_i^{LU} and G_j^{LU}). Additionally, G_k can skip further comparisons with G_j via the operation $G_i^{Obs} - G_j^{LU}$, leading to computational speedup. A detailed trace of Algorithm 5 is provided in Section 4.6.

4.5 Execution Trace of AGD Algorithm

Figure 4.5 illustrates the execution trace of the AGD algorithm using gaps $\langle A, 1 \rangle$, $\langle B, 1 \rangle$, ..., $\langle F, 1 \rangle$, where each entry denotes a gap and its count. Gaps are sorted based on their start time t_s. At $t_s = 1$, $\langle A, 1 \rangle$ is added to the empty observed list. At $t_s = 2$, $\langle B, 1 \rangle$ is added since it intersects with an existing gap, resulting in the sets $\langle A, 2 \rangle$, $\langle B, 2 \rangle$, and $\langle A, B, 2 \rangle$. At $t_s = 3$, a similar expansion occurs, producing the sets $\langle A, 3 \rangle$, $\langle B, 3 \rangle$, $\langle C, 3 \rangle$, $\langle A, B, 3 \rangle$, $\langle B, C, 3 \rangle$, $\langle A, C, 3 \rangle$, and $\langle A, B, C, 3 \rangle$. At $t_s = 8$, $\langle D, 1 \rangle$ is added without interaction. At $t_s = 9$, $\langle E, 1 \rangle$ interacts with $\langle D, 1 \rangle$, producing $\langle D, 2 \rangle$, $\langle E, 2 \rangle$, and $\langle D, E, 2 \rangle$. At $t_s = 12$, $\langle F, 1 \rangle$ is added independently. Finally, we retain maximal sets $\langle A, B, C, 3 \rangle$, $\langle D, E, 2 \rangle$, and $\langle F, 1 \rangle$ for AGM scoring via the signal coverage map and threshold-based prioritization by analysts.



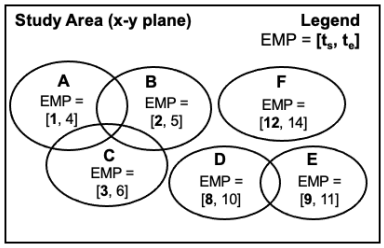
Time	Non-Observed List (G_i)	Observed List (G_j)
t_s = 1	{<B>,1}, {<C>,1}, {<D>,1}, {<E>,1}, {<F>,1}	{<A>,1}
t_s = 2	{<C>,1}, {<D>,1}, {<E>,1}, {<F>,1}	{<A>,2}, {<B>,2}, {<A,B>,2}
t_s = 3	{<D>,1}, {<E>,1}, {<F>,1}	{<A>,3}, {<B>,3}, {<C>,3}, {<A,B>,3} , {<B,C>,3}, {<A,C>,3}, {<A,B,C>,3}
t_s = 8	{<E>,1}, {<F>,1}	{<A>,3}, {<B>,3}, {<C>,3}, {<A,B>,3} , {<B,C>,3}, {<A,C>,3}, {<A,B,C>,3}, {<D>,1}
t_s = 9	{<F>,1}	{<A>,3}, {<B>,3}, {<C>,3}, {<A,B>,3} , {<B,C>,3}, {<A,C>,3}, {<A,B,C>,3}, {<D>,2}, {<E>,2}, {<D,E>,2}
t_s = 12	Empty	{<A>,3}, {<B>,3}, {<C>,3}, {<A,B>,3} , {<B,C>,3}, {<A,C>,3}, **{<A,B,C>,3},** {<D>,2}, {<E>,2}, **{<D,E>,2}, {<F>,1}**

Figure 4.5: Execution trace of the baseline AGD algorithm

4.6 Execution Trace of Memo-AGD Algorithm

Figure 4.6 presents the execution trace of the Memo-AGD algorithm. At $t_s = 1$, Step 1 initializes G_i with variables G_i^{LU} and G_i^{Obs}, which are then added to the empty observed list. At $t_s = 2$, A^{Obs} copies the current observed list (containing only $\langle A \rangle$) to B^{Obs} upon satisfying

spatial and temporal overlap conditions. If satisfied, Step 2 computes the resultant shape $\langle A \cup B \rangle$, storing it in both A and B, and updating A^{LU} and B^{LU} with $\langle A, B \rangle$. Removing $\langle A \rangle$ from B^{Obs} leaves the list empty, terminating the loop. At $t_s = 3$, $\langle C \rangle$ interacts with the shape derived from $\langle A, B \rangle$, resulting in A^{LU}, B^{LU}, and C^{LU} being updated to $\langle A, B, C \rangle$. At $t_s = 8$, $\langle D \rangle$ does not intersect with A, and hence skips intersection checks with B and C using $G^{Obs} - G^{LU}$, and is added directly to the observed list, leading to performance gains over AGD. A similar operation occurs at $t_s = 9$, where $\langle E \rangle$ bypasses A but intersects with D, yielding $D^{LU} = \langle D, E \rangle$. Finally, $\langle F \rangle$ is added directly as a maximal set due to no interactions. Remaining steps mirror those of the baseline, with maximal sets aggregated from each G^{LU} in the lookup table of the observed list.



Time	Non-Observed List (G_i)	Observed List (G_j) with Look Up Table (G_j^{LU})
$t_s = 1$	B^{LU} = {<B>}, C^{LU} = {<B>}, D^{LU} = {<D>}, E^{LU} = {<E>}, F^{LU} = {<F>}	A^{LU} = {<A>}
$t_s = 2$	C^{LU} = {<B>}, D^{LU} = {<D>}, E^{LU} = {<E>}, F^{LU} = {<F>}	A^{LU} = {<A,B>}, B^{LU} = {<A,B>}
$t_s = 3$	D^{LU} = {<D>}, E^{LU} = {<E>}, F^{LU} = {<F>}	A^{LU} = {<A,B,C>}, B^{LU} = {<A,B,C>}, C^{LU} = {<A,B,C>}
$t_s = 8$	E^{LU} = {<E>}, F^{LU} = {<F>}	A^{LU} = {<A,B,C>}, B^{LU} = {<A,B,C>}, C^{LU} = {<A,B,C>}, D^{LU} = {<D>}
$t_s = 9$	F^{LU} = {<F>}	A^{LU} = {<A,B,C>}, B^{LU} = {<A,B,C>}, C^{LU} = {<A,B,C>}, D^{LU} = {<D,E>} , E^{LU} = {<D,E>}
$t_s = 12$	**Empty**	A^{LU} = **{<A,B,C>}**, B^{LU} = **{<A,B,C>}**, C^{LU} = **{<A,B,C>}**, D^{LU} = **{<D,E>}**, F^{LU} = **{<F>}**

Figure 4.6: Execution Trace of Proposed Memo-AGD Algorithm

4.7 Asymptotic Time Complexity Analysis

Given N gaps, both algorithms first perform sorting operations in $O(Nlog(N))$ time and perform k operations for k subsets of coalesced gaps. The time complexity of comparison operations for each algorithms are as follows:

AGD Algorithm: Given k gaps that intersect within all the gaps in the Observed List, it is necessary to check the total number of subsets (i.e., $\binom{k}{1} + \binom{k}{2} + \binom{k}{3} + \ldots + \binom{k}{k} = \sum_{i=1}^{k} \binom{k}{i}$). When a new $(k+1)$-th gap is added, it is necessary to check whether it intersects with all the existing subsets. Since checking with the subsets of size-i gap costs i, the cost of each intersection is $\sum_{i=1}^{k} i \times \binom{k}{i} = k \times 2^{k-1}$. Therefore, assuming there are $|N|$ gaps in the dataset, the total cost for adding is $\sum_{k=1}^{|N|} k \times 2^{k-1} = O(N \times 2^{|N|})$. Since the worst case rarely happens because of the sparsity of geo-ellipse intersections in real datasets, the best case is when all the gaps are disjoint i.e., $O(N^2)$.

Memo-AGD Algorithm: The worst case of Memo-AGD will be similar to *AGD*

except we only check maximal sets instead of $\binom{k}{i}$ gaps. Hence, for each N and k, we update their resultant shape and the elements in the lookup table, resulting in $O(kN)$ and the cost of updating the lookup table. In contrast, the cost will be $O(kN)$ (similar to best case of AGD algorithm) if no gaps are intersecting with each other.

4.8 Experimental Validation

Synthetic Data Generation: For *solution quality* experiment, we lack ground truth data (i.e., absence of information on whether each gap is abnormal or not). Therefore, we evaluated the proposed algorithms on synthetic data derived from a real-world dataset. First, we gather trajectories gathered on a fixed study area ranging from $179.9W$ to $171W$ degrees in longitude and from $50N$ to $58N$ degrees in latitude in the Bering Sea with 1.25 x 10^5 with 1500 trajectory gaps spanning from 2014 to 2016. Then we preprocessed to fetch trajectory gaps with a gap duration greater than 30 mins to qualify it as a trajectory gap. Finally, we calculated an AGM score using linear interpolation and proposed methods and classified each gap as abnormal and non-abnormal based on a specific AGM threshold (i.e., 0.6). For instance, gaps with AGM scores greater than **0.6** are considered abnormal.

Real World Data: We used MarineCadastre [1], a real world dataset containing many attributes (e.g., Longitude, Latitude, Speed Over Ground etc.) for $150,000$ objects from 2009 to 2017. In addition, we also used MarineTraffic [67] for a case-study near the Galapagos Islands to verify the effectiveness of the proposed algorithms.

Computing Resources: We performed our experiments on a system with a 2.6 GHz 6-Core Intel Core i7 processor and 16 GB 2667 MHz DDR4 RAM.

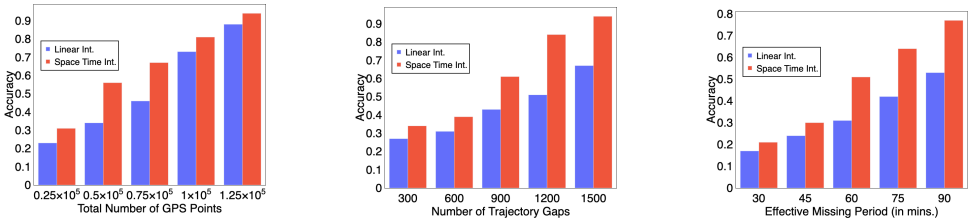
Experiments for Solution Quality: We compared the accuracy of a linear interpolation based method [64, 68] and our space-time interpolation based AGD and Memo-AGD algorithms. Using our synthetic dataset, we varied three parameters: number of GPS points, number of trajectory gaps, and effective missing period (EMP), i.e., the total time when a given object was missing. Figure (4.7a - 4.7c) shows that space-time interpolation outperforms linear interpolation on all three parameters. The reason is that the AGM scores captured are more accurate in space-time interpolation as compared to linear interpolation based methods. First, we fixed the EMP at 60 mins, the number of trajectory gaps at 500, and varied the number of GPS Points from 2.5 $\times$ 10^4 to 1.25 x 10^5.

(1) GPS Points: The results in Figure 4.7a show that space-time interpolation initially

captures abnormal gaps with better accuracy as compared to linear interpolation but slows down as GPS density grows (Fixed Parameters: 500 Trajectory Gaps, EMP: 60 mins). We fixed the number of GPS points to 2.5×10^4, EMP to 60 mins, and varied the number of trajectory gap from 300 to 1500.

(2) Trajectory Gaps: Figure 4.7b shows that space-time interpolation captures more signal coverage for a number of GPS points, resulting in better accuracy as compared to the linear interpolation methods (Fixed Parameters: 2.5×10^4 GPS Points, EMP: 60 mins). Finally, we kept the trajectory gaps to 900, GPS points to 2.5×10^4, and varied the EMP threshold from 30 to 90 mins.

(3) Effective Missing Period (EMP): Figure 4.7c shows that space time interpolation outperforms linear interpolation as large gaps captures more spatial variability resulting in more *accurate* AGM scores (Fixed Parameters: 900 trajectory gaps, 2.5×10^4 GPS Points).

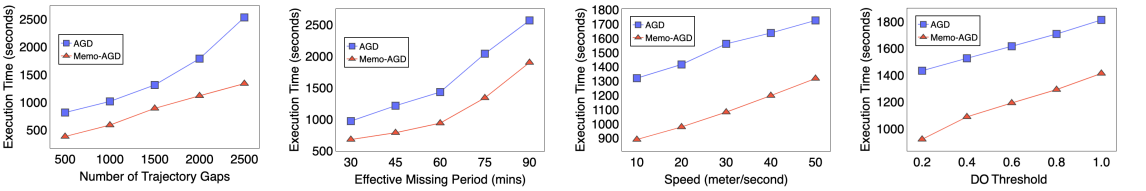


(a) GPS Points (b) Trajectory Gaps (c) EMP

Figure 4.7: Spatio-temporal interpolation is better than linear interpolation for different parameters.

Experiments for Computational Efficiency: Next, we compared Memo-AGD against AGD based on computation efficiency using the MarineCadastre [1] dataset. Figure 4.8a shows that Memo-AGD consistently outperforms AGD. The reason is that the potential interactions of two or more geo-ellipses increase as we increase the number of trajectory gaps. Figure 4.8b shows Memo-AGD outperforms AGD as we increase EMP since larger geo-ellipses result in a higher number of potential interactions. Figure 4.8c also shows MemoAGD is faster than AGD. The reason is that high S_{max} (i.e., maximum possible speed an object can attain during the EMP of its trajectory gap) produces larger geo-ellipses, resulting in more potential interactions. Figure 4.8d shows that higher DO threshold means gaps are less likely to intersect with each other. This helps in avoiding large coalesced gap pairs which result in higher ground-truth verification cost. Hence, MemoAGD is more

efficient than the AGD algorithm.

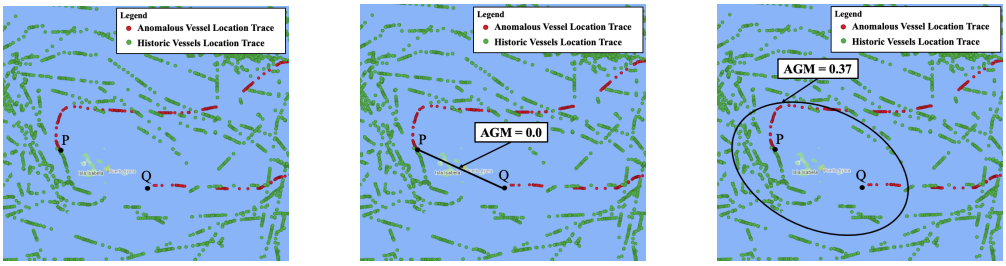


(a) Trajectory gaps (b) Effect of EMP (c) Effect of S_{max} (d) Effect of λ

Figure 4.8: Comparison of execution times of AGD and Memo-AGD under different parameters

Case Study: We conducted a case study of an actual illegal shipping event [13] by applying the proposed Memo-AGD algorithm on a set of real-world data from MarineTraffic [67] data consisting of around 10^5 location trace broadcasted from over 121 fishing vessels for October 2014 (i.e., 31 days) covering the Galapagos marine reserve. Figure 4.9 shows where a fishing vessel (in red) inside a protected habitat area reported no signals for 15 days because it had switched off its transponder. We applied our Memo-AGD algorithm and a linear interpolation method on the historic location traces (in green). In linear interpolation, the vessels interacted small number of island and did not intersected with any historic location trace (Figure 4.9b) resulted in an AGM score of **0.0**, indicating no abnormal activity detected. By contrast, the Memo-AGD algorithm returned a geo-elliptical area (Figure 4.9c) within which some historic trajectories resulted in an AGM score of **0.37**, which accords with the known abnormal activity. The domain experts can set a priority threshold of 0.3 in the flagged area for further investigation by the human analysts.

Discussion: In this chapter, we used a signal coverage map (SCM) as input to significantly reduce the false-positive rate by filtering out gaps in zones with historically weak signal coverage. However, estimating signal coverage is challenging if the vessel enters certain regions (e.g., the arctic) where limited or no historical trace data exists. This result in cold start problem [66]. In addition, the proposed methods results in similar execution times in case of no potential interactions among geo-ellipses (more details described in Section 4.7). In addition, none of the local gaps (e.g., G_i to G_n in Figure 4.2) exceed the time range of the signal coverage map. For instance, the time range of the largest signal



(a) No Interpolation (b) Linear Interpolation (c) Memo-AGD

Figure 4.9: Comparison of Linear Interpolation and Memo-AGD around Galapagos Marine Reserve

gap in Figure 4.9 is 15 days which does not exceed the signal coverage which is spanned for an entire month (i.e., 31 days). Finally, this chapter considered only the object's average speed at the start and endpoint while modeling trajectory gaps that may produce smaller ellipse regions due to low speed. For instance, the vessel's speed at P and Q in Figure 4.9 was 0.16 and 0.32, resulting in a relatively smaller ellipse, given the object disappeared for 15 days. Hence, considering aggregate metrics around the start and endpoint may give a more accurate geo-ellipse representation which is to be considered for future work.

4.9 Conclusion and Future Work

We investigated Abnormal Gap Measure and proposed AGD and Memo-AGD algorithms. We performed experimental evaluation under varying parameters where the results show that compared to linear interpolation methods, space-time interpolation based methods [60] are better at detecting abnormal gaps in trajectory data. Also, Memo-AGD is computationally more efficient than AGD. In the future, we plan to develop spatiotemporal statistics and conduct a statistical significance test to eliminate chance patterns. We will add other use cases such as a cold-start problem [66] and include acceleration using kinetic prisms [26]. Finally, we plan to explore datasets in network space for other potential societal use-cases.

Chapter 5

A Physics-based Abnormal Trajectory Gap Detection

5.1 Introduction

In the previous chapter, we introduced the problem of detecting abnormal trajectory gaps without known spatiotemporal signatures. In this chapter, we extend this formulation by proposing computationally efficient algorithms. We evaluate these methods against models that interpolate gaps using linear and most frequent path approaches. Experimental results demonstrate that our proposed anomaly detection method outperforms both linear and most frequent path interpolation techniques.

Most of the studies [63, 64, 33] on trajectory gaps that bound the location of the moving object is based on shortest path discovery. For instance, [33] derives knowledge of maritime traffic to detect low-likelihood behaviors and predict a vessel's future positions using linear interpolation. Other linear interpolation methods are based on K Nearest Neighbor [69] reputation where each point is predicted based on Constant Velocity Model (CVM) followed by nearest neighbor computation. Some studies [16, 44] do bound an object's possible location by providing some deterministic methods (e.g., space-time prism model) and probabilistic methods [65], but none of the approaches address abnormality within trajectory gaps.

Our previous work defined an abnormal trajectory gap measure and provided a scalable

algorithm Memo-AGD [70] that bounds the entire range of an object's possible movements during the trajectory gap to derive possible anomaly hypotheses for human analysts. However, the previous approach was ineffective in handling spatial comparisons and computing abnormality scores. Here we extend this work by leveraging space-time indexing techniques to further reduce the number of required comparison operations. We also optimize merging operations to promote a large number of merged groups (Figure 4.2c) and lower abnormal gap measure computations while preserving correctness and completeness.

Contributions :

- We propose a Space-Time Aware Gap Detection (STAGD) with Dynamic Region Merge (DRM) approach where STAGD efficiently performs space-time indexing and merging of trajectory gaps, and DRM further improves the computations of gap abnormality scores.
- We evaluate the proposed algorithms theoretically for correctness and completeness and also analyze the time complexity of the baseline and proposed algorithms.
- We provide experimental evaluations of the proposed algorithms using the evaluation metrics such as computation time and accuracy under varying parameters where the proposed STAGD+DRM proves substantially faster than Memo-AGD.

Scope: In this work, we leverage the space-time prism idea for computing an abnormal gap measure and designing the proposed algorithms. For modeling gaps, we did not include acceleration, as this data was not available in the dataset. A detailed interpretation of acceleration using kinetic prism is discussed in Appendix (reference removed). We are not considering low-density or sparse regions for detecting abnormal gaps. Other factors, such as device malfunction, hardware failure, and signal strength due to external factors (e.g., severe weather conditions), fall outside the scope of this work. In addition, we do not model anomalous trajectories without gaps. The proposed framework has multiple phases (i.e., filter, refinement, and calibration), but we limit this work to the filter phase. The refinement phase requires input from a human analyst and is not addressed here. Calibration of the cost model parameters also falls outside the scope of this work.

Organization: The chapter is organized as follows: Section 5.2 introduces basic concepts, the general framework, and the problem statement. A study of related work

is provided in Section 5.3. Section 5.4 provides an overview of the baseline approach AGD and Memo-AGD [70]. Section 5.5 describes the proposed Space-Time Aware Gap Detection (STAGD) and Dynamic Region Merge (DRM) algorithms, respectively, along with their execution traces. Theoretical analysis of both algorithms is provided in Section 5.6. Experiment design and results are reported in Section 5.7. Section 5.8 discusses some of the discussions of our study. Finally, Section 5.9 provides the conclusion and future work.

5.2 Abnormal Trajectory Gap Detection Problem

In this section, first, we define key concepts and describe our general framework. Then, we formally define the problem, followed by some brief remarks about other ways the problem could be formulated. Finally, we present our problem formulation.

5.2.1 Basic Concept

Definition 5.2.1. A **study area** is a two-dimensional rectangular area in which the geo-locations of the input data are represented with latitude and longitude coordinate systems.

Definition 5.2.2. A **spatial trajectory** is a trace of chronologically sorted GPS points in a series generated by a moving object in a geographic space.

For example, $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$ represents a spatial trajectory, $p_i = (x_i, y_i, t_i)$ where each point (p_i) is associated with a geographic coordinate set (x_i, y_i) and a time stamp (t_i). Figure 1 shows the chronological sequence of GPS points (in grey), each associated with gap G_i.

Definition 5.2.3. An **object maximum speed (S_{max})** is the maximum speed an object can attain based on the domain knowledge. The variable S_{max} can be identified from publicly available maritime vessel databases [1]. For vehicles, humans, or animals, we can use the maximum physically allowed speed.

Definition 5.2.4. An **effective missing period (EMP)** is a time period when a GPS signal is missing for longer than a certain threshold (e.g., 30 mins) which is externally specified by the end-user. Figure 4.2 shows that the EMPs for gaps G_1,..,G_n are between the foci of the ellipses with greater than a certain interval (e.g., 30 mins, etc).

Definition 5.2.5. A **signal coverage map (SCM)** is defined as a discretized grid space where each grid cell GC_i represents the historically reported location traces p_i generated by a set of trajectories.

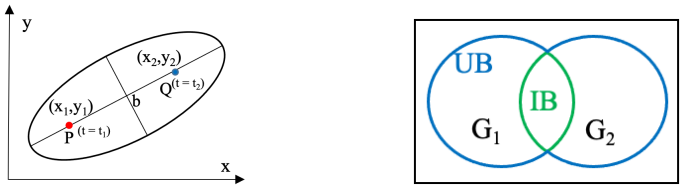
The process initiates by dividing the study area into a grid structure for a specific geographic region, ensuring global coverage of location traces. Initially, we calculate the maximum and minimum latitudes and longitudes to establish the minimum orthogonal bounding rectangle (MOBR). Subsequently, we evenly subdivide the MOBR using both latitude and longitude to construct grid cells and each grid cell is populated by the historic location-traces for a **fixed** time-frame. We finally classify each cell as reported (grey) or not reported (white) based on certain threshold θ.

The signal coverage map classifies each grid cell denoted as GC_i (e.g., GC_1, GC_2,..., GC_n) as reported or not. A reported cell (GC_m) is a cell whose total number of location traces is greater than a certain threshold θ (i.e., $\sum_{i=1}^{n} P_i \geq \theta$). In Figure 4.2, the grid cells in *grey* are reported cells or cells which are frequently reported by end-user, whereas the not reported cells are cells, where reported ship traffic is low or signal strength is under capacity (i.e., low density or sparse regions), are shown in *white*. Hence, the signal coverage map is a boolean representation of discretized grid $\forall\ GC_i \in [0, 1]$. The accuracy of the proposed algorithm significantly depends on the selection of the spatial region and the time frame (more in Section 5.8).

Definition 5.2.6. A **trajectory gap (G_i)** is defined as a space-time interpolated region within a missing location signal time period between two consecutive points.

In Figure 4.2, G_1,G_2,G_3...G_{10} represent trajectory gaps that have been estimated via space-time interpolation in the form of geo-ellipses based on the properties of a space-time prism model [16]. A space-time prism can be projected onto an $x - y$ plane [60, 16] in the form of a geo-ellipse which spatially delimits the extent of a moving object's mobility given a maximum speed during a missing signal period. Figure 5.1a shows a geo-ellipse representation with start point P at (x_1,y_1) at time t_1 and end point Q at (x_2,y_2) at time t_2, where $t_1 < t_2$ with focii (x_1, y_1) and (x_2, y_2).

The minimum orthogonal bounding rectangle (MOBR) is constructed over this ellipse, particularly focusing on the endpoints of its major axis by calculating the angle of this axis



(a) Projected Geo-Ellipse from Space-Time Prism

(b) IB (green) and UB (blue) of gaps G_1 and G_2.

Figure 5.1: Example of (a) a geo-ellipse and (b) an intersection (IB) and union boundary (UB) (Best in Color)

relative to a fixed coordinate system. This step is followed by rotating the ellipse to align this axis with one of the coordinate axes and then identifying the maximum and minimum extents of the ellipse in the direction perpendicular to the major axis. These extents, combined with the major axis endpoints, establish the MOBR's boundaries. Employing geometric transformations and the inherent properties of an ellipse, this approach is an effective means of determining the MOBR.

The R*-tree hierarchically organizes MOBRs within bounding boxes that cover child nodes allowing addition or removal of MOBRs without needing to restructure the entire tree. Through insertion, controlled splitting, and, when necessary, reinsertion, the tree maintains balance and enhances query performance. Incremental indexing of MOBRs in the R*-Tree, guided by a meticulous selection and splitting process, ensures efficient storage, access, and querying of spatial data, capitalizing on the R*-Tree's architectural advantages.

The temporal dimension, specifically the start and end points of a given period, is indexed separately within a large array, where each array index corresponds directly to a unique timestamp. This method ensures that every position in the array represents a distinct moment in time, allowing for efficient mapping and retrieval of temporal data. By aligning each index with a specific timestamp, this approach facilitates quick access to the associated start and end points, optimizing the process of querying temporal intervals within the dataset.

Definition 5.2.7. An **intersection and union boundary (IB and UB)** is the outline derived from a geo-ellipse region participating in intersection and/or union operations when two or more gaps have a spatial and temporal overlap.

For instance, in Figure 5.1b, two polygons, G_1 and G_2 undergo intersection (green outline) and union operations (blue outline). G_1^{IB} or G_2^{IB} is represented as Polygon($G_1 \cap G_2$) whereas G_1^{UB} and $G_2^{UB}$is represented as Polygon($G_1 \cup G_2$).

Definition 5.2.8. An **abnormal gap measure (AGM)** for a gap G_i is the probability that a possible location of an object during a gap (unreported data time interval) has signal coverage. A higher value AGM indicates anomalous behavior since it means an object is not reporting its location despite having signal coverage.

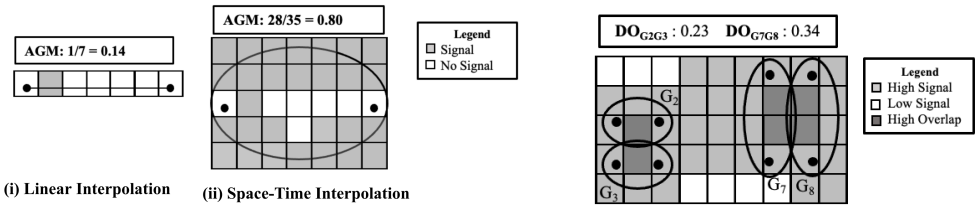
GC_{int} denotes as the interpolated grid cells that reside within a spatial boundary (in the case of space time interpolation) or that intersect a straight line (in the case of linear interpolation). The AGM computes the ratio of interpolated grid cells GC_{int} intersecting with reported calls GC_m (where $GC_m \subseteq GC_{int}$) to GC_{int} . The formula is as follows:

$$AGM = \frac{GC_m \cap GC_{int}}{GC_{int}} \tag{5.1}$$

Figure 5.2a (i) and (ii) show an example of computing AGM scores for linear and space-time interpolation, respectively. The linear interpolation captures only one GC_m within its interpolated space GC_{int} (i.e., a straight line). As a result, only 1 grey cell within 7 white cells is intersected by the straight line (i.e., $\frac{1}{7}$ or 0.14). By contrast, the space-time interpolation is able to capture 28 grey cells which intersect or are within its spatial boundary, resulting in $\frac{28}{32}$ or 0.80, and a better estimate than the linear interpolation.

In this study, we confined our analysis to a static time frame spanning two years (2014-2016) as the computation of the abnormal gap measure is subject to significant variations over time. For example, employing a longer temporal range with a constant threshold, such as from 2009 to 2019, results in every cell being categorized as grey, in contrast to a more limited time frame, like from 2014 to 2016. This demonstrates that selecting an optimal threshold value is necessary for achieving a meaningful distribution of grey and white cells in the signal coverage map, which in turn facilitates the derivation of a significant AGM score within a trajectory gap.

Definition 5.2.9. A **degree overlap (DO)** is a measure to calculate the minimum of the ratio of interpolated grid cells G_i that intersect with a signal coverage map within an intersection boundary (IB) of G_i and G_j. It provides a measure to calculate the degree of participation for each gap G_i within a gap set (G_1,G_2), (G_3,G_4), and so on.



(a) Abnormal gap measure (b) Degree overlap threshold

Figure 5.2: Examples of abnormal gap measure (Left) and degree overlap threshold (Right)

$$DO = min[\frac{(GC_m^{G_i} \cap GC_{int}^{G_i}) \cap (GC_m^{G_j} \cap GC_{int}^{G_j})}{GC_{int}^{G_i}}, \frac{(GC_m^{G_i} \cap GC_{int}^{G_i}) \cap (GC_m^{G_j} \cap GC_{int}^{G_j})}{GC_{int}^{G_j}}] \tag{5.2}$$

In this chapter, we denoted the DO threshold by λ. The motivation behind the threshold is to provide early-stage filtering of a set of smaller coalesced gaps (e.g., 30 mins) interacting with relatively larger gaps (e.g., 3 hrs). This avoids the need for large coalescing operations, which result in more effective pruning of grid cells while calculating AGM scores with the signal coverage map. In addition, the length of the gap is independent of pixel sizes but directly proportional to the number of pixels (e.g., longer gaps tend to have larger spatial coverage, resulting in a large number of pixels in geo-ellipses. In addition, a higher lambda value provides a stricter condition, resulting in fewer merged groups (more details in Sec. 5.2.3). Figure 5.2b shows two gap pairs G_2, G_3 and G_7, G_8 intersecting with some degree of overlap. According to Equation 5.2, the degree of overlap DO$_{G2,G3}$ and DO$_{G7,G8}$ is 0.23 (i.e., min[$\frac{2}{9}$, $\frac{2}{9}$]) and 0.34 (i.e., min[$\frac{4}{12}$, $\frac{4}{12}$]) respectively, which means each gap participates equally in each pair.

5.2.2 Framework

Our aim is to identify possible abnormal regions for a given set of trajectory gaps and signal coverage area through a three-phase *Filter* and *Refine* approach. This chapter introduces an intermediate filter phase to compute abnormal gaps and reduce the number of overlapping grid cells when there are pairs of trajectory gaps (Figure 4.2c). Merging gaps will reduce the number of a high number of overlap grid cells (dark grey in Figure 4.2c) and allow us to compute AGM scores for a larger number of merged groups (i.e., *maximal group*) rather than individual trajectory gaps. This reduces the redundant work of considering

high overlap region associated with each gap and further computing AGM scores, which is a time-intensive operation.

The trajectory signals are first preprocessed to filter out trajectory gaps associated with multi-attributes (e.g., speed) to perform space-time interpolation. Then we apply the baseline Memo-AGD and proposed STAGD+DRM algorithms to optimize the number of spatial interactions by merging gaps, avoiding the need for redundant computations and improving computational efficiency. The output is a summary of significant abnormal gaps (Figure 4.2d) which helps a human analyst to scan a comparatively minimal area for inspection. The results can be further verified via satellite imagery to derive a possible hypothesis about the anomaly (Figure 5.3).

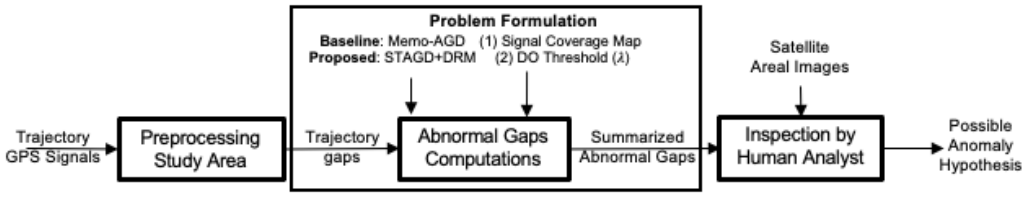


Figure 5.3: Framework for detecting possible abnormal gaps to reduce manual inspection by analyst

5.2.3 Problem Formulation

We formulated the problem to optimally identify an abnormal trajectory gap region in a spatiotemporal domain is formulated as follows:

Input:

1. A study area S,
2. A set of $|N|$ trajectory gaps
3. A signal coverage map SCP
4. A priority and a degree overlap threshold λ

Output: Summarized abnormal trajectory gaps.

Objective: Solution Quality and Computational Efficiency

Constraints:

(1) Grid layout is not available.

(2) Acceleration is not available.

(3) Correctness and Completeness.

Figure 4.2 (a) and (b) show the input as a two-dimensional representation of a signal coverage area and a set of trajectory gaps. Figure 4.2 (c) and (d) show the significant summarized abnormal output after execution of the proposed algorithms.

DO Threshold (λ) Interpretation: The Degree Overlap (DO) threshold can be adjusted externally by a human analyst to yield more significant merged gap regions as shown in Figure 5.3, where λ serves as an input parameter for the proposed algorithm. Specifically, increasing the value of λ imposes stricter criteria for merging gaps, allowing human analysts to inspect individual gaps with greater precision. The concept of Degree Overlap is founded on the principle of the minimum participation index between pairs of trajectory gaps or merged groups (collections of trajectory gaps). This metric provides insight into the potential proximity of two entities to engage in unauthorized activities in the maritime domain (e.g., illegal oil transfers) despite adequate signal coverage. By adjusting the Degree Overlap threshold, analysts are afforded greater flexibility in scrutinizing each gap within merged groups. This includes the identification of significantly anomalous gaps within groups characterized by high anomaly scores. For example, a higher DO threshold λ leads to a larger number of distinct abnormal gaps for verification by human analysts using satellite imagery, compared to lower threshold values. Nonetheless, verifying gaps through satellite imagery necessitates extensive post-processing work. Therefore, finding an optimal DO threshold is necessary for balancing solution quality and computational efficiency.

5.2.4 The Choice of Problem Formulation

The proposed abnormal gap measure (AGM) is based on identifying all possible occurrences of the vessels not reporting their location even though other moving vessels in the area did periodically. We *chose* to formulate the abnormal gap detection problem based on mandatory global maritime safety guidelines where ships must periodically report their geo-location to prevent potential collisions and illegal activities. The absence of location traces despite strong signal coverage makes it harder for authorities to disregard them and they are later sent to human analysts (Figure 5.3).

However, the abnormality in ship behavior can be defined in other ways as well, such as

monitoring the aggregated gaps based on the time of day or a ship's physical and voyage-specific attributes. For instance, an increase in the total number of trajectory gaps at a specific time of day may alarm maritime authorities for specific vessel types (e.g., increased fishing efforts in night time is more suspicious as compared to day-time). Abnormality can also be defined with other physical attributes (e.g., headings, draft). For instance, draft, the depth of water that a ship needs in order to float, is inversely proportional to speed since loading of cargo reduces a ship's speed but raises its draft and vice-versa. Another possibility is to study the distribution of a gap population to define more accurate probabilistic models such as Gaussian Processes to quantify potential locations within geo-ellipses further [16]. However, each choice of problem formulation still lacks physical interoperability and requires ground-truth information, which later requires domain interpretation. More details are discussed in Section 5.8.

5.3 Related Work

The extensive amount of trajectory data being generated via location-acquisition services today has spurred new research in anomaly detection, pattern recognition, etc. The survey in [23] gives a broad overview of trajectory data mining techniques related to pre-processing, data management, uncertainty, pattern mining, etc. Methods for detecting anomalous trajectories are included in a broad classification of anomaly detection methods [62], which also includes trajectory anomaly detection. The literature [24] also provides a taxonomy of movement patterns, classifying them as primitive, compound, individual, group, etc. There are several methods for determining the similarity of trajectories, including edit distance [54], longest common sub-sequence (LCSS) [55], and dynamic time warping (DTW) [56]. A recent technique called TS-Join [49] uses both spatial and temporal factors in a two-step process to determine similarity. Another method, Strain-Join [57], uses a signature-based approach to identify similar trajectories by creating unique signatures and eliminating dissimilar ones. Other techniques include using a sliding window approach [50, 58] to identify potential candidates and filtering out non-similar trajectories using time interval thresholds [51, 52]. For trajectories with silent periods, Clue-aware Trajectory Similarity [59] uses clustering and aggregation based on movement behavior patterns. In terms of processing queries for spatiotemporal joins, methods based on 3D R-Trees, such as Spatiotemporal R-Tree and TB-Tree [42], are used. However, none of the literature has addressed anomaly

detection within trajectory gaps. In the maritime domain, Riveiro et al. [32] provide a generalized view of maritime anomaly detection but do not cover trajectory gaps.

Uncertainty within trajectory gaps can be quantified based on behavioral patterns of moving objects, where many techniques perform linear interpolation assuming shortest path discovery. There are many realistic frameworks [63, 64, 33] that do quantify uncertainty within trajectory gaps, but they are loosely based on shortest path discovery. For instance, one study [63] derives knowledge of maritime traffic in an unsupervised way to predict a vessel's future positions. However, the movement behavior assumes linear interpolation and relies on historical traces, which require extensive training data.

Besides the shortest path, the chapter [71] uses k-Nearest Neighbors (k-NN) imputation to estimate missing Received Signal Strength Indicator (RSSI) values in AIS vessel trajectory data. When a dropout (missing AIS message) is detected, the vessel's position is first predicted using a tracking system like the Constant Velocity Model (CVM). Then, the k-NN method finds the k AIS messages nearest in position to the predicted vessel position, and the RSSI value of the single nearest AIS message (k=1) is used as the imputed RSSI for the missing message, using the spatial correlation in the trajectory data. A variant of KNN [72] was also studied, which is a resampling method used to estimate the variance of a statistic, such as a sample mean. It is useful when the theoretical variance is difficult to derive or when the data has been subject to complex procedures like imputation. Bayesian Networks [73] were used, where multiple Bayesian networks are constructed, one for each attribute with missing values. However, all methods either consider the shortest path or the most frequented path, which does not fully account for object movement capabilities.

The space-time prism model [16, 60] is a more realistic estimation of trajectory gap because it identifies a larger spatial region surrounding a gap where an object could have potentially deviated from the predefined linear path. A number of applications based on joins [60], summarization [74], rendezvous patterns [53, 75, 76] and probabilistic interpretation [65] have been proposed, but they are limited to theoretical simulations and need real-world validation. Other probabilistic frameworks [61, 47] incorporate space-time prisms but do not address the abnormal mobility behavior of an object. A recent work [66] considers trajectory gaps but only focuses on the space-time prism and does not consider abnormal behavior within gaps. This chapter uses a space-time prism model to capture abnormal

gaps based on historical data for verifying possible anomaly hypotheses.

5.4 Baseline Approach

Our previous work [70] was based on a plane sweep approach for effective pruning of pairs of gaps in the filter phase and improving gap enumeration. The baseline method lays the groundwork for our proposed new work that will be explained later in Section 5.5. In this section, we provide a brief overview of the operations used in the baseline approach:

Plane Sweep Approach: We used a plane-sweep approach which is a filter and refine technique [76] where the given study area is projected on a low-dimensional space (i.e., from 2D ellipses to 1D time-segments). In the filtering phase, all gaps are sorted based on x,y, or time coordinates to reduce storage and I/O cost. This allows the computation of intersections in a single pass. In the refinement phase, the AGD and Memo-AGD algorithm [70] extracts gaps based on their spatiotemporal overlap. We used time as an initial filter phase to filter out gaps that are temporally synchronous. We perform a linear scan and check t_{start} and t_{end} for both gap pairs and check if t_{start} and t_{end} of the previous gap overlap with the t_{start} and t_{end} of the current gap.

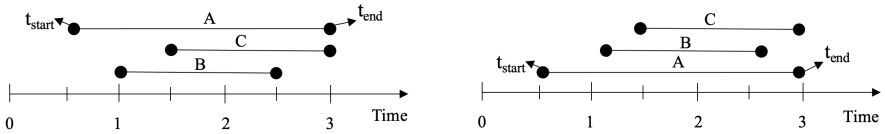
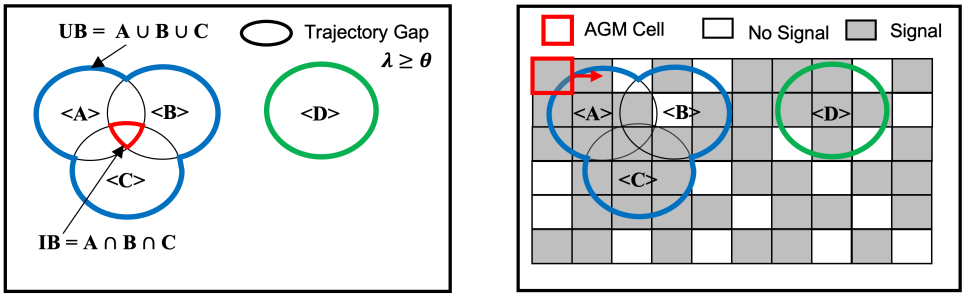


Figure 5.4: Plane Sweep Approach (Filter Step)

Figure 5.4 shows the initial filtering where we check t_{start} and t_{end} for gaps A, B, and C. Initial sorting prior to performing gap enumerations helps in considering linear scan via comparing t_{start} and t_{end}. In the refinement step, imagine a plane sweeping through the start-time and end-time of each gap and check whether the start time of B is smaller than the end time of A. If it is, then we consider A and B as potential candidates for the geo-ellipse spatial intersection.

Trajectory Gap Enumeration: After performing the plane sweep, the abnormal gap detection enumerates gaps by modeling every gap as geo-ellipse and applies an operation that checks which gaps intersect spatially, filtering out large gaps that don't meet the DO threshold λ. However, the operation is exorbitantly expensive because it considers all possible combinations of gap pairs. For instance, in Figure 5.5a, since gaps A and B are

intersected, gap C also needs to satisfy the spatial intersection of A and B in order to be a part of the maximal group A, B and C. Similarly, D also compared with A, B, and C resulting in an exponential candidates for human analysts to post-process satellite imagery.



(a) Memo-AGD Gap Enumeration (b) AGM computations

Figure 5.5: Examples of (a) Memo-AGD Gap Enumeration and (b) Memo-AGD AGM computations

Memoized abnormal gap computation (Memo AGD) uses additional variables, namely, G_i^{Obs} and G_i^{LU}, where G_i^{Obs} keeps track of the total current elements in an Observed List and G_i^{LU} provides a *lookup table* which allow us to store information which was already involved in a prior intersection with G_i. *Memoization* reduces the need to compute gaps unnecessarily. For instance, Polygon (A ∩ B ∩ C) in Figure 5.5a can be cached so that computing gap D does not require redundant computations of A, B, and, C resulting in an only comparison of Polygon $(A \cap B \cap C)$.

Abnormal Gaps Measure (AGM) Computations: After the gap enumeration, we merge all maximal subsets to reduce the total number of candidate pairs for reducing the redundant computation of cells in polygon operations in overlapping regions. The aim is to reduce the number of gap candidates while performing AGM computations which is typically a cell in polygon operation. For instance, Figure 5.5b shows the gap $\langle A \rangle$, $\langle B \rangle$, and $\langle C \rangle$ are merged into Polygon $(A \cup B \cup C)$, since $A \cap B \cap C \neq 0$. The merged regions then undergo *cell-in* polygon operations to compute the abnormal gap measure (AGM) for Polygon $(A \cup B \cup C)$. For instance, in Figure 5.5b, AGM cells linearly perform a cell in polygon operation with Polygon $(A \cap B \cap C)$ and Polygon(D) to compute AGM scores of Polygon $(A \cap B \cap C)$ as 0.55 and Polygon(D) as 0.66.

5.5 Proposed Approach

Our baseline AGD and MemoAGD method performed certain computations (e.g., sorting, comparison), which can be further optimized by introducing a less strict merging condition to reduce redundant computations in real time. In addition, computing the AGM score is computationally expensive since it heavily depends on geographical grid computations and other factors (e.g., DO threshold) described in Section 5.4. Here, we introduce space-time aware gap detection (STAGD) to reduce redundant comparison operations and Dynamic Region Merge (DRM) to improve the efficiency of cells in polygon operation. In this section, we describe the proposed algorithm (STAGD+DRM) for enumerating gaps where each gap intersects in space-time and forms a merged cluster that is later merged dynamically to perform efficient AGM computations.

5.5.1 Spatial-Time Aware Gap Detection (STAGD)

First, we describe our conceptual understanding of the *temporal* and *spatial merge-aware indexing approach* with examples, which essentially handle the indexing of the ellipse by first handling the temporal indexing in the array list followed by hierarchical spatial indexing. After indexing, at the very end of the leaf node, we introduce the overlapping condition maximal union merge-based criterion along with its algorithm and execution trace.

Temporal Merge Aware Indexing: Traditional sorting algorithms (e.g., quicksort) have the interesting property of comparing and sorting elements in $\mathcal{O}(n \log n)$ time which proves to be computationally expensive, especially in case of large data volume. In this chapter, we use *comparison-less* sorting by assuming n gap elements in the start time (t_{start}) range 1 to k such that each gap can be indexed in an input array. We initiate our analysis by defining a global start time, denoted as t_{start}, and a global end time, denoted as t_{end}. These temporal bounds are subdivided into intervals of one second each, serving as indices within an array list. The value assigned to each index corresponds to t_{start}. Each time segment (t_{start}, t_{end}) is efficiently indexed via binary search, exhibiting O($nlogn$) time. Subsequently, we also maintain a monotonic stack with (t_{start}, t_{end}) elements sorted by t_{start}, efficiently comparing the t_{end} of the current gap and the t_{start} of the previous gap. For instance, in Figure 5.6, $\langle A \rangle$ to $\langle H \rangle$ can be indexed based on t_{start} where we can get a sorted order of trajectory gaps $\langle C \rangle$,$\langle B \rangle$, $\langle A \rangle$,$\langle D \rangle$, $\langle E \rangle$, $\langle F \rangle$, $\langle H \rangle$, and $\langle G \rangle$. We can now

perform a plane sweep to an already sorted list and compare the end-time and start-time of the gap, similar to the traditional Memo-AGD [70]. Although this operation still requires quadratic computations, it will more efficiently pre-process time.

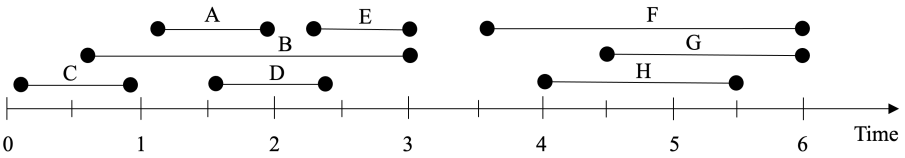


Figure 5.6: Temporal-Merge Aware Indexing of trajectory gaps

Spatial Merge Aware Indexing: A spatial index organizes access to data so that spatial objects can be found quickly without searching in a given spatial partition. It also enables the indexing of multidimensional objects, which can drastically speed up GIS operations like intersections and joins. For instance, R-Trees organize data in a tree-shaped structure, with a minimum orthogonal bounding rectangle (MOBR) at the nodes. MOBRs indicate the farthest extent of the data and can be indexed easily, but they can lead to "false positives" (i.e., MOBRs may intersect when the precise geometries (geo-ellipses) do not actually intersect). We use this space-time access method to compare spatially proximate gaps efficiently.

Index Generation: We first calculate the minimum and maximum values for xmin, ymin, xmax, and ymax across all MBRs to create a global rectangle that encompasses all individual MBRs. This effectively sets the stage for a hierarchical organization in which each node represents a spatial subdivision, thereby enhancing query performance through spatial locality. We have found that operations such as node splitting, bounding box updates, and overflow handling are better managed by estimating the overall spatial extent of the study area. Moreover, this approach efficiently handles a wide range of queries, including point queries, range queries, and nearest neighbor searches, by exploiting the spatial hierarchy and locality inherent in the global rectangle and its subdivisions.

Insertion: After checking for temporal overlaps, we check whether the geo-ellipse of two objects intersects or not. However, certain geo-ellipses are not involved in any groups (e.g., $\langle F \rangle$, $\langle G \rangle$, and $\langle H \rangle$ in Figure 5.7), resulting in high computational cost. For instance, in Figure 5.7, $\langle A \rangle$, $\langle B \rangle$, and $\langle C \rangle$ intersect in space and time, but $\langle F \rangle$, $\langle G \rangle$, and $\langle H \rangle$ are spatially distant. Hence, future gaps that are closer to $\langle F \rangle$, $\langle G \rangle$, and $\langle H \rangle$ but distant from

$\langle A \rangle$, $\langle B \rangle$, and $\langle C \rangle$ will only perform intersections within their defined spatial partition (i.e., L_1). Therefore, indexing such isolated gaps using hierarchical data structures may reduce quadratic comparisons.

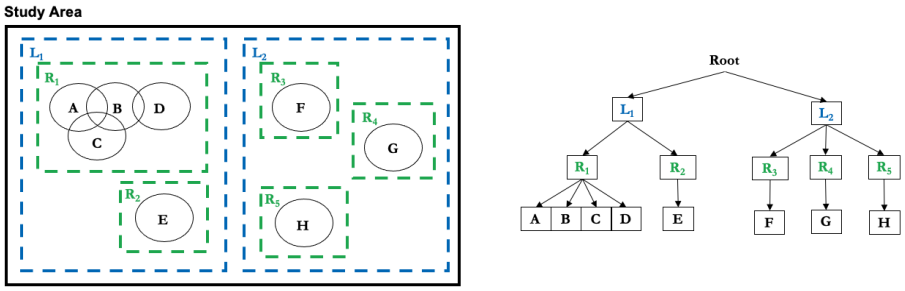


Figure 5.7: Spatial Merge Aware Indexing of trajectory gaps

Figure 5.7 shows gaps $\langle A \rangle$ to $\langle H \rangle$ spatially indexed at various hierarchical levels L_1 and L_2, where gaps $\langle A \rangle$ to $\langle E \rangle$ are indexed in L_1, and $\langle F \rangle$ to $\langle H \rangle$, which are isolated, are indexed in L_2. Suppose gap $\langle H \rangle$ needs to satisfy the criteria of spatiotemporal joins, which can be avoided by comparing the maximal subsets of $A \cap B \cap C$ and $D \cap E$ via indexing at L_2. Then, $\langle H \rangle$ can only be compared with $\langle F \rangle$ and $\langle G \rangle$, along with a few logarithmic-level comparisons, thereby reducing operations from quadratic to logarithmic complexity.

In addition, maximal intersection criteria often result in multiple isolated (or independent) gaps, which are computationally expensive to compare and evaluate for abnormal gap measures during future intersection operations. For instance, in Figure 5.10, if $\langle D \rangle$ partially intersects with the maximal set $\langle A, B, C \rangle$, it can be included in the maximal set to reduce the number of isolated gaps, as discussed in the following section.

Maximal Union Merge-based Criteria

We first index the gaps and then determine if they intersect with the largest merged unit (maximal union) within the merged group. If they do, they are then merged with the group, but if not, they are considered isolated (or independent) gaps. This leads to a large number of isolated gaps, even in denser regions, which makes it difficult to perform spatial comparison operations and AGM computations. In addition, if all gaps are stored in a single index, leading toward high space and time complexity. This can negatively impact spatial or spatiotemporal indexing methods (e.g., 3D R-Trees). To overcome this issue, we first perform spatiotemporal indexing followed by proposing a *maximal union*

merge-based criterion to form maximum contiguous merged regions by using the union and intersection properties of gaps while taking into account the DO threshold value. More details in Algorithm 6.

Step 1: First we initialize the AGM list and Non-Observed List to model every gap G_i as a geo-ellipse (i.e., Polygon(G_i)) along with an EMP (Effective Missing Period) based on $G_i^{EMP} = [G_i^{t_{start}}, G_i^{t_{end}}]$. We further initialize the Temporal index list $\in [\min(t_{start}), \max(t_{end})]$ as a filtering step for performing the linear scan, which results in avoiding comparison-less operations.

Step 2: For each $G_i^{EMP} \in [t_{start}, t_{end}]$, we use t_{start} to index the gap's position in the temporal index list. We then perform a linear scan to gather all the gaps G_i and save them in a Non-observed list to preserve *monotonicity* based on the increasing order of the timestamp. For each G_i in the Non-Observed list, we compare the DO threshold (λ) with the topmost gap in the Local Observed List and, if the condition satisfies the threshold, we add the pair to a list of candidate pairs of $G_i^{CP} \in [G_i, G_j, ..., G_n]$. These candidate pairs in G_i^{CP} are then spatially indexed using widely used indexing (e.g., R-Trees) or any other hierarchical spatial partitioning method.

Step 3: Next, we perform spatial intersections of geo-ellipse based on the *maximal union merge criterion* on the existing groups for gaps that reside within the leaf node and satisfy DO threshold condition and within the same partition at the leaf node level of the hierarchical spatial index. For each G_i, we then compare each gap G_j in G_i^{CP} (candidate pairs) and check the *maximal union merge criterion* based on the *Intersection* and *Union* group. The criterion states that if gap G is not intersecting with intersection bounds (IB) but is intersecting with the union bound (UB) of the maximal group, we then merge the gap pair with the union of the maximal group. For instance, the pseudo-code related to lines 17-22 shows the maximal union criterion where we first check if each gap G_i is intersecting with both Union and Intersection of the maximal group. If it is, we first merge, then terminate and move on to the next candidate pairs, otherwise we further check if gap G is intersecting with the Union of the maximal group. If it is, we merge all participating gaps and create MOBRs for effective AGM computations. This provides a stricter condition to make maximal groups and lower the count of isolated gaps.

Step 4: For each gap $G_i \in$ maximal union group, we perform AGM computation via

Algorithm 6 Space-Time Aware Indexing Algorithm

Input:
A Study Area S and A Set of Gaps [G_1,...,G_n]
An Estimate Signal Coverage Area MP
Participation Index $\leftarrow \lambda$ and Significant Score $\leftarrow K$
Output:
Abnormal Gaps with Top K AGM[G_1,..,G_k]

procedure :
  Step 1: Initialize AGM list $\leftarrow \emptyset$, Polygon(G_i), and Non Observed List $\leftarrow \emptyset$
  Temporal Index List $\leftarrow [min(t_{start}), max(t_{end})]$ $\forall$ gaps $\in G_i$
  Step 2: Perform temporal indexing and check overlap $\forall$ $[t_{start}, t_{end}] \in G_i^{EMP}$
  Index $t_{start} \in G_i^{EMP}$ in Temporal Index List
  Insert G_i in Non Observed List to preserve *monotonic property*
  for each: $G_i \in$ Non Observed List **do**
    $G_j \leftarrow$ topmost element from local observed list
    Local Observed List $\leftarrow$ Global Observed List
    while Local Observed List $\neq \emptyset$ **do**
      if $G_j^{t_{end}} \leq G_i^{t_{start}}$ where $[t_{start}, t_{end}] \in$ both G_i and G_j **then**
        Add G_j in potential candidate gap pairs $\in G_i^{CP}$
      end if
      Remove G_i^{CP} from Local Observed List
    end while Add G_i to Global Observed List
    Step 3: Perform Spatial Indexing and Check Maximal Union Criteria
    Perform Hierarchical Spatial Indexing
    for each: $G_j \in$ candidate pairs of $G_i \in G_i^{CP}$ **do**
      if Polygon(G_i) $\cap$ Union and Intersection Group of G_j and DO $\geq \lambda$ **then**
        Merge G_i over union group $\in G_j$ and update both G_i^{IB} and G_j^{UB}
        Terminate and skip towards next iteration
      end if
      if Polygon(G_i) $\cap$ Union Group of G_j and DO $\geq \lambda$ **then**
        Merge G_i over Union group $\in G_j$ and update G_j^{UB}
      end if
    end for
  end for
  Step 4: Compute AGM Score $\forall$ G_i **in Observed List**
  for each: $G_j \in$ Observed List **do**
    if G_j^{count} = Number of gaps in the subset ¡G_j¿ **then**
      AGM List $\leftarrow AGM(MP \cap G_j)$
    end if
  end for
  Step 5: Return K Significant Abnormal Gaps
  return Top K gaps [G_1,..,G_k]based on AGM Score
end procedure

linear scan operation of the maximal union group (e.g., $A \cup B \cup C$), intersection group (e.g., $A \cap B \cap C$), and isolated gaps (e.g., A) intersection with the signal coverage map SCM and subsequently save all merged gaps in the Observed List.

Step 5: Extract abnormal gaps based on the AGM Score.

Execution Trace: Figure 5.8 shows the execution trace of Steps 1–2 of Algorithm 6. Step 1 initializes gaps $\langle A \rangle$, $\langle B \rangle$, $\langle C \rangle$, $\langle D \rangle$, and $\langle E \rangle$ with a pre-defined EMP $\geq \theta$, along with additional *a priori* variables as defined in Memo-AGD [70], namely G^{CP}, G^{UB}, and G^{IB}. The Non-Observed List is initialized as $\emptyset$, and the Temporal Index List ranges over $[\min(t_{\text{start}}), \max(t_{\text{end}})]$ for all gaps, as shown in Figure 5.8 (e.g., $t_{\text{start}} \in \langle A \rangle$, $t_{\text{end}} \in \langle E \rangle$).

For *comparison-less* sorting, Step 2 performs a linear scan toward each gap's t_{start} in the Temporal Index List to populate the Non-Observed List, as illustrated in Figure 5.8. We then check whether the temporal overlap condition is satisfied by comparing the previous gap's start time (t_{start}) to the current gap's end time (t_{end}), using a temporary data structure. In Figure 5.8, we use a *stack* to compare prior elements with current elements from the Local Observed List, and *cache* them in G_i^{CP} if the temporal condition holds.

For instance, at $i = 1$, $t_{\text{end}} \in \langle B \rangle$ is compared to $t_{\text{start}} \in \langle A \rangle$ in the stack, and $\langle A \rangle$ is stored in $G^{CP} \in \langle B \rangle$ (resulting in $G_i^{CP} \in \langle A, B \rangle$ and $G_j^{CP} \in \langle A, B \rangle$) since it satisfies the temporal overlap condition. Similarly, for each gap, we store G^{CP} for spatial comparison at the leaf level of the spatial index.

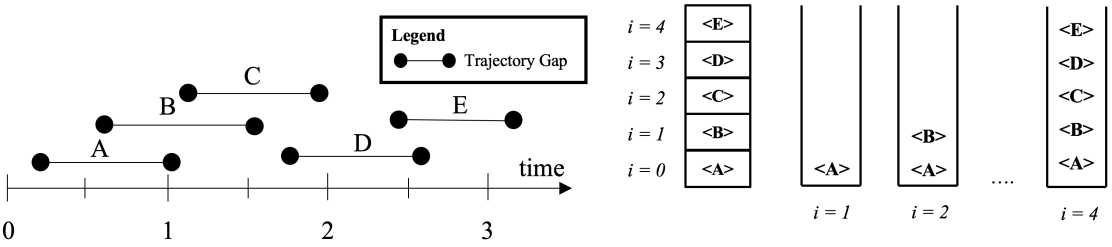
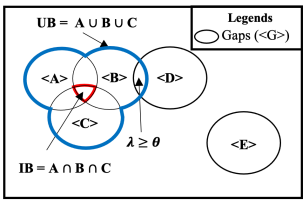
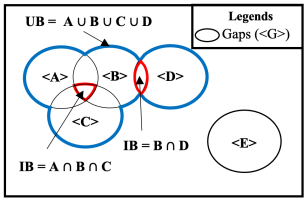


Figure 5.8: Spatial Merge-Aware with Maximal Union Merge Criteria Execution Trace (Step 1-2)

Figure 5.9 illustrates the execution trace of the maximal union merge criterion in Algorithm 6. After hierarchical indexing, as explained in Figure 5.7, we focus on gaps $\langle A \rangle$ to $\langle E \rangle$ which reside in the same index and are associated with candidate pairs G_i^{CP},

subsequently intersected with IB and UB. For instance, gaps $\langle B \rangle$ and $\langle C \rangle$ satisfy IB with $\lambda \geq \theta$. However, the G_i^{CP} of $\langle D \rangle$ only intersects with the UB of $\langle B \rangle$ (i.e., $\langle A \cup B \cup C \rangle$). If $\langle D \rangle \cap (A \cup B \cup C)$ exceeds the DO threshold λ, then it will be merged into the union set $A \cup B \cup C \cup D$. Similarly, if the G_i^{CP} of $\langle E \rangle$ does not spatially intersect with $\langle D \rangle$, the algorithm terminates.





Trajectory Gaps (G)	Candidate Pairs (CP)	Intersection Bounds (IB)	Union Bounds (UB)
<A>	<B,C>	<A,B,C>	<A,B,C,D>
<B>	<A,C>	<A,B,C>, <B,D>	<A,B,C,D>
<C>	<A,B>	<A,B,C>	<A,B,C,D>
<D>	<D>	<B,D>	<A,B,C,D>
<E>	<E>	<E>	<E>

Figure 5.9: Spatial merge-aware with maximal union merge criteria execution trace (Step 3)

5.5.2 Dynamic Region Merge (DRM) Approach

Our method of caching temporal information and spatial indexing reduces comparison operations prior to performing AGM computations. In addition, maximal union merge criteria reduce the total number of merged gap unions to aid cells in polygon interactions for the AGM computations. However, the maximal union criterion relies on a DO threshold and the nature of the data distribution, i.e., whether the distribution of gaps is uniform or skewed. For instance, if a gap distribution is skewed and no gaps are involved in any spatiotemporal intersection operations, then all gaps will reside in a single index. This results in a large number of isolated gaps, which also increases cell in polygon operations as shown in Figure 5.5b. Hence, we propose two more efficient methods for finding significant AGM scores based on maximal groups considering the sliding window approach and early termination criterion.

Sliding Window with Memoization Approach

Given multiple gaps intersecting within a spatial partition, we compute AGM scores incrementally whenever a new gap intersects a maximal group spatially. In this technique, we cache the previously computed AGM cells for each maximal group defined by the maximal union merge criterion, which permits intersection and union criteria thresholds. Hence, we only need to append the reported signal coverage cells associated with the new

gap and avoid the need to recompute the AGM scores of the entire maximal subgroup. The sliding window technique can achieve this, along with caching current cells involved with the group in a linear time (similar to the linear scan) but caching ensures fewer cells in polygon comparisons. An example of the approach is illustrated in Figure 5.10 as follows:

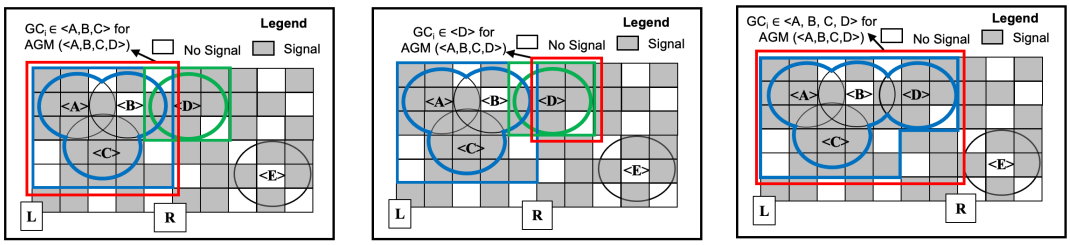


Figure 5.10: Proposed sliding window approach to compute AGM incrementally

Figure 5.10 shows an illustration of a sliding window with caching, where gaps $\langle A \rangle$, $\langle B \rangle$, and $\langle C \rangle$ are part of maximal subsets. We initialize the left (L) and right (R) pointers around the Minimum Orthogonal Bounding Rectangle (MOBR) defined for the polygon Polygon($\langle A, B, C \rangle$). The AGM score is computed based on the cells enclosed within the MOBR. For example, if gap $\langle D \rangle$ intersects $A \cap B \cap C$ above the Decision Overlap (DO) threshold λ, we can merge $\langle D \rangle$ with the group $\langle A \cup B \cup C \rangle$ based on the *maximal union merge criterion*. In this case, we would need to recompute the AGM score for $\langle A \cup B \cup C \cup D \rangle$ (see leftmost diagram in Figure 5.10). To eliminate such recomputation redundancy, we cache the cells belonging to $\langle A, B, C \rangle$ and incrementally append only $GC_i \in \langle D \rangle$ to the cached $GC_i \in \langle A, B, C \rangle$.

Incremental Spatial Indexing: The minimum orthogonal bounding rectangle (MOBR) of the group $\langle A, B, C \rangle$ is defined by the coordinates (x_1, y_1), (x_2, y_1), (x_1, y_2), and (x_2, y_2). The left pointer (L) spans the range $[(x_1, y_1), (x_1, y_2)]$, and the right pointer (R) spans $[(x_2, y_1), (x_2, y_2)]$. When an intersecting MOBR from gap $\langle D \rangle$ is encountered, we perform a union of $\langle A, B, C \rangle$ and $\langle D \rangle$, and update the minimum $(x_{\min}, y_{\min})$ and maximum $(x_{\max}, y_{\max})$ coordinates from the combined MOBRs, resulting in the merged group $\langle A, B, C, D \rangle$. If a union occurs, the right pointer (R) is updated to the new coordinates $[(x_2^{\text{new}}, y_1^{\text{new}}), (x_2^{\text{new}}, y_2^{\text{new}})]$; otherwise, the left pointer (L) is updated to $[(x_1^{\text{new}}, y_1^{\text{new}}), (x_1^{\text{new}}, y_2^{\text{new}})]$. We index the grid cells between the updated L and R pointers and use them to recompute the AGM score for the merged group.

Early Termination Criteria:

Since the proposed criterion helps reduce isolated gaps by relaxing the strictness of merge conditions through union operations with maximal subgroups, it may also result in a long chain of gaps. This chaining could potentially lead to false positives or false negatives under the AGM criterion. For instance, if there is variability in the signal coverage map (as shown in Figure 5.13 (b) and Figure 5.13 (c)), a new gap (e.g., $\langle D \rangle$) might fall below the defined threshold, causing the entire maximal subgroup to be incorrectly classified as abnormal or not. To avoid such false negatives, we introduce an early termination criterion that incrementally checks both the previous and updated AGM scores before finalizing the maximal union merge. This process is demonstrated in Equation 5.3.

$$||AGM(\langle G_i, ..G_k \rangle) - AGM(\langle G_j \rangle)|| \geq \delta \tag{5.3}$$

where δ is the threshold for determining whether the difference between the previous and new AGM scores is significant. If it is not, we merge the groups $\langle G_i, \ldots, G_k \rangle$ and $\langle G_j \rangle$ and compute AGM($\langle G_i, \ldots, G_k \rangle \cup \langle G_j \rangle$). Otherwise, we compute the AGM scores separately, i.e., AGM($\langle G_i, \ldots, G_k \rangle$) and AGM($\langle G_j \rangle$).

AGM-based Decreasing Queue:

Given multiple merged regions in some leaf node, we can further use the *monotonic property* mentioned while indexing the temporal gap start times, which results in *comparison-less* sorting. Here we use a decreasing queue (deque) data structure, which maintains the non-increasing order of AGM scores of the merged groups. The main intuition is to perform an on-the-fly comparison of the AGM score of incoming gap G_i with the AGMs of gaps that are already sorted in decreasing order such that Equation 5.3 terminates the comparison at an early stage. For instance, a gap G_i with a high AGM score is likely to merge with groups having a high AGM score such that Equation 5.3 is satisfied.

Figure 5.11 illustrates the decreasing queue approach, where the AGM scores for merged gaps $\langle A, B, C \rangle$, gap $\langle D \rangle$, and gap $\langle E \rangle$ are 0.84, 0.23, and 0.33, respectively. Consider gap $\langle F \rangle$ undergoing spatial comparison with the groups $\langle A, B, C \rangle$, $\langle D \rangle$, and $\langle E \rangle$. With a specified threshold $\delta = 0.15$, it becomes evident that $\langle F \rangle$, having an AGM score of 0.83, is more likely to merge with the group $\langle A, B, C \rangle$ than with $\langle D \rangle$ or $\langle E \rangle$, due to the similarity in AGM

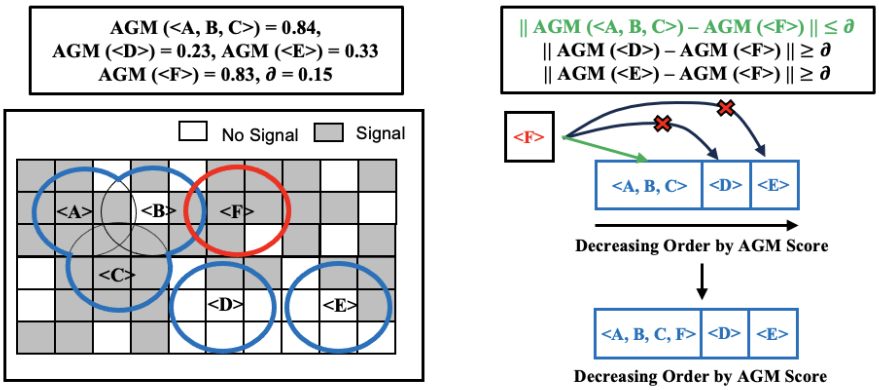


Figure 5.11: Proposed Decreasing Queue strategy to sort gaps in leaf node by decreasing order

values. Organizing all Minimum Orthogonal Bounding Rectangle (MOBR) coordinates $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ into a decreasing queue, sorted by descending AGM values, enabling early termination of comparisons. For example, $\langle F \rangle$ only needs to compare with $\langle A, B, C \rangle$ and can terminate further comparisons upon reaching $\langle D \rangle$, as

$$|\text{AGM}(\langle A, B, C \rangle) - \text{AGM}(\langle F \rangle)| \geq \delta.$$

This optimization reduces the number of comparisons from n to k (where $k < n$), thereby improving computational efficiency.

Doubly-Linked List Decreasing Queue:

AGM-based Decreasing Queues offer early comparisons with trajectory gaps (or merged groups) that have high AGM scores, enabling early termination. However, the comparison of the candidate gap followed by insertion and its adjustment towards the appropriate index to preserve monotonicity results in-place sorting (e.g., insertion-sort). Consequently, each candidate gap adjustment and array resizing incurs $O(n \log n)$ or $O(n)$ operations, which prove to be computationally expensive in a scalable environment. To address this limitation, we introduced a Doubly Linked List within the Decreasing Queue. This provides dynamic insertion and deletion of candidate gaps with on-the-fly pointer updates, achieving a constant $O(1)$ time operation and avoiding adjustments in the Decreasing Queue.

Figure 5.12 illustrates an example of a trajectory gap, denoted as $\langle F \rangle$, with a specific

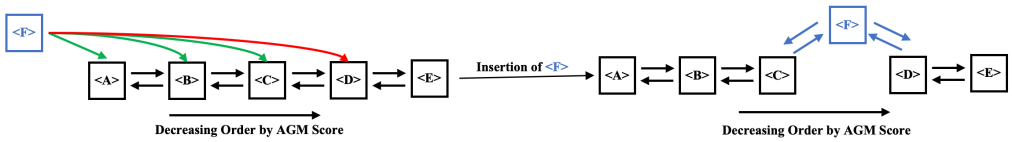


Figure 5.12: Proposed Doubly Linked List on Decreasing Queue for O(1) insertion

AGM score, being inserted into a doubly linked list ordered by decreasing AGM values. We check for the early termination condition while performing on-the-fly spatial intersection involving gap $\langle F \rangle$. Assuming gap $\langle F \rangle$ does not intersect with any other gaps and $\text{AGM}(\langle F \rangle) \geq \text{AGM}(\langle D \rangle)$, we update both the left and right pointers of $\langle F \rangle$ relative to its neighboring gaps (i.e., between elements $\langle C \rangle$ and $\langle D \rangle$, which have higher and lower scores, respectively). During this update, the next pointer of $\langle C \rangle$ is set to $\langle F \rangle$, and the previous pointer of $\langle F \rangle$ is set to $\langle C \rangle$. Similarly, the next pointer of $\langle F \rangle$ points to $\langle D \rangle$, and the previous pointer of $\langle D \rangle$ is updated to $\langle F \rangle$. This results in a constant-time insertion of $\langle F \rangle$ without requiring in-place adjustment of gaps $\langle D \rangle$ and $\langle E \rangle$.

Algorithm 7 explains in detail.

Step 1-3: Same as the Algorithm 1

Step 4: For each gap G_i in the spatial index, we perform a sliding window operation over the qualified gaps which satisfy the maximal union merge criteria by adding only new $GC_i \in G_i$ and then recomputing the AGM score for the entire union-merged group and also for the new gap. For instance, line 7 shows the overall operation after G_i satisfies the maximal union merge criteria with the Union Group and appends new $GC_i \in G_i$. The sliding window operation ensures linear operations are given prior information about the left and right pointers along with GC_i. We then perform early termination based on Equation 5.3, where we check if the change in prior AGM score (i.e., union group $\notin G_i$) with new AGM score (i.e., G_i). If the change is not significant, we consider AGM of maximal union group $\in G_i$; otherwise, we compute the AGM separately.

Step 5: Same as the Algorithm 1

Execution Trace: Given gaps $\langle A \rangle$, $\langle B \rangle$, $\langle C \rangle$, $\langle D \rangle$, and $\langle E \rangle$ within the same index, we first initialize the left (L) and right (R) pointers at the boundary of the MOBR along the group $\langle A, B, C \rangle$, where $\langle A, B, C \rangle$ satisfies the Intersection Bounds (IB) criterion. If gap $\langle D \rangle$ intersects with $\langle A, B, C \rangle$ and $\lambda \geq \theta$, we increment the right pointer (R), provided that

Algorithm 7 Dynamic Region Merge Algorithm

Input:
A Study Area S and A Set of Gaps [G_1,...,G_n]
An Estimate Signal Coverage Area MP
Participation Index $\leftarrow \lambda$ and Significant Score $\leftarrow K$
Output:
Abnormal Gaps with Top K AGM[G_1,..,G_k]

procedure :
  Step 1 to Step 3 are same as Algorithm 1
  Step 4: Compute AGM Score $\forall$ G_i **in Observed List**
  for each: $G_i \in$ Spatial Index **do**
    Perform sliding window increment to add $GC_i \in G_i$
    AGM (G_i) $\leftarrow$ AGM($GC_i \in G_i$)
    AGM ($G_i \cap$ Union Group) $\leftarrow$ AGM ($GC_i \in$ Union Group + $GC_i \in G_i$)
    if $G_i \in$ Union and/or Intersection Group **then**
      if AGM($G_i \cap$ Union Group) - AGM(G_i) $\leq \delta$ **then**
        AGM Decreasing Queue $\leftarrow$ AGM($G_i \cap$ Union Group)
      else Compute separate AGM for $G_i \cap$ Union Group and G_i
      end if
    end if
  end for
  Step 5: Return Abnormal Gaps to the Human Analysts
  return [G_1,..,G_k]based on Priority Threshold PI
end procedure

the early termination criterion is satisfied. For example, the middle box in Figure 5.13 (b) shows that the difference between AGM($\langle A, B, C\rangle$) and AGM($\langle D\rangle$) is smaller than the predefined threshold δ, consistent with Equation 5.3. However, the rightmost box in Figure 5.13 (c) shows a case where Equation 5.3 is not satisfied for $\langle D\rangle$ due to a change in the signal coverage map, which alters the abnormal gap measure for both $\langle A, B, C\rangle$ and $\langle D\rangle$. Consequently, AGM($\langle A, B, C\rangle$) and AGM($\langle D\rangle$) are computed separately, and both the left and right pointers are incremented by one position.

5.6 Theoretical Evaluation

We evaluate our proposed algorithm STAGD for correctness and completeness in Section 5.6.1. Section 5.6.2 further analyzed STAGD's time complexity compared with the baseline Memo-AGD for both average and worst case.

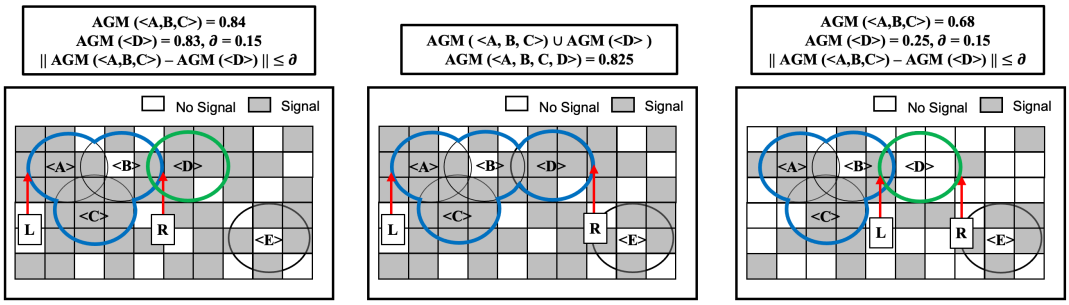


Figure 5.13: Dynamic Region Merge Execution Trace

5.6.1 Correctness and Completeness

Lemma 5.6.1. The space-time-aware gap detection and dynamic region merge algorithms are correct.

Proof. Given a finite set of $|N|$ trajectory gaps, space-time aware gap detection (STAGD) loops around a finite number of times and terminates. The algorithm prunes on each trajectory gap and then conducts a spatial-temporal intersection producing a finite number of candidate pairs.

In addition, the intersection boundary IB within candidate pairs does not allow pairs whose IB does not intersect with any of the candidate pairs. For instance, if $G_i^{IB} \in Polygon(G_1) \cap Polygon(G2)$ and $G_j^{IB} \in Polygon(G_3)$, then $G_j^{IB} \cap G_i^{IB} \leftarrow Polygon(G_1) \cap Polygon(G_2) \cap Polygon(G_3) \neq \emptyset$. This proves that $Polygon(G_1) \cap Polygon(G_2)$, $Polygon(G_2) \cap Polygon(G_3)$ and $Polygon(G_1) \cap Polygon(G_3) \neq \emptyset$ since the overall intersection of $Polygon(G_1) \cap Polygon(G_2) \cap Polygon(G_3) \neq \emptyset$.

In STAGD, the Union Boundary (UB) considers candidate pairs which involved in intersection with other gaps within the entire subgroup (i.e., $Polygon(G_1) \cap Polygon(G_2)$ or $Polygon(G_2) \cap Polygon(G_3)$ or $Polygon(G_1) \cap Polygon(G_3) \in \emptyset$). This may result in long chains which can be discontinued or broken by any of the proposed interest measures, such as the DO threshold and early termination criteria defined within the chapter.

In terms of AGM computations correctness, the proposed Dynamic Region Merge (DRM) Algorithm provides a correct output within a given signal coverage map by considering signal coverage variability at a given defined geographic space. The proposed interest measures (e.g., DO threshold) and early termination criteria ensure correct AGM computations over

derived spatial bounds from both STAGD and DRM algorithms. Hence STAGD and DRM are correct. □

Lemma 5.6.2. Space time-aware gap detection and dynamic region merge are complete.

Proof. Given a finite set of $|N|$ gaps, both STAGD and DRM ensure complete coverage by including all gaps within their respective candidate pair sets. Specifically, the total number of gaps $G_1, G_2, \ldots, G_n$ included in the selected subsets matches the number of gaps listed in the Non-Observed List. For example, the execution trace of a Space-Time Aware Gap Detection algorithm may output groups such as $\langle A, B, C, D\rangle$ and $\langle E\rangle$, which collectively correspond to the Non-Observed List entries $\langle A\rangle$, $\langle B\rangle$, $\langle C\rangle$, $\langle D\rangle$, and $\langle E\rangle$. This demonstrates the completeness of the Space-Time Aware Gap Detection approach. Furthermore, all gaps, whether part of a maximal set (e.g., $\langle A, B, C\rangle$) or independent sets (e.g., $\langle A\rangle$, $\langle B\rangle$, $\langle C\rangle$) which are accounted for. Therefore, both STAGD and DRM satisfy the criterion of completeness. □

5.6.2 Asymptotic Time Complexity

We briefly discuss the asymptotic complexity of baseline Memo-AGD and the proposed STAGD+DRM algorithm based on preprocessing gap enumeration and AGM computations. However, both algorithms heavily depend upon gap distribution within a study area which indirectly depends upon the worst and best case. For instance, if independent gaps (i.e., gaps that are not participating in any maximal sets) have skewed distribution at a given index, then both algorithms has the same performance asymptotically. However, if gaps are merged and/or in uniform distribution, the proposed algorithms out-performs the baseline via indexing and DRM optimization. Hence, we will consider both the worst (a skewed distribution with independent gaps) and average-case (merged and independent gaps) for both algorithms.

Memo-AGD: Given n gaps, sorting operations require $\mathcal{O}(n \log n)$ time based on the t_{start} of the gap. The Memo-AGD use caching based on IB resulting in k maximal sub-groups in quadratic time-complexity. In the case of AGM computations, given k merged gaps and M X N grid cells, results in $k \times \mathcal{O}(M \times N)$ where (k $\langle\langle$ N). However, in the worst case, $\mathcal{O}(n^2)$ and $n \times \mathcal{O}(M \times N)$ is performed for gap enumeration and AGM computations, respectively.

STAGD + DRM: Given n gaps, the space time-aware gap detection algorithm uses comparison-less sorting based on t_{start} in linear time (i.e., $\mathcal{O}(n)$) followed by *gap enumeration* and *AGM computation*. In the case of gap enumerations, we index each gap in $\mathcal{O}(n \log n)$ time along with k linear comparisons at the leaf-node level, where k are merged groups ($k \langle\langle n$). This results in overall $\mathcal{O}(n \log n) + \mathcal{O}(n \times k)$. In the case of AGM computations, the cell in polygon operation will speed up due to sliding window and memoization in the average case. However, if all gaps are independent and has skewed distribution in a single index (i.e., worst case), results in $\mathcal{O}(n)^2$ in comparison operations. In addition, given N gaps and M cells, we perform $N \times M$ operations in the worst case (where no gaps are intersecting). If gaps are involved in union operations with k maximal groups, we perform $k \times M \times N$ operations where $k \langle\langle N$. Hence, the worst and average time complexity of DRM is $n \times \mathcal{O}(M \times N)$ and $k \times \mathcal{O}(M \times N)$, respectively.

5.7 Experimental Validation

In this section, we present the comprehensive design of the experiments, including the related work, baseline, and proposed algorithms, which were outlined in Section 5.7.1. We also discuss the real-world dataset [1] and the synthetic data generation method used for evaluating the solution quality of the algorithms. Furthermore, in Section 5.7.2, we present the results of the experiments, which were based on accuracy and computation time, and varied depending on different parameters, such as the number of gaps and GPS points.

5.7.1 Experiment Design

Comparative Study: The goal of the experiments was to assess and compare the solution quality of three methods: Linear Interpolation (Related Work), K Nearest Neighbor Imputation (Related Work), and STAGD+DRM (Proposed). In the case of execution time, we compared Memo-AGD (Baseline) and STAGD+DRM (Proposed). Both solution quality and computational efficiency were compared based on five parameters: the number of GPS points, the number of trajectory gaps, the effective missing period (EMP) (i.e., the minimum time period the object is missing), the average speed (S_{avg}), and the degree of overlap (DO) threshold. The detailed design of the experiments is presented in Figure 5.14.

KNN Imputation: In this chapter, we apply the k-Nearest Neighbors (k-NN) imputation method, previously utilized for anomaly detection in [71], to identify and address the

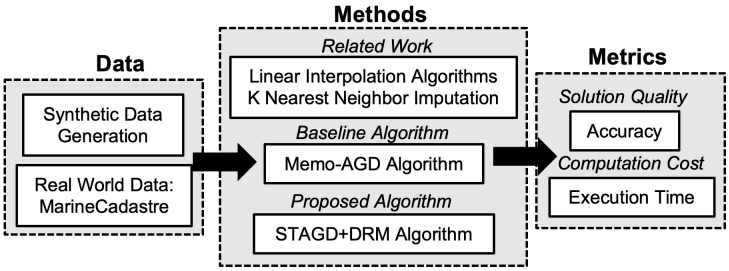


Figure 5.14: Experiment Design

issue of missing Automatic Identification System (AIS) messages, referred to as dropouts, for vessels at specific times. This method aims to reconstruct the missing data, including the estimated position of the vessel and the Received Signal Strength Indicator (RSSI). The vessel's position is initially estimated or predicted using tracking systems and models such as the Constant Velocity Model (CVM). Subsequently, we attempt to estimate the missing RSSI values. However, as RSSI data is not provided in [71], we employ a KNN imputer that calculates based on the inverse-weighted Euclidean distance from the predicted position to the fixed K neighbors. In our experimental setup, we chose K=5 and computed the weighted average from all existing neighbors to update the latitude-longitude coordinates derived using the Constant Velocity Model (CVM).

Real World Dataset: We utilized the real-world MarineCadastre [1] dataset to measure computational efficiency. The dataset contains various attributes, such as longitude, latitude, speed over ground, and others, for 150,000 objects that were recorded between 2009 and 2017. The geographical area covered by the data extends from 180W to 66W degrees longitude and from the 90S to 90N degrees latitude, based on the WGS 1984 coordinate system. In our experiments, we only considered the attributes of longitude, latitude, time, speed over ground (SOG), and Maritime Mobile Service Identity (MMSI). The maximum speed was calculated by taking the average SOG of the effective missing period (EMP). Figure 5.15 shows an Entity Relationship (E-R) Diagram for three entities: vessels, trips, and location signals. A vessel can take multiple trips, and each trip can emit multiple location signals.

Synthetic Dataset Generation: The experiment for solution quality, lacked ground truth data, meaning we had no information on whether a gap in a trajectory was normal or

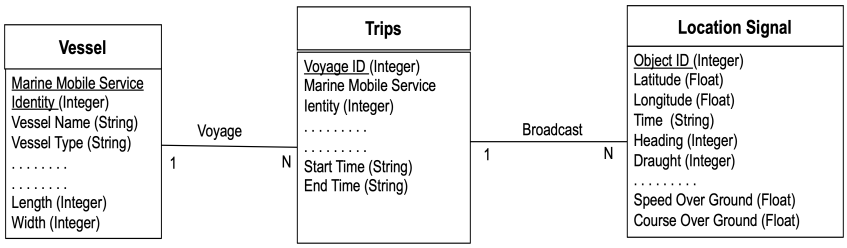


Figure 5.15: Entity Relation diagram of the MarineCadastre [1] dataset

abnormal. To overcome this, the proposed algorithms were evaluated using synthetic data created from a real-world dataset. The dataset consists of 1.25 x 10^5 trajectories collected in the Bering Sea, covering a study area from $179.9W$ to $171W$ degrees longitude and from $50N$ to $58N$ degrees latitude, with 1500 trajectory gaps between 2014 and 2016. Figure 5.16 shows the detailed methodology for synthetic data generation.

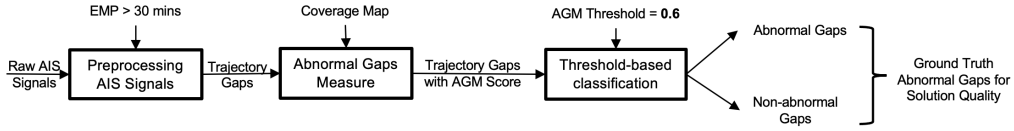


Figure 5.16: Synthetic Data Generation

The data was preprocessed to identify gaps lasting more than 30 minutes, which were considered trajectory gaps. The AGM (Abnormal Gap Measure) score was calculated using linear interpolation along with the proposed methods, and each gap was classified as either abnormal or non-abnormal based on an AGM threshold of 0.6. Gaps with an AGM score above 0.6 were deemed abnormal. The choice of the AGM score threshold was designed to reduce bias in the accuracy of the results, which was achieved by conducting multiple trial runs of both the baseline and proposed approaches.

Computing Resources: We performed our experiments on a system with a 2.6 GHz 6-Core Intel Core i7 processor and 16 GB 2667 MHz DDR4 RAM.

5.7.2 Experiment Results:

Solution Quality: The proposed STAGD+DRM approach was compared with a Linear interpolation and KNN Imputation method that is widely used in the literature [64] using space-time interpolation approach (also used in Memo-AGD [70]) in terms of accuracy. The accuracy metric is defined in Equation 5.4 as the ratio of the actual trajectory gaps involved

in abnormal behavior to the total number of abnormal gaps.

$$Accuracy = \frac{Actual\ Gaps\ involved\ in\ Abnormal\ Behavior}{Total\ Gaps\ involved\ in\ Abnormal\ Behaviour} \tag{5.4}$$

In Equation 5.4, the "Actual Gaps" in the numerator are the "true positives" (correct abnormal gaps) and "true negatives" (correct non-abnormal gaps) generated by the two baseline approaches from the related work (Linear Interpolation and KNN Imputation) and the proposed approach (STAGD+DRM). Total Gaps in the denominator are all the gaps that are correctly and incorrectly classified as abnormal in a given study area.

The detailed sensitivity analysis based on different parameters for the related work, baseline, and the proposed algorithms are as follows:

(1) GPS Points: First, we fixed the EMP at 60 mins, the number of objects (or vessels) at 500, the speed at 20 $meters/second$, DO threshold to 0.5, and varied the number of GPS points from 2.5×10^5 to 1.25 x 10^6. The results in Figure 5.17a show that linear interpolation under-performs against STAGD with Dynamic Region Merge(STAGD+DRM). Since DRM ensures to capture of more accurate gaps, especially in case of the high variability of signal coverage map. Linear Interpolation and KNN imputation captures abnormal gaps with less accuracy as compared to the proposed STAGD+DRM as GPS density grows.

(2) Objects (Vessels): We fixed the number of GPS points to 5.0×10^5, EMP to 60 mins, Speed at 20 $meters/second$, DO threshold to 0.5 and varied the number of objects from 2500 to 12500. Figure 5.17b shows that the proposed approach captures more accurate signal coverage in terms of maximal subgroups as compared to linear threshold since it does not consider the DO Threshold. Linear interpolation and KNN imputation is less accurate since it does not consider space-time interpolation, which later incorrectly classifies abnormal gaps.

(3) Effective Missing Period (EMP): In this experiment, we set the number of objects (or vessels) to 900, the number of GPS points to 5.0×10^5, the Degree Overlap (DO) threshold to 0.5, the speed to 20 $meters/second$, and varied the effective missing period (EMP) from 2 to 10 hours. Figure 5.17c shows that STAGD+DRM is again the more accurate since DRM provides a more precise capture of variable signal coverage in cases of large maximal union groups. With an increasing EMP, the larger gaps capture more variability in the spatial distribution of the reported cells, resulting in more accurate AGM

scores. Linear Interpolation and KNN imputation both underperform in this measure.

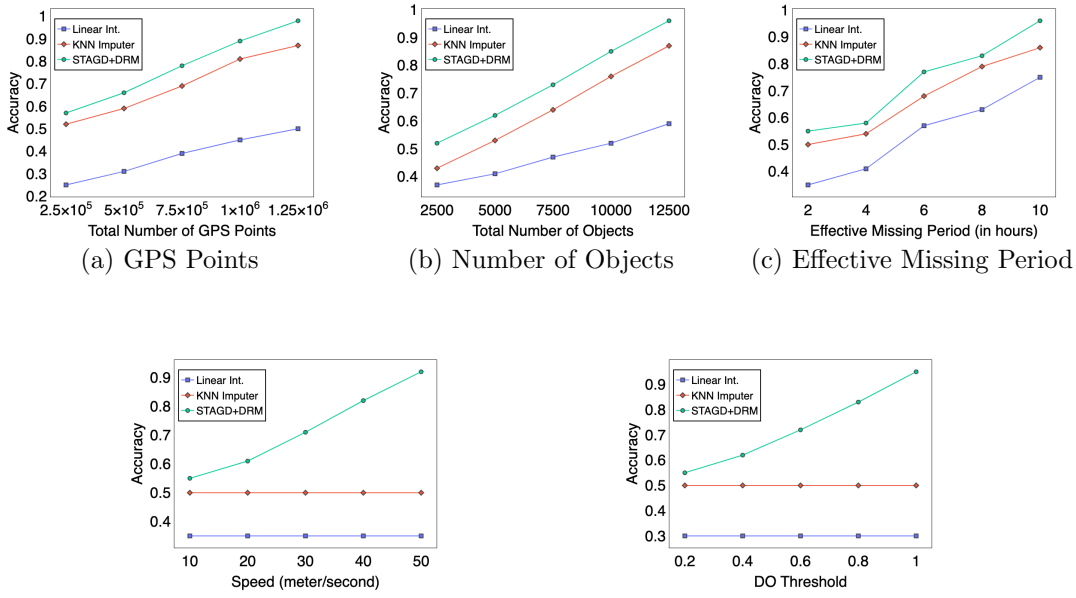
(4) Speed (meter/second): The number of objects was set to 900, GPS points to 5.0×10^5, DO threshold to 0.5, speed to 20 $meters/second$, and the S_{max} threshold was varied from 10 to 50 meters/second. Figure 5.17d shows similar trends as EMP increases, the size of geo-ellipses also increases which later results in more potential interactions with other geo-ellipses. In addition, large geo-ellipses also result in more spatial distributions in terms of signal coverage which is better captured by STAGD+DRM and preserving its correctness due to the early stopping criterion property. Linear interpolation and KNN imputation shows no benefit because it does not take speed into account.

(5) Degree Overlap (DO) Threshold (λ): Last, the degree of overlap (DO) threshold was varied from 0.2 to 1.0 while keeping the number of objects at 900, the number of GPS points at 5.0×10^4, the speed at 20 meters/second, and the EMP threshold at 60 mins. As seen in Figure 5.17e, STAGD+DRM algorithms display linear trends, with STAGD+DRM capturing signal coverage more accurately than linear interpolation. Linear interpolation and KNN imputation by contrast, shows no benefit because they both do not take the DO threshold into account.

Computational Efficiency: The proposed STAGD+DRM algorithm was compared to the baseline Memo-AGD in terms of computational efficiency using the MarineCadastre dataset [1]. Five different parameters were tested: number of GPS points, number of objects, effective missing period (EMP), maximum speed S_{max} (the maximum speed achieved within the object's missing time period), and degree overlap (DO) threshold. The runtime was monitored while considering the indexing and AGM computation cost for STAGD+DRM.

(1) Number of GPS Points: First, we set the effective missing period (EMP) to 60 minutes, the maximum speed S_{max} to 15 meters/second, the degree overlap (DO) threshold to 0.2, and varied the number of trajectory gaps from 500 to 2500. As shown in Figure 5.18a, STAGD+DRM consistently outperforms Memo-AGD due to comparison-less temporal indexing and reduced spatial comparisons in 3D R-Trees. Additionally, as the density of GPS points increases in the study area, the potential number of interactions between two or more geo-ellipses also increases.

(2) Number of Objects: In this experiment, we set the effective missing period



(a) GPS Points (b) Number of Objects (c) Effective Missing Period

(d) Speed (e) DO Threshold

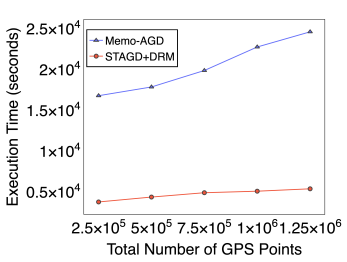
Figure 5.17: Proposed STAGD+DRM is always more accurate than Linear Interpolation and KNN-Imputation under varying parameters.

(EMP) to 60 minutes, the maximum speed S_{max} to 15 meter/second, the number of GPS points to 5.0×10^5, and the DO threshold to 0.2, and we varied the number of objects from 2500 to 12500. As shown in Figure 5.18b, the results are similar to when the number of GPS points was varied. STAGD+DRM outperforms Memo-AGD as it effectively indexes and computes AGM scores of gaps derived from different objects or vessels.

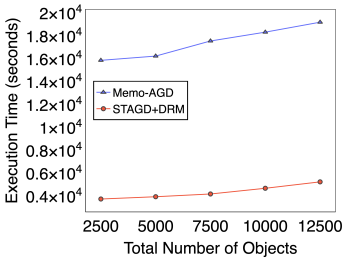
(3) Effective Missing Period (EMP): Next, we set the number of objects to 2500, the maximum speed S_{max} to 15 meters/second, the DO threshold to 0.2, and the number of GPS points to 5.0×10^5. We then varied the EMP threshold from 2 hours to 10 hours. As shown in Figure 5.18c, STAGD+DRM runs the fastest. Increasing the EMP results in larger geo-ellipses and, thus, more potential interactions and larger maximal groups. 3D R-Trees, allow large gaps to be indexed, leading to a skewed distribution of gaps within the index. Both Memo-AGD and STAGD+DRM use the DO threshold to avoid these issues. However, STAGD's Dynamic Region Merge also allows for union and intersection operations and memoizes grid cells, resulting in improved performance compared to Memo-AGD.

(4) Speed: We set the number of objects to 2500 and the effective missing period (EMP) to 2 hours the DO threshold was set to 0.2, and varied the maximum speed, S_{max}, from 10 meters/second to 50 meters/second. Again, as seen in Figure 5.18d, the results are similar to when the EMP was varied, as a high-speed value leads to larger geo-ellipses, resulting in more potential interactions. Furthermore, the proposed algorithm with STAGD+DRM's flexibility towards union and intersection operations outperforms the Memo-AGD algorithm.

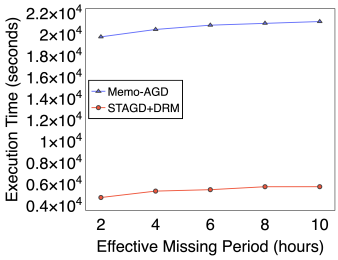
(5) DO Threshold (λ): Last, we set the number of objects to 2500, the effective missing period to 2 hours, the maximum speed to 15 m/s, the number of GPS points to 5.0×10^5, and varied the DO threshold from 0.2 to 1.0. The proposed approach still outperformed Memo-AGD (Figure 5.18e). A higher DO threshold meant that gaps were less likely to overlap, which led to fewer geo-ellipse clusters within the signal coverage area. While both Memo-AGD and STAGD+DRM allow for the merging of trajectory gaps, STAGD+DRM's early termination criteria also allow it to ignore all the isolated gaps within the same spatial index, thus saving time.



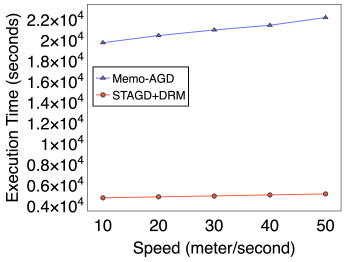
(a) GPS Points



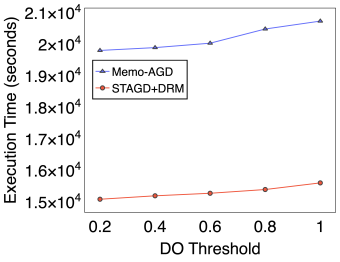
(b) Number of Objects



(c) Effective Missing Period



(d) Speed

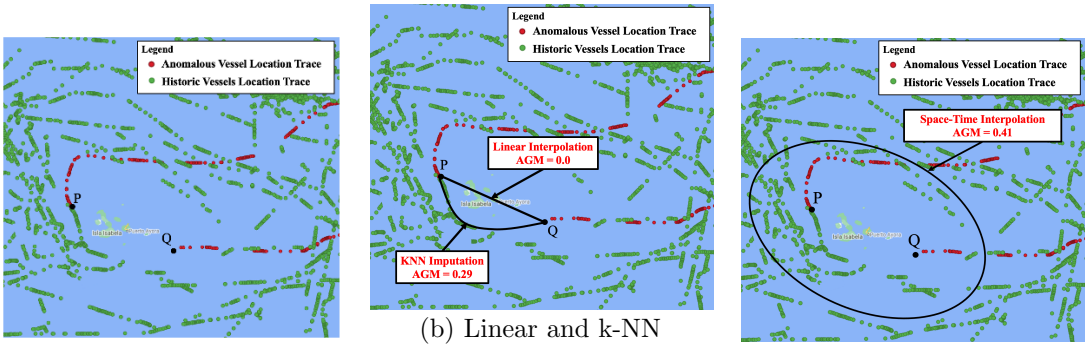


(e) DO Threshold

Figure 5.18: STAGD+DRM runs faster than Memo-AGD under different parameters

5.7.3 Case Study

We conducted a case study based on an actual event [13] where a fishing vessel switched off its transponder while entering a protected marine habitat and then switched it back on after 15 days. We applied linear (shortest path) interpolation, KNN imputation, and our proposed space-time interpolation based algorithms on a MarineTraffic, [67] dataset to find abnormal regions in a study area ranging from 48.3 W to 65.8 W degrees in longitude and from 23.4 N to 29.6 N degrees in latitude near the Galapagos Marine Reserve. Figure 5.19a shows that a ship switched off its transponder at point P in the Marine Reserve and switched it back on at point Q. Figure 5.19b assumes a linear interpolation where the vessel interacts with a small number of islands situated in the north resulting in an AGM score of 0.0 due to the absence of historical trajectories. Nearest Neighbor imputation shows that the vessels take a more frequented path than linear interpolation. Figure 5.19c shows the geo-ellipse formed by our space-time interpolation method. The elliptical region encompasses more historical trajectories, generating a higher AGM score (0.41) and thus meriting further investigation by human analysts.



(a) No Interpolation

(b) Linear and k-NN Imputation

(c) Space Time Interpolation

Figure 5.19: Comparison of k-NN Imputation, Linear Interpolation and Space-Time Interpolation

5.8 Discussion

Signal Coverage Map: Each grid cell GC_i is classified as either reported or not reported by the signal coverage map based on whether its total location trace count exceeds a

threshold, θ_e. However, this threshold value θ_e is influenced by factors such as geographic region and the time frame for historical location traces. For example, accurately estimating signal coverage becomes difficult when a vessel enters a region with limited or no historical trace data, leading to a cold start problem. A signal coverage map is heavily dependent on the time frame range used to calculate it. For instance, a lower threshold value is needed if the time frame for global signal coverage is short, while a higher threshold value is required if the time frame is long. In this study, we only considered historical location traces based on the longest voyage time frame within a specific geographic area. Additionally, differences in geographic regions can affect the computation of the signal coverage map. For example, polar regions may have low signal coverage compared to more dense regions such as ports. Even within denser regions, signal coverage may vary due to factors such as weather or the designation of an area as a protected habitat.

Abnormal Gap Measure (AGM): AGM scores heavily depend on the geometric constraints of the geo-ellipse estimations within trajectory gaps. Similar to a classical space-time prism, a geo-ellipse is limited to the maximum speed achieved by the object within the duration of the trajectory gap. We did consider other measures for scoring anomalies that had to be abandoned because the data and ground truth information we needed was not available. However, they are worth considering:

- **Attributes:** Abnormality can also be defined with a combination of voyage-specific and physical attributes (e.g., speed, headings, draft). For instance, a draft is inversely proportional to speed since loading of cargo lower the ship's speed but raises the draught and vice-versa. Anomaly detection based on directional change (i.e., sudden change in movement direction) can also be extended toward trajectory gaps. However, post-processing human interpretation is required along with ground truth verification.
- **Temporal reasoning:** Including time of day as a factor can refine the proposed abnormality measure. For instance, an increased number of gaps during night-time may be an alarm signal to authorities. The addition of this factor would further enrich the score and open the way towards more sophisticated measures of abnormality based on emerging and vanishing hotspots of trajectory gaps.
- **Speed Distribution:** Monitoring a distribution based on an object's velocity profile

may result in multiple geo-ellipses, which more accurately quantify the continuous space within the geo-ellipse derived from the object's maximum speed. A physical interpretation based on dwell time and speed could be demonstrated by studying a vessel's temporal profile based on previous routes or voyages. For instance, since dwell time and speed are inversely related, a ship within a gap could spend more time maneuvering in the shortest path and move at a lower speed compared to a ship reaching the boundary region of a geo-ellipse at max speed.

Probabilistic Interpretation: Monitoring the distribution of location traces also helps define probabilistic interpretations of trajectory gaps. For instance, Gaussian processes may provide the assumption of availability of location trace based on the Gaussian distribution instead of the precise location. Other probabilistic interpretations within a geo-ellipse can further be considered as an average case in contrast to geo-ellipse (worst case) and shortest path (best case) interpretations.

Kernel Density Estimation: Kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable. KDE transforms point data into a continuous surface, estimating the density of points across the area. It is particularly useful in various fields like ecology, urban planning, and criminology to identify clusters or "hotspots" of activity or occurrences. Incorporating time as an additional dimension in kernel density estimation (KDE) enables the analysis of spatiotemporal data, revealing dynamic patterns, trends, and hotspots that change over both space and time. This approach involves preparing spatiotemporal data with precise timestamps, selecting appropriate spatial and temporal kernel functions, and optimizing bandwidths for both dimensions to accurately capture the scale of patterns.

5.9 Conclusion and Future Work

Given multiple trajectory gaps and a signal coverage map based on historical object activity, we find possible abnormal gaps, which are later sent to human analysts for ground-truth verification (inspired by the human-in-the-loop framework in [77]). The problem is important since it addresses societal applications such as improving maritime safety and regulatory enforcement (e.g., detecting illegal fishing, transshipments, etc.) and many

other application domains. Current methods rely on linear interpolation and our baseline algorithm Memo-AGD [70], which uses geo-ellipse estimation to capture abnormal gaps.

In this paper, we extended our previous work by proposing a space-time aware gap detection (STAGD) algorithm that uses space-time partitioning to index and filter out non-intersecting gaps effectively. In addition, we incorporate a dynamic region merge (DRM) to compute abnormal gap measures more efficiently while preserving correctness and completeness. The results show better computation time and accuracy for the proposed STAGD+DRM compared to the baseline linear interpolation and Memo-AGD.

Future Work: Computing abnormal gaps in trajectories is time-consuming and challenging. We are investigating new indexing methods that are more efficient than hierarchical indexing and linear search and better suited for modeling gaps in regional space. We intend to use these techniques in other application areas. Our plan also includes other methods (e.g., Bayesian estimation [73], regression-based methods [78]) and more accurate estimation of signal coverage density using kernel density estimation, while identifying hotspots [79] and colocations [80, 81, 82, 83] with time as an additional dimension. In addition, we aim to validate our evaluation metrics with real-world data and consider other external factors (e.g., device malfunction, hardware failure, etc.). We also aim to improve our measurement of unusual gaps by incorporating more physics-based parameters (e.g., draft) and exploring more expressive models (e.g., kinetic prisms). Finally, we plan to study ways to identify anomalies that are based on deception, such as fake trajectories or gaps generated by users who manipulate their location devices.

Chapter 6

Towards a Tighter Bound on Possible-Rendezvous Areas: Preliminary Results

6.1 Introduction

In Chapters 2 and 3, we investigated the problem of possible rendezvous detection in continuous Euclidean space by analyzing trajectory gaps characterized by prolonged durations, often spanning multiple days. These approaches relied on static historic trajectories and did not explicitly consider temporal variability, thereby treating time as an implicit dimension and overlooking periodic behavioral patterns. Consequently, rendezvous detection was primarily driven by spatial uncertainty, modeled using space-time prisms and geo-ellipses without accounting for time-dependent mobility dynamics. In this chapter, we extend the rendezvous detection problem to spatial networks, where temporal periodicity is a prominent characteristic, e.g., diurnal patterns, such as rush-hour traffic versus off-peak mobility. To address this, we introduce a time-slicing-based framework that explicitly models the temporal dimension by partitioning the trajectory gap interval into discrete time slices and constructing spatial sub-networks constrained by both temporal and spatial bounds. Each time slice yields a bounded region, referred to as a lens, that encapsulates an object's feasible movement at a given time instant. The intersection of such lenses

across multiple objects provides a more accurate approximation of the possible rendezvous region compared to traditional space-time prism-based methods. This temporally aware extension aligns with realistic mobility behavior in spatial networks and offers improved computational efficiency and solution quality.

Given trajectories with gaps (i.e., missing data), we investigate methods to tighten bounds on the spatial networks (e.g., road network, river network, etc.) for detecting potential rendezvous or meetup locations. Figure 6.1 (a) shows a pair of trajectory gaps on an underlying spatial network topology where a gap exists from $t = 2$ to $t = 6$ for both trajectory 1 (blue) and trajectory 2 (red) with a given object speed at 1 unit/second. Current approaches output possible meeting locations of two objects via the intersection of two space-time prisms. For instance, Figure 6.1 (b) shows six interpolated nodes (green) that qualify as rendezvous locations for both objects. Figure 6.1(c) shows the result after a time slicing method has reduced the number of nodes (by a factor of 3), filtering out nodes N_{10}, N_{17}, N_{12}, and N_{19} (yellow) and leaving N_{11} and N_{18} (green) as the output. The resultant interpolated nodes are then sent to human analysts for further investigation.

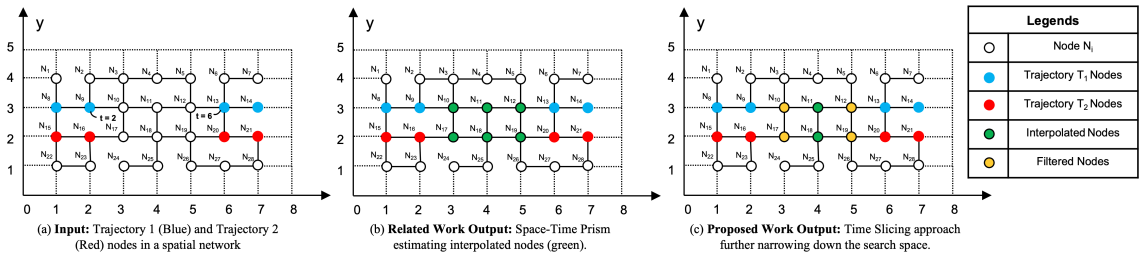


Figure 6.1: A illustration of Rendezvous Region Detection in Spatial Networks (Best in color)

Reducing the size of the space with possible rendezvous nodes is important for helping human analysts to detect and analyze trajectory data gaps. The smaller spatial area can be more effectively verified with ground truth via satellite imagery which can further aid early-stage decision-making. The problem has many societal applications related to homeland security, public health and safety etc. For instance, maritime safety involves monitoring activities such as illegal fishing and illegal oil transfers and transshipments [84]. Similarly, public health officials can analyze a given affected area where two objects could have met to control the potential spread of the disease. This chapter focuses on the specific

use case of improving public safety when two objects intentionally mask their movements to meet secretly within a trajectory gap area.

The problem is challenging since it is hard to model and interpret specific behavioral patterns, especially the rendezvous of two or more objects within a trajectory gap. Many methods rely on linear interpolation, which may lead to many missed patterns since moving objects do not always travel in a straight path. Methods based on the space-time prism are more geometrically accurate but identify large spatial regions, resulting in a time-intensive operation for the post-processing step performed by the human analyst. Further, such methods have a high computation cost due to large data volume. This work proposes computationally efficient time-slicing methods to effectively capture rendezvous regions with tighter spatial bounds in spatial networks.

The traditional literature [23, 85] on the mobility patterns of objects in spatial networks considers realistic scenarios and events (e.g., traffic congestion) and other behavioral patterns [24]. However, little attention has been given to movement patterns within trajectory gaps. Most works in this area are limited to linear interpolation methods using shortest path discovery [52, 86]. Other works consider an object's motion uncertainty via geometric-based methods (e.g., space-time prisms [16, 44, 45]) using spatial geo-ellipse boundaries constructed via motion parameters (e.g., speed). One recent work [87] does explicitly consider rendezvous or meetup queries in a spatial network, which the authors call *assembly queries*, but they used a loosely bounded geo-ellipse estimation from the space-time prism model. Our work proposes time-slicing methods to give tighter bounds on a given rendezvous region and provide further computational speedup using a dual convergence approach.

Contributions: The chapter contributions are as follows:

- We propose a time slicing model and theoretically show that it provides a tighter bound on possible rendezvous area relative to traditional space-time prisms.
- Using the time slicing model, we propose a Time Slicing-based Gap-Aware Rendezvous Detection (TGARD) algorithm to effectively detect rendezvous nodes and a Dual Convergence TGARD (DC-TGARD) algorithm using bi-directional pruning to improve the computational efficiency.
- We provide a theoretical evaluation for both algorithms based on correctness, completeness, and time complexity.

- We validate both algorithms experimentally based on solution quality and computation efficiency on both synthetic and real-world datasets.

Scope: The chapter proposes TGARD and DC-TGARD algorithms to tighten the spatial bounds of the rendezvous region in the spatial networks. We do not consider gaps with short time intervals (i.e., minutes, seconds etc.) or spatial areas that are low density or sparse. Kinetic prisms [26] fall outside the scope of the chapter. In addition, we do not model the rendezvous of the object's in trajectories without gaps. The proposed framework has multiple phases (i.e., filter, refinement, and calibration), but we limit this work to the filter phase. The refinement phase requires input from a human analyst and is not addressed here, and calibration of the cost model parameters is outside the scope.

Organization: The chapter is organized as follows: Section 6.2 introduces basic concepts, framework and the problem statement. Section 6.3 provides an overview of time slicing model. A broad and detailed literature survey is given in Section 6.5. Section 6.4 describes the baseline TGARD and refined DC-TGARD algorithms. Section 6.6 shows theoretical evaluations for proposed algorithms based on correctness, completeness and asymptotic complexity. Experimental evaluations are presented for both algorithms and related work are presented in Section 6.7. Finally, Section 6.8 concludes this work and briefly lists the future work.

6.2 Problem Formulation

6.2.1 Framework

We aim to identify possible rendezvous locations in a given set of trajectory gaps through a two-phase *Filter* and *Refine* approach. We introduce an intermediate time slicing filter to reduce number of interpolated nodes residing within the rendezvous region via proposed algorithms in a computationally efficient manner. The refinement phase further improves the solution quality so that human analysts may extract and analyze a comparatively fewer number of nodes involved in possible rendezvous by two (or more) objects. The inspection is further verified via satellite imagery to derive a possible hypothesis about the rendezvous activity (as shown in Figure 6.2).

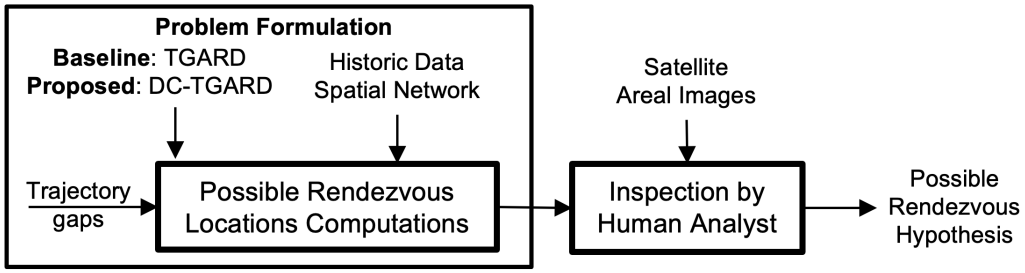


Figure 6.2: Framework for detecting possible rendezvous locations

6.2.2 Basic Concepts

A **spatial network** $G = (\mathcal{N}, \mathcal{E})$ consists of a node-set $\mathcal{N}$ and an edge set $\mathcal{E}$, where each element N $\in \mathcal{N}$ is a geo-referenced point, while each element $E \in \mathcal{E} \subseteq N \times N$ has an edge weight E_w i.e., the minimum time to travel from node N_i to node N_j. Figure 6.1 shows a spatial network where circles represent nodes (e.g., N_1) and the lines represent edges. A road system is an example of a spatial network where nodes are intersections, and edges are segments.

A **trajectory gap** (G_i) is a spatial interpolated region within a missing location signal time period or Effective Missing Period (EMP) above a certain threshold θ (e.g., 30 mins) between two consecutive points. Such interpolated region can be modeled as a geo-ellipse on the x-y plane, which spatially delimits the extent of a moving object's mobility given a maximum speed (MS) and effective missing signal period (EMP) (e.g., 30 mins etc) [16]. Figure 6.3 shows trajectory gap as spatial interpolated region in the form of a geo-ellipse (in blue) with C_1 and C_2 as focii and t_e-t_s (≥ 0) as EMP with MS_i as maximum speed.

A **possible rendezvous region** is defined as a set of overlapping nodes and edges when two trajectory gaps (i.e., spatially estimated in the form of geo-ellipse) involving different objects intersect with each other. Hence, such nodes can be defined as **possible rendezvous locations** for a set of objects within a trajectory gap. For instance, Figure 6.1 (b) shows interpolated nodes (in green) which may involved in a possible rendezvous location derived from the intersection of the two ellipses (as shown in Figure 6.4) using a space-time prism based method [87] from trajectory T_1 and T_2.

6.2.3 Problem Formulation

The problem to optimally identify a trajectory gap region in a spatio-temporal domain is formulated as follows:

Input:

1. A Spatial Network
2. A set of $|N|$ Trajectory Gaps
3. Historic Traffic Data.

Output: A more tightly bound Possible Rendezvous Region

Objective: Solution Quality and Computational Efficiency

Constraints: (1) Trajectories have long gaps (2) Maximum Acceleration is not available (3) Correctness and Completeness

Figure 6.1 (a) shows the input as a two-dimensional representation of a trajectory gap. Figure 6.1 (b) shows the output based on intersection of geo-ellipses resulting in interpolated nodes in green. Figure 6.1 (c) provides a more refined output resulting in a smaller number of interpolated nodes for human analysts to inspect.

6.3 Time Slicing Model

Our time slicing model uses a space-time prisms [16] to provide a detailed representation of an object's physical space. We first describe space-time prisms and then describe the time slicing model. Space-Time (ST) Prisms [16] are a collection of spatial points bounded by a physical space which is represented as an interpolated region where moving objects could have passed at a given maximum speed MS. Figure 6.3 shows the ellipse region in blue for a given time range [t_s,t_e] where t_s denote the start time and t_e denote the end time of the trajectory gap G_i. Equation 6.1 defines the geo-ellipse with focii (x_s, y_s) and (x_e, y_e) for a missing period t_e-t_s as follows:

$$\sqrt{(x - x_s)^2 + (y - y_s)^2} + \sqrt{(x - x_e)^2 + (y - y_e)^2} \leq (t_e - t_s) \times MS_1 \tag{6.1}$$

where, (x_s,y_s) and (x_e,y_e) are the start and end points of the trajectory gap with start time t_s and end time t_e (t_e ¿, t_s). The ellipse spatially delimits the extent of a moving object's mobility with maximum speed MS_1. Figure 6.4 shows the possible rendezvous region where we perform the spatial intersection of two ellipses $Ellipse_1 \in G_1$ and $Ellipse_2 \in G_2$ and time range is the calculated via inequality 6.2,

$$[t_s^{Ellipse_1}, t_e^{Ellipse_1}] \cap [t_s^{Ellipse_2}, t_e^{Ellipse_2}] \neq \emptyset \tag{6.2}$$

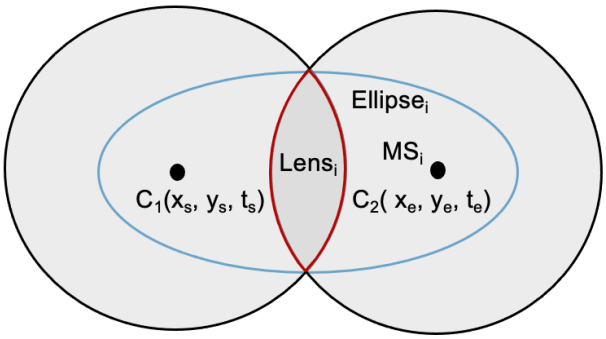


Figure 6.3: Time Slicing Model with Lens at time instant t

A **Time Slice** is an object's physical space sampled within a geo-ellipse (i.e., a ST-Prism) of a trajectory gap. A time slice bounds the spatial intersection of two circles C_1 and C_2 at a given time instant t where $t_s \leq$ t $\leq t_e$. Equations 6.3 and 6.4 represent the two circles generated from (x_s, y_s) and (x_e, y_e) at given time instant t (where $t_s \leq t \leq t_e$) as follows:

$$C_1 : (x_i - x_s)^2 + (y_i - y_s)^2 \leq (t - t_s)^2 MS_i^2 \tag{6.3}$$

$$C_2 : (x_i - x_e)^2 + (y_i - y_e)^2 \leq (t_e - t)^2 MS_i^2 \tag{6.4}$$

Figure 6.3 shows the spatial bounds of a time slice denoted as *Lens* which is defined as the geometry derived via $C_1 \cap C_2$ at a given time instant t. For instance, $Lens_i$ within an ellipse $Ellipse_i$ generated via maximum speed MS_i at time $t \in [t_s,t_e]$ i.e.,

$$t_s \leq t \leq t_e \tag{6.5}$$

$$0 \leq MS_i(t - t_s) \leq MS_i(t_e - t_s) \tag{6.6}$$

$$0 \leq MS_i(t_e - t) \leq MS_i(t_e - t_s) \tag{6.7}$$

Subtracting Inequality 6.6 and 6.7, we get inequality 6.8 which provides condition to define a lens $Lens_i$ as follows i.e., whether the radii intersection of $C_1 \cap C_2 \geq 0$.

$$MS_i(t - t_s) - MS_i(t_e - t) \geq 0 \tag{6.8}$$

The bounded region in Figure 6.3 shows $Lens_i$ i.e., intersection of $C_1 \cap C_2 \geq 0$. The bounded rendezvous regions at a given time instant t is shown in Figure 6.4 with intersection of $Lens_1$ and $Lens_2 \subseteq Ellipse_1 \cap Ellipse_2$.

A rendezvous region using a time slicing model is defined as the intersection of $Lens_i$ and $Lens_j$ derived from $Ellipse_1 \in G_1$ and $Ellipse_2 \in G_1$. Figure 6.4 further represents

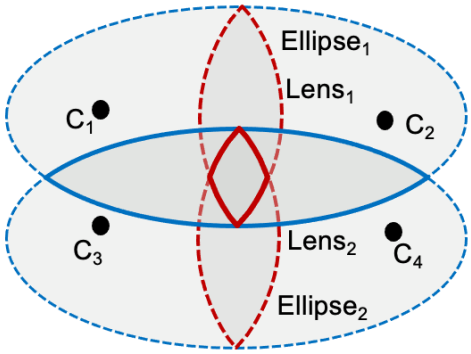


Figure 6.4: Time slicing model with lens intersection at time instant t

the rendezvous region via $Lens_1 \cap Lens_2$ at a given time instant t which provides an even tighter bound as compared to the $Ellipse_1 \cap Ellipse_2$. Lemma 6.3.1 and Lemma 6.3.2 provide formal proofs to the Theorem 6.3.3 which states that a time slice is a subset of the intersection of two space time prisms.

Lemma 6.3.1. Given a pair $< G_1, G_2 >$ of gaps in trajectories in an isometric euclidean space, the space-time prism model bounds the area of possible rendezvous by the intersection of two ellipses $Ellipse_1$ and $Ellipse_2$ where $Ellipse_1$ specifies the possible locations during gap G_1 given maximum speed MS_1 and $Ellipse_2$ specifies the possible locations during gap G_2 given maximum speed MS_2.

Proof. The proof is straightforward. Given a pair of intersecting trajectory gaps G_1 and G_2, any possible point $P \in \{Ellipse_1, Ellipse_2\}$ such that $Ellipse_1$ (P) ≥ 0 and $Ellipse_2$ (P) ≥ 0 must also satisfies inequality 6.2. Using the inequality 6.2, P spatially qualifies within possible rendezvous location bounded by Equation 6.11:

$$E_1(P) \geq 0 \; and \; E_2(P) \geq 0 \tag{6.9}$$

$$E_2(P) \geq 0 \tag{6.10}$$

assuming the time intervals of gaps G_1 and G_2 are $[t_s^{Ellipse_1}, t_e^{Ellipse_1}]$ and $[t_s^{Ellipse_2}, t_e^{Ellipse_2}]$ respectively

$$Ellipse_1 \cap Ellipse_2(P) \geq 0 \subseteq \{Ellipse_1 \; , \; Ellipse_2\} \tag{6.11}$$

where, object's $t \in [t_s^{Ellipse_1}, t_e^{Ellipse_1}] \cap [t_s^{Ellipse_2}, t_e^{Ellipse_2}] \subseteq [t_s, t_e]$.

Equation 6.11 shows a possible rendezvous area via intersection of $Ellipse_1$ and $Ellipse_2$ $\subseteq \{Ellipse_1, Ellipse_2\}$ where $Ellipse_1$ (P) ≥ 0 and $Ellipse_2$ (P) ≥ 0. Hence, the intersection

of space-time prisms bounds the area of possible rendezvous. □

Lemma 6.3.2. Given a pair $< G_1, G_2 >$ of gaps in trajectories in isometric euclidean space, the time slicing model bounds the instantaneous area of possible rendezvous at time t within the gap time interval by the intersection of $Lens_1$ and $Lens_2$ where $Lens_1$ specifies possible locations during gap G_1 given maximum speed MS_1 and Lens $Lens_2$ specifies possible locations during gap G_2 given maximum speed MS_2.

Proof. $Lens_1$ within Ellipse E_1 at time t within the time range $[t_s, t_e]$ must satisfy the circle intersection condition of whether circle C_1 and C_2 defined within Ellipse E_1. Inequality 6.5 defines the time range $[t_s, t_e]$ of a time instant t and Inequality 6.6 and 6.7 defined upper and lower spatial of the radius from t_s and t_e respectively.

$$t_s \leq t \leq t_e \tag{6.12}$$

$$0 \leq MS_1(t - t_s) \leq MS_1(t_e - t_s) \tag{6.13}$$

$$0 \leq MS_1(t_e - t) \leq MS_1(t_e - t_s) \tag{6.14}$$

The circle intersection condition $C_1 \cap C_2 \geq 0$ is already satisfied from Inequality 6.8. We can extend Inequality 6.15 and 6.16 for $Lens_1$ and $Lens_2$ respectively at a given time t:

$$MS_1(t - t_s) - MS_1(t_e - t) \geq 0 \tag{6.15}$$

$$MS_2(t - t_s) - MS_2(t_e - t) \geq 0 \tag{6.16}$$

Inequalities 6.15 and 6.15 qualifies any point P such that $Lens_1(\mathrm{P}) \geq 0$ and $Lens_2(\mathrm{P}) \geq 0$. Using Lemma 6.3.1, any point $P \in Lens_1 \cap Lens_2 \subseteq Ellipse_1 \cap Ellipse_2 \subseteq \{\ Ellipse_1\ ,\ Ellipse_2\ \}$ is bounded by instantaneous time t. □

Theorem 6.3.3. Given a pair $< G_1, G_2 >$ of trajectory gaps in isometric euclidean space, the possible rendezvous area (i.e., lens intersection) bounded at any instant during the gap time interval is a subset of the possible rendezvous area (i.e., ellipse intersection) bounded by the space-time prism model.

Proof. Given points (x_s, y_s, t_s) and (x_e, y_e, t_e) are focii of the ellipse E, According to Equation 6.1, a point (x,y) can lie anywhere in the geo-ellipse such that sum of distance from focii (x_s, y_s) and (x_e, y_e) $\leq$ (t_e-t_s). Equation 6.17 below derived from addition of Equation 6.3 and 6.4 of two circles also denotes a property of the ellipse.

$$(x_i - x_s)^2 + (y_i - y_s)^2 + (x_i - x_e)^2 + (y_i - y_e)^2 \leq 2(t_s - t_e)^2 \times MS_i^2 \tag{6.17}$$

The left hand side of Inequality 6.17 also denotes each $Lens_i$ is valid $\forall$ t $\in t_s \leq t_i \leq t_e$:

$$\bigcup_{t=t_s}^{t_e} Lens_t \subseteq \bigcup_{t=t_s}^{t_e} (t_s - t_e)^2 \times MS_i^2 \tag{6.18}$$

In addition, using Lemma 6.3.1, and 6.3.2, the areal bounds defined by $Lens_i$ will not exceed the bounds defined by the geo-ellipse. □

These lemmas and theorems can easily be generalized to spatial networks (e.g., road networks) by generalizing the ellipses and lenses to the subgraphs reachable during a gap time interval and the subgraphs reachable at a particular time instant within a gap time interval respectively, given a gap-start node, gap-end node, maximum speed and a gap time-interval. Due to lack of space, we are omitting the detailed proofs.

6.4 Proposed Approach

In this section, we first explain some underlying concepts which are later used to define the output of the problem (i.e., *possible rendezvous nodes*) in spatial networks. We then discuss the required pre-processing steps related to gathering candidate trajectory gap pairs which are later temporally sampled to construct spatial sub-networks. Finally, we present our proposed TGARD and DC-TGARD algorithms, which use time slicing properties for better solution quality and computational efficiency.

To calculate possible rendezvous locations, we first estimate the *availability interval* for each object and then calculate how early that object is able to reach (i.e., *earliest arrival time*) and the latest time the object can depart (*latest departure time*) at a given node. We formally define them as follows:

An **Earliest Arrival Time** (N_i^{EA}) is the minimum time object O_i takes from start node N_s to an intermediate node N_u i.e., $N_u^{EA} = N_u^{EA} + E_w$. A **Latest Departure Time** (N_i^{EA}) is the minimum time the object O_i takes from end node N_d to the intermediate node N_u where $N_u^{LD} = N_u^{LD}$ - E_w. An **availability interval (α)** is the time interval that object O_i wait at node N_u such that α(u) = [N_u^{EA},N_u^{LD}] $\neq \emptyset$, where EA_i(u) $\leq LD_i$(u). A node N_u is defined as *reachable* when $\alpha(u) \neq \emptyset$ and *not reachable* if α(u) = $\emptyset$ or $N_{EA}^u \leq N_{LD}^u$.

Figure 6.5 shows nodes N_0 to N_4 with intervals [0,4], [2,6], [3,7], [4,8] and [6,10] respectively. Nodes N_0 and N_4 are start and end nodes respectively and nodes [N_1,N_2,N_3] $\in N_u$. The earliest arrival is calculated by adding E_w to a given intermediate node N_u. For

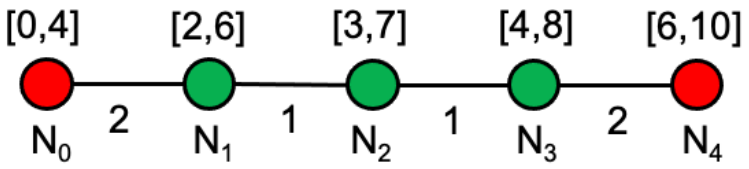


Figure 6.5: An illustration of Possible Rendezvous Nodes

instance, $N_1^{EA} = N_0^{EA} + E_w(N_0, N_1) = 0 + 2 = 2$. Similarly, the values of N_2^{EA}, N_3^{EA} and N_4^{EA} are 3, 4 and 6 respectively. In contrast, the latest departure is calculated by subtracting E_w to a given intermediate node N_u. For instance, $N_4^{LD} = N_5^{LD}$ - d(N_5, N_4) = 10 - 2 = 8. Similarly, the values of N_2^{LD}, N_3^{LD} and N_4^{LD} are 3, 4 and 6 respectively.

In the case of multiple paths Ps from a single source (i.e., N_s or N_e), shortest path algorithms (e.g., **Bi-directional Dijkstra's**) or other breadth first search approaches are considered. Hence, multiple arrival times and late departure times need to be considered for a given node N_i such that the earliest arrival time will be the **minimum** of all the arrival times derived via the shortest path from N_s to N_u (i.e., min($[N_u^{EA_1}$,$N_u^{EA_2}$,...,$N_u^{EA_k}]$)). In contrast, the latest departure time will be the **maximum** of all the departure times that result via the shortest paths from N_e to N_u (i.e., max($[N_u^{LD_1}$,$N_u^{LD_2}$,...,$N_u^{LD_k}]$)).

$$\alpha(N_u) = [min([N_u^{EA_1}, .., N_u^{EA_k}]), max([N_u^{LD_1}, .., N_u^{LD_k}])] \tag{6.19}$$

A **Possible Rendezvous Node (N_R):** is defined as a possible node N_u (or location) where two or more gaps belongs to different gaps could have physically met. For instance, the availability intervals of gaps G_i and G_j at N_u are defined as $\alpha(N_u, G_i)$ and $\alpha(N_u, G_j)$ respectively. The rendezvous is feasible only if :

$$\alpha(N_u, G_i) \cap \alpha(N_u, G_j) \geq TO \tag{6.20}$$

where TO ($\neq \emptyset$) is defined as **Time Overlap threshold**.

If the above conditions are satisfied, the nodes are then the nodes are forwarded to human analysts for manual inspection.

6.4.1 Constructing Spatial Sub-networks

Section 6.3 describes how time slicing in an unconstrained (i.e., euclidean) space narrows down the search space through the construction of a lens at given time instant t to provide tighter bounds as compared to entire geo-ellipse region.

In spatial networks, we restrict such unconstrained movements which closely resembles an vehicle's mobility in a road network. However, we use the combination of geo-ellipses and lenses to geometrically restrict the object movement capabilities in a Manhattan search space, thereby providing an upper bound towards the mobility. In addition to tightening the spatial bounds, time slicing also capture the time-dependent properties of spatial networks which more accurately model an object's mobility in real-world scenarios.

Algorithm 8 first gathers trajectory gap pairs based on their temporal properties via a *Filter* and *Refine* approach to minimize the combinatorial computations. Then we temporally sample each trajectory gaps which captures a time-dependent edge which can later affects overall object's mobility.

Extract Trajectory Gap Pairs: For the effective gap collection, we use a *plane-sweep approach* to sort the trajectory gaps G_i and then filter out gaps that are not involved in a rendezvous. First, we sort all gaps $G_i \in [t_s,t_e]$ based on the start time t_s of the coordinates of G_i. Then, we filter gap pairs by checking the *necessary* condition i.e., whether their respective time ranges overlap. For instance, Gap G_i and G_j should satisfy their respective time ranges $[t_s^{G_i},t_e^{G_i}]$ and $[t_s^{G_j},t_e^{G_j}] \neq \emptyset$ (using Equation 6.2). For the sufficient condition, we check if G_i and G_j are spatially intersecting such that $Ellipse_i \cap Ellipse_j \neq \emptyset$. If yes, then we save the resultant shape of $Ellipse_i \cap Ellipse_j$ and time range $[t_s^{G_i},t_e^{G_i}] \cap [t_s^{G_j},t_e^{G_j}]$ $= [t_s^R,t_e^R]$ for creating spatial sub-networks.

Creating Spatial Sub-Networks: Here, we generate spatial sub-networks based on the qualifying nodes within the spatial (i.e., $Ellipse_1 \cap Ellipse_2$) and temporal $[t_s^R,t_e^R]$ constraints of the trajectory gap pairs $< G_1,G_2 >$ as described in Section 6.4.1. To create a spatial network (SN), we perform a linear scan of (N) nodes and edges $(E) \in Ellipse_1 \cap Ellipse_2$ and to calculate edge weight E_w by considering historic location traces residing within the time range of $[t_s^R,t_e^R]$ and calculate the time cost by taking average speed μ(MS) and dividing it by total edge distance between N_i and N_j. During time slicing, we capture the dynamic edge weights within $[t_s^R,t_e^R]$. First perform uniform sampling for a given temporal range into N samples where $[t_i^R, t_{i+1}^R] \in [t_s^R,t_e^R]$. We then compute edge weights E_w^i with given time range $[t_i^R, t_{i+1}^R]$ and filter nodes which are qualified within the $Lens_i$ where $N_i \subseteq N$. Hence for each time frame we generate subnetworks SN_i with nodes N_u^i edge weights E_w^i $\forall$ i $\in [t_i^R, t_{i+1}^R]$. Algorithm 8 summarizes the process of gathering

trajectory gap pairs and creation of spatial networks in steps as follows:

Algorithm 8 Spatial Sub-Networks with Rendezvous Gap Pairs

Input:
Historic Trajectory Data (HTD)
A set of Trajectory Gaps [G_1,...,G_n]
A Spatial Network N
A Sampling Rate K
Output:
Spatial Sub-Networks List

procedure :
  Sub-Networks Map $\leftarrow \emptyset$
  for each: $G_i \in$ Non-Observed List **do**:
    for each: $G_j \in$ Observed list if Observed list $\neq \emptyset$ **do**
      if $[t_s^{G_i},t_e^{G_i}] \cap [t_s^{G_j},t_e^{G_j}]$ and $G_i \cap G_j \neq \emptyset$ **then**:
        $[SN_0, SN_1, .., SN_k] \leftarrow$ Subnetworks ($G_i \cap G_j$, HTD, N, K)
        Sub-Networks Map [¡G_i, G_j¿] $\leftarrow [SN_0, SN_1, .., SN_k]$
      end if
    end for
  end for
  return Sub-Networks Map
end procedure

Step 1: For a given trajectory gap G_i in a Non-Observed list, we first check if G_i intersects with G_j in the Observed List $\notin \emptyset$. In case Observed List $\in \emptyset$, we add $G_i \in$ Observed List and move to G_{i+1} in the Non-Observed List. If G_j present in the Observed List, we check if the Ellipse bounds of $G_i \cap G_j$ and $[t_s^{G_i},t_e^{G_i}] \cap [t_s^{G_j},t_e^{G_j}] \neq \emptyset$. If not, then we derive intersected temporal range $[t_s^R, t_e^R] \in [t_s^{G_i},t_e^{G_i}] \cap [t_s^{G_j},t_e^{G_j}]$ and the resultant geo-ellipse intersection boundary of $(G_i \cap G_j)$ for unconstrained rendezvous study area which is later projected on a the given spatial network N to extract sub-network SN for gap pairs $< G_i,G_j >$ where $SN \subseteq N$.

Step 2: For a given trajectory gap pair $< G_i,G_j >$ and sampling rate K, we sample time interval $[t_s^R, t_e^R]$ in K samples such that each sub-network SN_i,SN_{i+1},...,SN_k has it's corresponding time-stamp $[t_i^R, t_{i+1}^R,...,t_{k-1}^R, t_k^R]$. Using historic trajectory data HTD, we calculate edge weights $E_w \; \forall \; SN$ where each $SN_i \in \{ E_w, t_k^R \}$ is then saved in Sub-Network Map which is later returned as the output.

6.4.2 Time Slicing Gap-Aware Rendezvous Detection Algorithm (TGARD)

The proposed TGARD algorithm captures an object's more realistic movements by considering parameters such as traffic congestion, which later affect the shortest path computation needed to calculate Earliest Arrival and Latest Departure time of the given sampled sub-network SN_k where each sub-network $SN_k \in [t_k^R]$ and $Lens_k$. Intermediate nodes N_u is defined within $Lens_k^{G_i} \cap Lens_k^{G_j}$ towards which shortest path computation form start and end nodes is computed to calculate N_u^{EA} and N_u^{LD}. However, we need to perform shortest path computation for every N_u of every sampled sub-network SN_k and edge weights E_w^k. Figure 6.6 is a time aggregated graph representation [88] where each edge represents time series edge weights $[E_w^{t_1}, E_w^{t_2}]$, and $E_w^{t_1} \in P_1$ and $E_w^{t_2} \in P_2$, which gives the shortest path from N_0 to N_6.

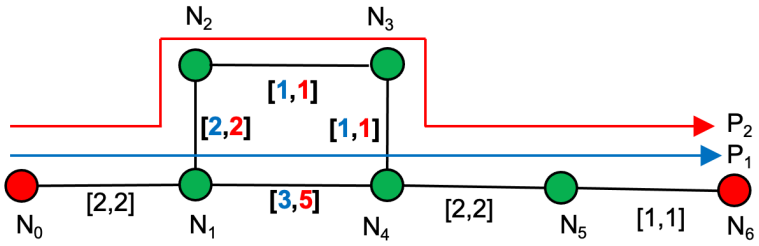


Figure 6.6: An illustration of Time Dependent Shortest Path

In Figure 6.6, P_1 takes the path N_0, N_1, N_4, N_5, and N_6 at t_0 (since 2+1+1 $\geq$ 3). In contrast, path P_2 uses N_0, N_1, N_2, N_3, N_4, N_5 and N_6 since the path from N_1 and N_2 at t_1 is 5 $\geq$ 2+1+1 (i.e., N_1, N_2, N_3 and N_4). In this case, node N_4 gets affected and the shortest path needs to be recomputed towards N_4 at time t_2. To capture such changes in edge weights in $E_w^i \in t_i$ and $E_w^{i+1} \in t_{i+1}$, we use a difference metric $\delta(E_w^{i,i+1})$ and compare it with a threshold τ (i.e., whether $\delta(E_w^{i,i+1}) \geq \tau$. If it is, then we recompute the shortest path towards that node. In Algorithm 9, we denote $\delta(E_w^{i,i+1})$ as $\delta(E_w^k)$. For instance, in Figure 6.6, node N_4 gets affected at time $t = 2$ given $\tau = 2$ since $\delta(E_w^{i,i+1}) \geq \tau$. Hence we recompute the shortest path at $P2$ from N_0 to N_6 via N_2 and N_3. If not, the current path remains the same for future computations and thereby retains it's optimal sub-structure property for shortest path, stated as follows:

Theorem 6.4.1. Given $G = (V, E)$ with edge weights E_w. Let $P = [N_1, N_2, N_3, ..., N_k]$ be the shortest path from N_1 to N_k such that $1 \leq \text{i} \leq \text{j} \leq \text{k}$. If $P_{ij} = [N_i, N_{i+1}, ..., N_j]$ is the sub-path of P from node N_i to N_j. Then, P_{ij} is the shortest path from P_i and P_j.

Proof. The proof is straightforwards. If we decompose P into $P_{1,j} \rightarrow P_{i,j} \rightarrow P_{j,k}$, where $E_w(P) = E_w(P_{1,j}) + E_w(P_{i,j}) + E_w(P_{j,k})$ and assume a new path $P_{ij}^{'}$ from N_i to N_j such that $E_w(P_{ij}^{'}) \leq E_w(P_{ij})$. Then, $E_w(P) \leq E_w(P_{1,i}) + E_w(P_{i,j}^{'}) + E_w(P_{j,k})$ which contradicts P as shortest path assumption from N_1 to N_k. □

Algorithm 9 Time Slicing Gap-Aware Rendezvous Detection

Input:
Time Overlap Threshold TO
Rest same as Algorithm 1
Output:
Possible Rendezvous Nodes Hmap N_R

procedure :
  Rendezvous Node Map RNMap $\leftarrow \emptyset$
  Use Algorithm 1 to construct Subnetwork Map SN Map
  for each: Sub Network $SN_k \in$ Trajectory Gap Pairs G_i, G_j **do**
    Construct $Lens_k^{G_i}$ and $Lens_k^{G_j}$ for $t_k^R \in [t_s^R, t_e^R]$
    if $Lens_k^{G_i} \cap Lens_k^{G_j} \neq \emptyset$ and $\delta(E_w^k) \leq \tau$ **then**:
      $N_u \leftarrow Lens_k^{G_i} \cap Lens_k^{G_j}$
      if N_u^k is reachable for Sub Network SN_k and $\notin$ RNMap **then**:
        Compute N_u^{EA} and N_u^{LD} for both $\alpha(N_u^{G_i})$ and $\alpha(N_u^{G_j})$
        if $\alpha(N_u^{G_i}) \cap \alpha(N_u^{G_j}) \neq \emptyset$ and $\geq TO$ **then**:
          Rendezvous Node Map RNMap$[N_R] \leftarrow N_u \; \forall \; t_k^R$
        end if
      end if
    end if
  end forreturn Rendezvous Node Map RNMap
end procedure

The TGARD algorithm steps are as follows:

Step 1: First we compute a sub-network (SN) map for each gap pair $< G_i, G_j >$ using Algorithm 8 and initialize a Rendezvous Node Map RNMap $\leftarrow \emptyset$. For each gap pair $< G_i, G_j >$, we have k sub-networks SN_k which generate $Lens_k$ from both G_i and G_j at t_k^R. We then filter out N_u^k residing within the spatial boundary of $Lens_k^{G_i} \cap Lens_k^{G_j}$, where $Lens_k^{G_i} \cap Lens_k^{G_j} \neq \emptyset$. In addition, we also check if edge weights $\delta(E_w^k)$ for sub-network SN_k

have changed considerably by comparing them with threshold τ. If yes, then we consider all the affected nodes within $Lens_k^{G_i} \cap Lens_k^{G_j}$ for computing *early arrival* and *late departure* times to preserve completeness. If not, we re-use the previous shortest path calculations for *early arrival* and *late departure* at t_{k-1}^R adhering to the optimal substructure property for a shortest path stated in Theorem 6.4.1.

Step 2: After gathering all $N_u \in Lens_k^{G_i} \cap Lens_k^{G_j}$, we check if N_u is reachable (i.e., $\alpha(N_u) \neq \emptyset$) by checking the availability interval $[N_u^{EA},N_u^{LD}] \not\geq [t_s,t_e]$ for each gap G_i and G_j. If valid, we then check whether both the availability intervals of G_i and G_j and their intersection is $\neq \emptyset$ and $\geq TO$. If yes, then N_u^k qualifies as N_R and is saved in a rendezvous node (RN) Map which is later returned as output.

6.4.3 A Dual Convergence Approach (DC-TGARD)

Since the time slicing operation is computationally expensive, the Dual Convergence method uses the symmetric property of the ellipse to reduce the iterations of time slicing used by *TGARD* while preserving correctness and completeness. We first define an early termination condition based on areal coverage of $Lens_k$, where $0 \leq k \leq n$ for a given time frame t with $0 \leq t \leq n$. We then perform *bi-directional pruning* using the ellipse symmetric property and compute EA and LD from both ends of the ellipse in parallel to improve computational efficiency.

Ellipse Symmetry Property: Given an ellipse with major and minor axes and center (0,0), its foci (±c,0) are equidistant from its origin at c. Hence, the areal coverage drawn from the lenses centered at (±c,0) will be the same, permitting an equal number of nodes within the spatial bounds of the lenses. For instance, Figure 6.7 (b) shows an equal number of nodes residing within the spatial bounds of $Lens_k$ and $Lens_{n-k}$. Using Lemma 6.3.3, time slicing does not leave out any other interpolated nodes (green) bounded within the ellipse, preserving the completeness of DC-TGARD.

Early Stopping Criteria: The early stopping criteria holds when the property of monotonicity (i.e., non-increasing) is violated while estimating the areal coverage of lenses throughout every time slicing operation from t_s to t_e. For instance, Figure 6.7 (a), shows the increasing areal-coverage i.e., $\mathrm{A}(Lens_k) \leq \mathrm{A}(Lens_{k+1})$ (given $\mathrm{A}(Lens_k)$ as the areal coverage of lens k) at t_k and t_{k+1} respectively (where, $t_{k+1} \geq t_k$). This is due to the increasing (or non-decreasing) length of the minor axis which preserves the monotonicity

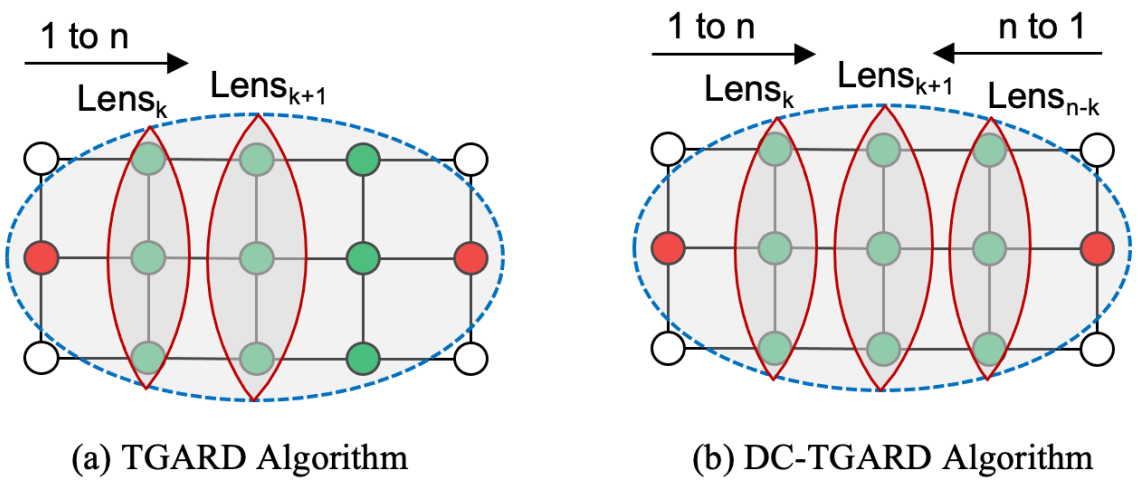


Figure 6.7: An illustration of TGARD vs Dual Convergence Method

until the $Lens_{k+1}$ of the ellipse. After $Lens_{k+1}$, the areal coverage start decreasing and thereby violating the monotonic property i.e., (A($Lens_k$)≤A($Lens_{k+1}$)≥A($Lens_{i+2}$)) and, resulting in early stopping of the time slicing operation.

Dual Convergence Approach: We need to consider each time-slice operation to preserve completeness of the algorithm. The dual convergence approach introduces **bi-directional pruning** in conjunction to efficiently converge towards *early stopping criteria*. Hence, a time-slicing operation is performed from both tail ends of the ellipse focii in parallel such that both lenses generated at t and $t_e - t$ follows the property of **ellipse symmetry**. While performing bi-directional pruning, we check the *early stopping criteria* if A($Lens_i$) ≥ A($Lens_{i+1}$) such that the monotonic property is violated. Figure 6.7 shows an example of dual convergence where baseline TGARD in Figure 6.7 (a) performs a linear time slicing operation from $Lens_k$, $Lens_{k+1}$ to $Lens_n$ covering all the interpolated nodes. Figure 6.7 (b) shows bi-directional pruning at $Lens_k$ and $Lens_{n-k}$ where k ∈ [1,n]. The areal coverage of $Lens_k$ i.e., A($Lens_k$) and A($Lens_{n-k}$) using the property of ellipse symmetry computed in parallel and further checking if A($Lens_k$) ≥ A($Lens_{k+1}$) which gives an early stopping condition. For instance, in Figure 6.7 DC-TGARD terminates when A($Lens_{k+1}$) ≥ A($Lens_{n-k}$) at t=2 as compared to TGARD in Figure 6.7 (a) which terminates at t=3. Hence, the areal coverage pruning is more efficient in DC-TGARD as compared to TGARD. Hence the pruning is similar to to a *bitonic array* where the peak areal coverage using

MaxOverlap variable.

Algorithm 10 Dual Convergence Time Slicing Gap-Aware Rendezvous Detection (DC-TGARD)

Input:
Same as Algorithm 2
Output:
Possible Rendezvous Nodes List N_R

procedure :
  Use Algorithm 8 to construct Subnetwork Map SN Map
  Rendezvous Node Map RNMap $\leftarrow \emptyset$ and Max Overlap $\leftarrow \emptyset$
  Construct Sub Network SN_k and SN_{n-k} for t_k^R and t_{n-k}^R respectively
  for each: Sub Network SN_k and $SN_{n-k} \in$ Trajectory Gap Pairs ¡G_i, G_j¿ **do**
    Construct $Lens_k^{G_i}$, $Lens_k^{G_j} \in t_k^R$ and $Lens_{n-k}^{G_i}$, $Lens_{n-k}^{G_j} \in t_{n-k}^R$
    while Max Overlap $\not\geq$ Area($Lens_k^{G_i} \cap Lens_k^{G_j}$) **do**
      if $Lens_k^{G_i} \cap Lens_k^{G_j} \neq \emptyset$ and $\delta(E_w^k, E_w^{n-k}) \leq \tau$ **then**
        $N_u^k \leftarrow Lens_k^{G_i} \cap Lens_k^{G_i}$ and $N_u^{n-k} \leftarrow Lens_{n-k}^{G_i} \cap Lens_{n-k}^{G_j}$
        if $N_u^k \in SN_k$ and $N_u^{n-k} \in SN_{n-k}$ reachable and $\notin$ RNMap **then**:
          Compute N_u^{EA}, N_u^{LD} $\forall$ t_k^R and $t_{n-k}^R \in \alpha(N_u^{G_i})$ and $\alpha(N_u^{G_j})$
          if $\alpha(N_u^{G_i})$ and $\alpha(N_u^{G_j}) \neq \emptyset$ and $\geq TO$ $\forall$ t_k^R and t_{n-k}^R **then**:
            Rendezvous Node Map RNMap[N_R] $\leftarrow N_u^k$ and N_u^{n-k}
          end if
        end if
      end if
    end while
  end for
  return Rendezvous Node Map RNMap
end procedure

The DC-TGARD algorithm steps are as follows:

Step 1: After generating Sub-Network Map (SN Map) via Algorithm 8, we first initialize Rendezvous Node Map RNMap and Max Overlap variable as $\emptyset$. We then generate sub-networks SN_k and SN_{n-k} for each t_k^R and t_{n-k}^R and generate $Lens_k$ and $Lens_{n-k} \subseteq SN_k$ and SN_{n-k} respectively. We then check if $Lens_k^{G_i}, Lens_{n-k}^{G_i} \cap Lens_k^{G_j}, Lens_{n-k}^{G_j} \leftarrow \emptyset$ and simultaneously check $\delta(E_w^k, E_w^{n-k})$ with threshold τ. Rest of the steps are similar to TGARD instead we simultaneously calculate every variables for early arrival, late departures and availability intervals along with their respective conditions.

Step 2: We update the Max Overlap variable with Area of $Lens_k^{G_i} \cap Lens_k^{G_j}$ or $(Lens_{n-k}^{G_i} \cap Lens_{n-k}^{G_j})$. Finally, we check the early termination condition (i.e., current

$Lens_k^{G_i} \cap Lens_k^{G_j}$ and $Lens_{n-k}^{G_i} \cap Lens_{n-k}^{G_j} \geq$ Max Overlap. If yes, then the loop has reaches to the *peak element* and algorithm terminates.

6.5 Other Related Work

The literature of trajectory mining includes a broad overview of movement patterns [24] and a taxonomy of spatial mining methods used in various application domains [23]. In trajectory data management, specific frameworks have analyzed trajectory gaps via indexing methods (e.g., hierarchical trees [89, 90], grids [91] but they are not designed to detect movement patterns. Other works have modelled regions of uncertainty [92, 93] via snapshot models etc. More realistic solutions are based on geometric models such as cylindrical [47] and space-time prism models [16, 45, 40] that construct an areal interpolation of the gaps using coordinates and a maximum speed of the objects. More recently, a kinetic prism [26] approach showed improved estimation by considering other physical parameters (e.g., acceleration). However, no studies have addressed rendezvous behavior patterns within trajectory gaps.

In spatial network research, many methods consider [94, 95, 86] uncertainty in trajectory gaps by a map matching the potential routes taken by a moving object in a road network. Many studies are based on deterministic [96, 97] and probabilistic [98, 99] methods that consider historic trajectories to find potential routes in a road network topology. However, none of these works consider rendezvous behavioral patterns in spatial networks. Recent work [87], studies rendezvous detection in the *static-spatial network* via space-time prisms based on objects and duration using type-2 uncertainty (geo-ellipse) resulting rather loose bounds. In this work, we are identifying rendezvous nodes via a time-slicing method[76] which tighten the bounds of rendezvous areas. We further consider dynamic edge weights [100, 85, 101] for better solution quality.

6.6 Theoretical Evaluation

6.6.1 Correctness and Completeness

In this section, we provide a theoretical analysis of the correctness and completeness of the proposed algorithms.

Lemma 6.6.1. TGARD and DC-TGARD algorithms are correct.

Proof. Given a finite set of $|K|$ trajectories with maximum N number of finite points resulting in a maximum bound of $|K| \times |N|$ finite trajectory points. Hence, finite number of trajectory gaps G_i generated while performing pre-processing for finite trajectory gap-pair generation resulting a finite number of operations will be performed by *TGARD and DC-TGARD* algorithms to terminate at a finite time. The correctness of the algorithm also depends upon the extent of the overlap of the two availability intervals $\alpha(1) \in G_1$ and $\alpha(2) \in G_2$. For instance, a low overlap threshold results in multiple false positive rendezvous whereas a high overlap threshold results in more false negatives. Hence, both *TGARD and DC-TGARD* algorithms are correct for a given overlap threshold. □

Lemma 6.6.2. TGARD and DC-TGARD algorithms are complete.

Proof. The proposed *TGARD and DC-TGARD* algorithms consider all the nodes within the spatial bounds of the lenes and their respective geo-ellipses. Theorem 6.3.3 proved that a time slice lens is a subset of geo-ellipse and does not exceed it's limit. This ensures that each node within gap is participating during time slicing operation. Hence, both *TGARD and DC-TGARD* algorithms are complete. □

6.6.2 Asymptotic Analysis

The time complexity of the comparison operations for each algorithm are as follows:

Exacting Candidate Gap Pairs: For generating candidate pairs, $\binom{|K|}{2}$ trajectories must be selected where each such pair will have $|N-1| \times |N-1|$ comparisons. Hence, the total number of comparisons will be $\binom{|K|}{2} \times |N-1| \times |N-1|$, which results in an asymptotic worst case of $O(|K|^2 \times |N|^2)$.

Creating Sub-Networks (SNs): For a given subgraph SN with $|E| \subseteq N \times N$ edges, the edge weights needs to be calculated $|T|$ times where E_w is calculated at constant time. Hence the overall time complexity for calculating E_w is O $(|E| \times |T|)$.

TGARD Algorithm: Given a sub-network SN for N nodes, we perform a linear scan for each time slice $|T|$ to gather intermediate node N_u where $N_u \leq N$ resulting in $O(|K| \times |T|)$ operations. For computing Early Arrival and Late Departure for N_i, we run a *bi-directional Dijkstra's* algorithm with $O(|N_i| + |E|)$. Hence, the algorithm perform $O(|K| \times |T|) + O(|N_u| \times |E|)$ operations.

DC-TGARD Algorithm: The worst case of DC-TGARD will be similar to *TGARD* except we perform bi-directional searches while computing the time slice operations and the early termination condition guarantees the algorithm stops in $T/2$ steps which reduces shortest path computations, which enhances computational efficiency in high density regions.

6.7 Experimental Evaluation

The goal of the experiments was to validate the benefit of the proposed time slicing approach for reducing the search space of rendezvous detection. We evaluated the solution quality by comparing the proposed DC-TGARD against space-time prism methods [87]. We also compared the execution time of TGARD and DC-TGARD under different parameters. Details shown in Fig 6.8.

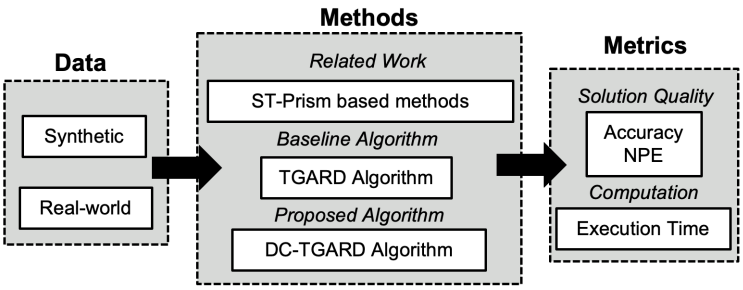
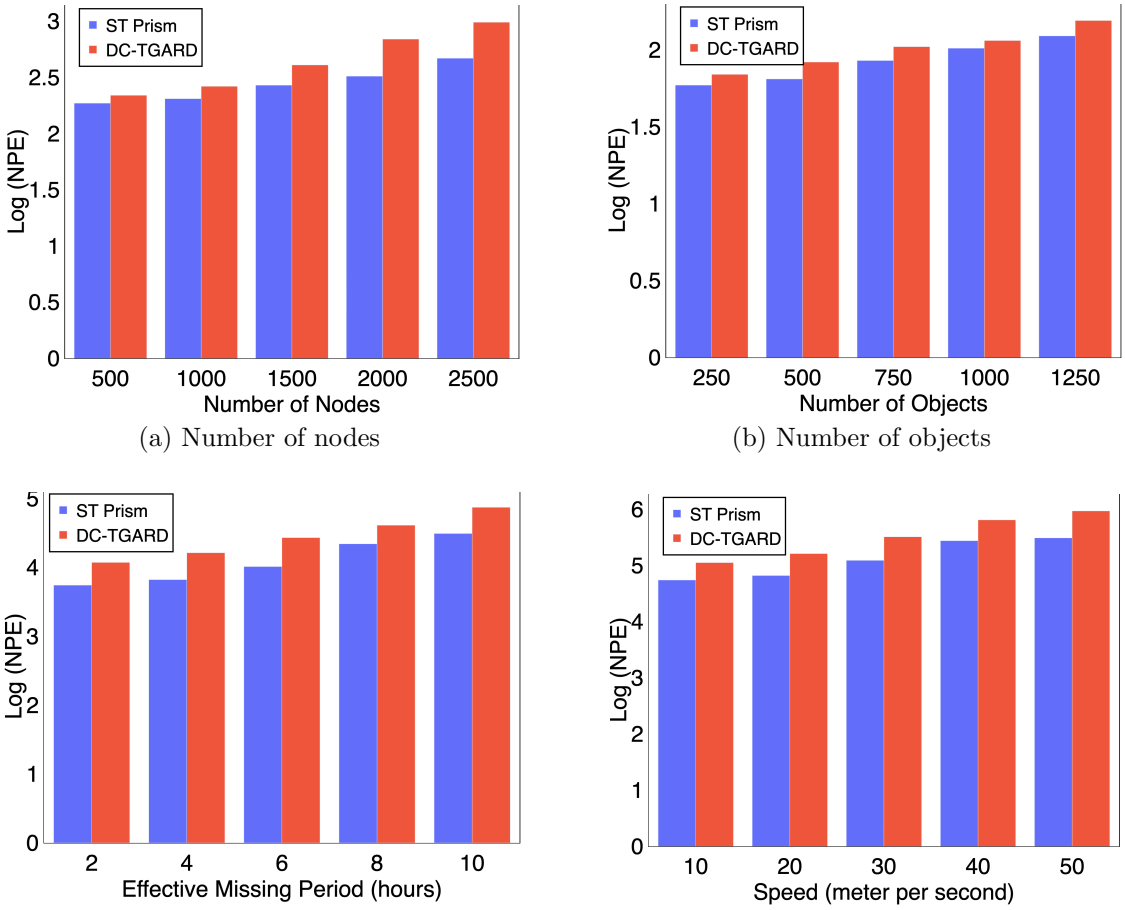


Figure 6.8: Experiment Design

Real World Data: We used Geolife [102] dataset based on Beijing Road Networks where each location has latitude, longitude, and height with a variety of travel modes (e.g., walking, driving, etc.). In this chapter, we limited our evaluation to driving patterns due to their accordance with road network topology (e.g., road segment and intersection). In addition, we further simulated certain trajectories by adding more objects and case scenarios of rendezvous patterns to test the effectiveness and scalability of the proposed methods.

Synthetic Data Generation: For *solution quality* experiment, we lacked ground truth data (i.e., information on whether a node was involved in a rendezvous or not). Therefore, we evaluated the proposed algorithms on synthetic data derived from the Geolife dataset. First, we gather trajectories on a fixed study area of the Beijing road network with mobility data. We then pre-processed the trajectory points with gap durations greater than 30 mins and randomly classified each gap based on whether N_r was reachable or not.



(a) Number of nodes (b) Number of objects

(c) Effective missing period (d) Speed

Figure 6.9: DC-TGARD vs. a space–time-prism method under different parameters

Computing Resources: We performed our experiments on a system with a 2.6 GHz 6-Core Intel Core i7 processor and 16 GB 2667 MHz DDR4 RAM.

6.7.1 Solution Quality

To assess solution quality, we considered bounding efficiency and accuracy. We developed an efficiency metric called node pruning efficiency (NPE), to measure the tightness of the proposed filter in the trajectory gap pairs. We defined NPE as the ratio of the nodes of a total study area to the nodes within a bounded region (Eq. 6.21):

$$NPE = \frac{Nodes\ in\ Total\ Study\ Area}{Nodes\ in\ the\ Bounded\ Region} \tag{6.21}$$

Nodes Pruning Efficiency (NPE): We fixed the study area and varied the Nodes

(N), Effective Missing Period (EMP), Speed (MS) and Number of Objects (O) within each trajectory gap G_i. Figure 6.9 shows the results for the space-time prism approach and the proposed time slicing approach. As can be seen in Figure 6.9a, DC-TGARD has better node pruning efficiency (NPE) as we increase the number of nodes (since node density increases). A similar trend is seen in Figure 6.9b except the effectiveness of time slicing is not as significant as we increase the number of objects of a fixed study area. Figure 6.9d and 6.9c again show the time slicing effectiveness is superior as compared to baseline approach as we increase the speed and duration of the effective missing period (EMP) of gap.

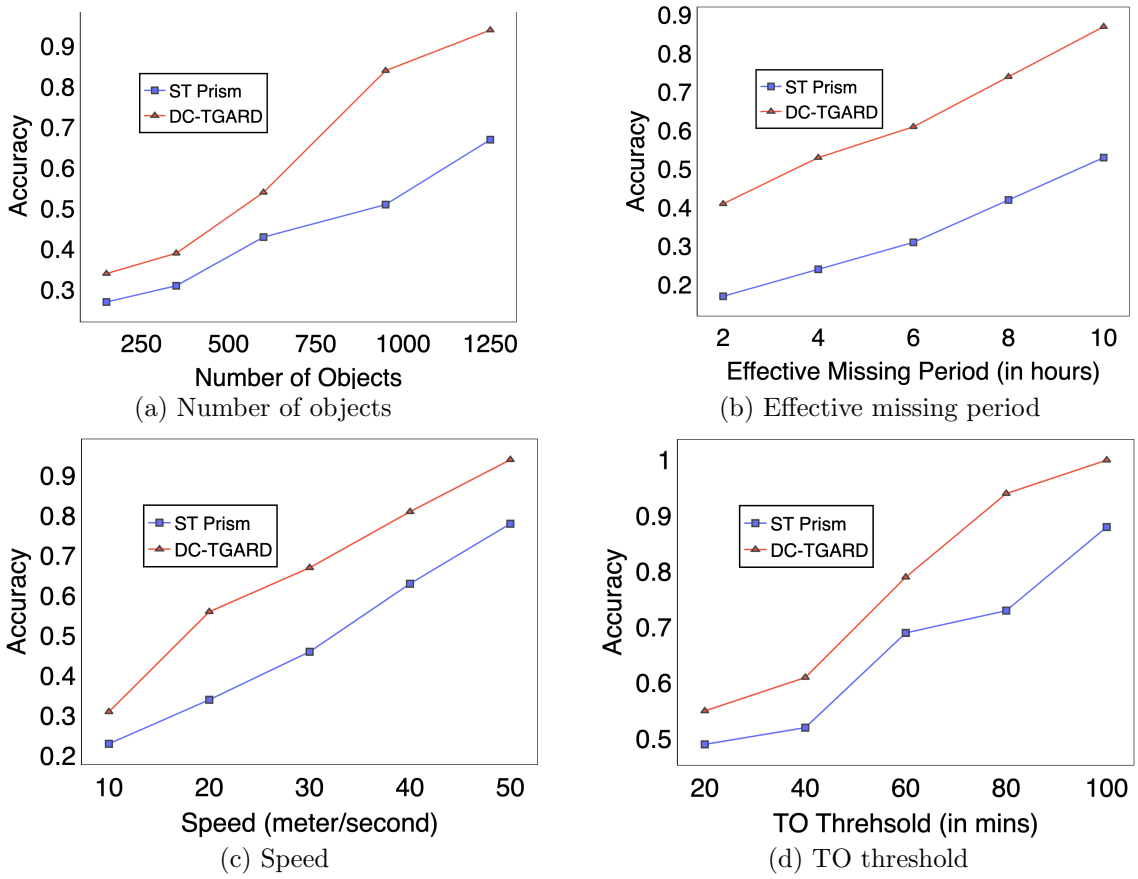
Accuracy: We used synthetic data with manually labeled ground truth data about whether or not the objects were involved in rendezvous for a fixed number of objects (i.e., 2000) within a fixed size network of 5000 nodes. We then varied different parameters including Time Overlap Threshold (TO).

(1) Number of Objects: We set the EMP to 4 hours, MS to 30 m/s, Time Overlap Threshold (T) to 30 mins and we varied the number of objects from 500 to 1250. Figure 6.10a shows DC-TGARD gives a more accurate representation of the minimum travel time from the start node and end node of the trajectory gap compared to space-time prism based methods. This results in more accurate estimates of N_r and reduces the number of false negatives, which proved to be quite common in the space-time prism based approaches [87].

(2) Effective Missing Period (EMP): Next, we set the number of objects at 500, the number of nodes of $5x10^4$, MS at 30 m/s, TO threshold to 30 mins, and varied the EMP threshold from 2 to 10 hours. As shown in Figure 6.10b, we find that DC-TGARD outperforms space time prisms. Higher EMP estimates increase the size of the geo-ellipses resulting in higher false positives.

(3) Speed: We kept the number of objects and EMP constant at 1000 and 4 hours respectively, the TO threshold at 30 mins and increased the maximum speed MS from 10 to 50 in $meter/second$ units. Figure 6.10c) shows DC-TGARD is more accurate as compared to ST-Prisms. The reason is that high speed outputs a greater number of potential rendezvous. This increases false positives on both algorithms but DC-TGARD filters out *non-reachable* nodes, thereby increasing accuracy.

(4) Time Overlap Threshold (TO): We set EMP at 4 hours, MS at 30 m/s and



(a) Number of objects (b) Effective missing period

(c) Speed (d) TO threshold

Figure 6.10: Accuracy of DC-TGARD versus the space–time-prism method under varying parameters

varied the TO threshold from 20 min to 100 mins. Again, DC-TGARD outperforms space-time prism as we increase TO threshold (Figure 6.10d)). We also see that greater TO threshold results in less N_r, resulting in more false negatives which are more accurately captured by the proposed DC-TGARD algorithm.

6.7.2 Computational Efficiency

We then compared the proposed DC-TGARD against the baseline TGARD algorithm on computation efficiency using Number of Nodes, Number of Objects, Effective Missing Period, Speed MS and Time Overlap Threshold (TO).

(1) Number of Objects: We set the EMP to 4 hours, MS to 30 m/s, Time Overlap

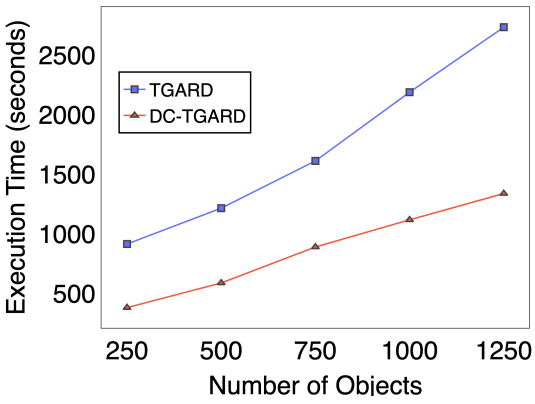
Threshold (T) to 30 mins, and we varied the number of objects from 500 to 2500. Figure 6.11a shows that DC-TGARD always runs faster than the TGARD algorithm because its dual-convergence time-slicing operation uses early stopping criteria.

(2) Effective Missing Period (EMP): We set the number of objects at 500, the number of nodes of $5x10^4$, MS at 30 m/s, TO threshold to 30 mins, and varied the EMP from 30 to 90 minutes. As shown in Figure 6.11b, we find that DC-TGARD outperforms TGARD with increasing EMP. The bi-directional pruning of the dual convergence approach reduces the time slicing operations.

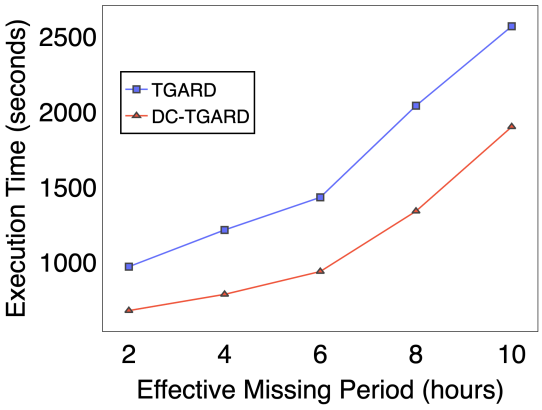
(3) Speed: We kept the number of gaps and EMP constant at 1000 and 4 hours respectively, set the TO threshold to 30 mins, and increased the maximum speed MS from 10 to 50 in m/s units. Figure 6.11c shows DC-TGARD is faster than TGARD. Hence, the dual convergence of DC-TGARD helps, although the gain is smaller as MS increases. The reason is similar to the EMP results: higher maximum speed produces larger ellipses and decreases the number of non-reachable nodes for both algorithms, resulting in more shortest path computations. However, DC-TGARD outperforms TGARD due to its dual convergence property, and the execution times for both algorithms are linear as the speed increases.

(4) Time Overlap Threshold (TO): We again kept the number of objects at 1000, the EMP at 4 hours, and MS at 30 m/s, but this time we increased the TO threshold from 20 min to 100 mins. Figure 6.11d shows DC-TGARD outperforms TGARD as we increase the TO threshold, i.e., the dual convergence nature of DC-TGARD efficiently filters out more N_r and fewer shortest path computations are performed.

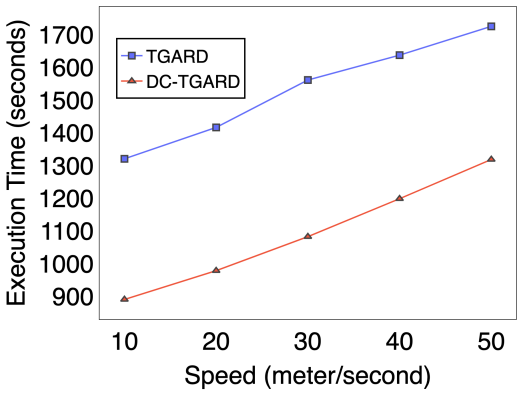
(5) Number of Nodes: We performed a high-density test by setting the number of objects to 1000 and varying the number of nodes from 500 to 2500. For the low-density test, we set the number of objects to 100 and varied the number of nodes from 200 to 1000. For both experiments, EMP was set to 4 hours, MS to 30 m/s, and Time Overlap Threshold (TO) to 40 mins for a fixed study area. Figure 5 6.11f and 6.11e show that DC-TGARD always outperforms the TGARD algorithm. Its time slicing operations are effective for both high-density and low-density road networks.



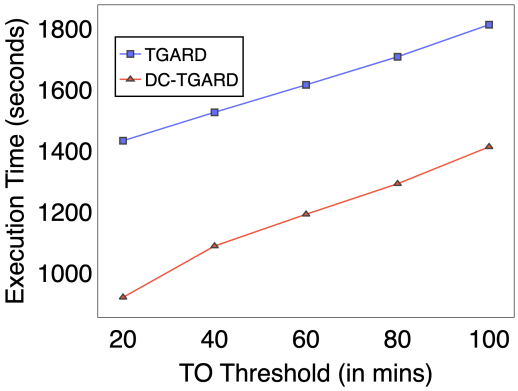
(a) Number of objects



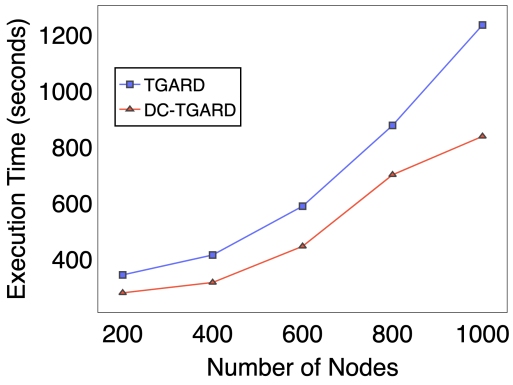
(b) Effective missing period



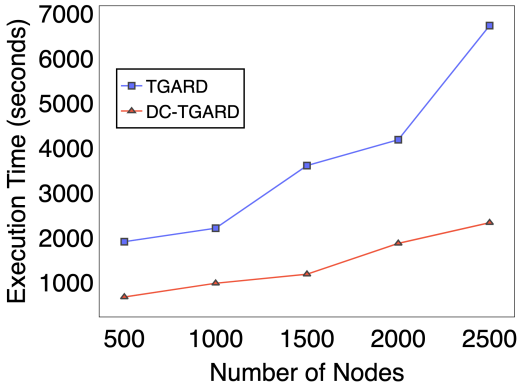
(c) Speed



(d) TO threshold



(e) Number of nodes (low density)



(f) Number of nodes (high density)

Figure 6.11: DC-TGARD is more efficient than TGARD under varying parameters

6.8 Conclusion and Future Work

We study the problem of identifying a set of possible rendezvous locations within a trajectory gap in a given spatial network. We theoretically study a time slicing model which provides tighter bounds as compared to traditional space-time prism models. We also proposed a TGARD algorithm for effectively finding rendezvous nodes while filtering non-reachable nodes in the spatial networks to improve solution quality. In addition, we refined our baseline approach with a Dual Convergence TGARD (DC-TGARD) algorithm using bi-directional pruning to further improve computational efficiency. Experimental results on both synthetic and real-world datasets shows that DC-TGARD is faster than TGARD and has better solution quality than other related work based methods.

Future Work: In the future, we plan to investigate the adjustments of the proposed algorithm needed to handle possibly negative travel costs. We also plan to examine whether parallelism can be used to further improve query processing. Finally, we intend to develop a more accurate prism model to calibrate realistic physics-based parameters (e.g., acceleration).

Chapter 7

Conclusion and Future Directions

The growing ubiquity of location-aware technologies and the scale of ship-trajectory data have increased the need for spatiotemporal analytics that can handle incomplete, distorted, or deliberately manipulated data. This thesis investigated these challenges by developing physics-based algorithms and spatial indexing methods to detect abnormal trajectory gaps and identify possible rendezvous events in maritime spaces. These approaches were motivated by denial-based data distortions, instances where ships may *go dark* by turning off their AIS devices, creating trajectory gaps that may signify illicit activities such as illegal fishing or unauthorized cargo transfers.

To address these challenges, this work proposes an anomaly detection framework that spans both known and unknown spatiotemporal signatures. First, it introduced a spatiotemporal co-occurrence model to detect possible rendezvous through geometric overlap of trajectory gaps. Next, it developed an abnormal gap measure (AGM) metric to identify spatiotemporal outliers in trajectories without known signatures, guided by signal coverage maps and domain-specific physics constraints. The thesis further advances current methods by proposing scalable algorithms such as Memo-AGD and STAGD, which use memoization, hierarchical spatial indexing, and spatiotemporal pruning to improve computational efficiency. Lastly, we presented a novel Dynamic Region Merge (DRM) algorithm that incrementally merges regions using AGM scores while supporting early termination to minimize unnecessary computation.

The proposed methods were evaluated on large-scale datasets such as MarineCadastre

and Danish Maritime, demonstrating better anomaly detection and computational scalability compared to traditional interpolation and imputation techniques. Through theoretical analysis, execution traces, and real-world case studies, the thesis provides practical algorithms for anomaly detection in maritime surveillance. Together, these contributions support distortion-aware spatial data science methods and motivate future research into adversarial modeling, multi-modal fusion with satellite imagery, and explainable spatiotemporal AI for global security applications. Table 7.1 summarizes these key results and outlines both short-term (i.e., ST1 and ST2) and long-term (LT1 and LT2) future work.

Table 7.1: Key Contributions and Future Directions

Data Distortion			
	Denial-based Distortion (e.g., Subtraction)	**Additive Noise Distortion (e.g., Addition)**	**Deception-based Distortion (e.g., Substitution)**
Anomaly Detection with Known Spatiotemporal (ST) Signatures	Analyzing Trajectory Gaps for Possible Rendezvous Region **(Chapter 2, 3)**, Towards tighter Bounds for Possible Rendezvous on Spatial Networks **(Chapter 6)**	**ST1:** Distortions in Spatial Query Processing (e.g., Shortest Path, Nearest Neighbor)	**LT2:** Exploring other Deception-based Distortion
Anomaly Detection without known ST Signatures	Physics-based Abnormal Trajectory Gap Detection **(Chapter 4, 5)**	**ST2:** Distortions in Other Spatial Pattern Families (e.g., Hotspots, Co-locations)	
Anomaly Detection with Classification & Prediction	**LT1:** Physics-Guided Foundation Models		

7.1 Short-Term Future Directions

ST1: Data Distortions for Query Processing: In the short term, I plan to study aspects of additive noise distortion. One promising direction is to examine how noise affects spatial query outcomes (e.g., shortest route, nearest neighbor nodes). This would address the inherent limitations of representing road networks as simplified spatial graphs, which introduces geometric and topological approximation (e.g., Mercator projections, geographic maps). Such standard approximations can be considered pre-existing distortion (data uncertainty) since any projection reference system has specific error variance. When performing spatial queries over a distorted study area, it is essential to consider the pre-existing distortion. Meanwhile, the model or algorithm applied to the query has assumptions

(e.g., independent and identical distribution) that further amplify the overall distortion. We can call this induced distortion (model uncertainty). For instance, computing shortest-path queries in terms of travel time may fail to capture spatial autocorrelation patterns and dependencies among road segments and intersections. Current methods often assume edge independence, neglecting spatial autocorrelation where proximate segments exhibit stronger correlations in behavioral patterns, traffic dynamics, and mobility characteristics than distant segments.

Indeed, the challenge of mitigating induced distortions is amplified by trajectory gaps that generate uncertainty regions inadequately bounded by conventional space-time prisms, which fail to reflect inter-segment dependencies and temporal movement dynamics such as those in DC-TGARD. Drawing on time-slicing techniques that yield tighter spatial bounds compared to geo-ellipse methods, future work should prioritize three tasks: (1) developing spatially informed models that integrate distance-decay functions to weight the influence of adjacent segments, (2) refining space-time prisms to incorporate network topology constraints over unconstrained Euclidean assumptions, and (3) employing adaptive correlation matrices responsive to trajectory gap characteristics and real-time network conditions. Moreover, trajectory gap analysis can clarify spatial autocorrelation within network-constrained contexts and help model dependencies arising from missing movement data, including their interplay with network topology and temporal dynamics.

ST2: Data Distortions in other Spatial Pattern Families: Another short-term direction is anomaly detection where there are no known spatiotemporal signatures, especially within broader spatial pattern families in the context of *pre-existing* and *induced* distortion discussed in ST1. While this thesis focused on denial-based distortions (e.g., trajectory gaps), both pre-existing and induced additive distortions may disrupt detection of emergent phenomena such as regional hotspots or feature colocations. This pattern is also significant in monitoring maritime traffic, as shown in Figure 7.1, where traffic changes abruptly over time due to external events. Additive distortions, characterized by random noise perturbing spatial coordinates or attributes, affect hotspot detection by altering the perceived concentration of events. Attribute noise, such as inflated crime counts, can skew statistical measures, leading to incorrect hotspot rankings. To mitigate these effects, statistical methods such as enhanced significance testing with effect size

considerations and domain-specific constraints (e.g., minimum hotspot size) can filter out noise-induced artifacts, enabling more reliable detection in applications such as epidemiology or criminology.

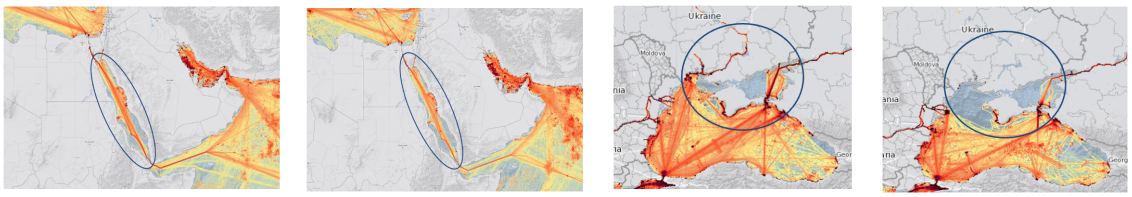


Figure 7.1: Marine traffic volume near Ukraine ports before and during disturbance phase

For **colocation** detection, additive distortions disrupt the spatial relationships critical to identifying features that frequently occur together, such as symbiotic species. Coordinate noise alters distances in neighbor graphs. This means it can misrepresent proximity and skew measures such as participation indices or statistical functions, potentially missing true colocations or suggesting false ones. In addition, geographically weighted regression (GWR), a traditional model that applies location-specific weighting [103], is more effective for linear-regression tasks. In microscale-domain applications (e.g., medical oncology), such patterns also struggle to capture key spatial interactions because they assume homogeneous regions [104] and require manual inspection by pathologists. Current spatially aware methods, such as SVANN [105, 106], aim to address spatial variability through deep learning by using location-dependent weights and learning a space partitioning of heterogeneous data. Notably, these works focus on Euclidean space and overlook variability in *non-Euclidean* spaces such as point clouds and multi-category point sets. Other methods [107, 108] learn to partition space to handle heterogeneous data, but they require dense training samples for each domain and struggle with high intra-domain variability. Most importantly, these approaches rely on conventional CNNs, which are limited to regular grids and thus ineffective in non-Euclidean spaces. Hence, a promising future direction is to reformulate the original Euclidean point set as a non-Euclidean graph structure, a transformation explored in studies such as [109]. Such a graph representation encodes relational information, capturing connections and proximity relationships that extend beyond those provided by traditional Euclidean frameworks of distances and angles.

7.2 Long-Term Future Directions

My work on subtractive denial-based and additive distortion culminates in a long-term plan to develop physics-guided foundation models, followed by investigation of the emerging challenge of deception-based distortion, and ultimately by extending these ideas into three-dimensional and immersive settings for autonomous driving.

LT1: Physics-Guided Foundation Models My first long-term plan is to develop physics-guided generative foundation models (PGFM) that are physically grounded and explainable. A generative model is a statistical model of the joint probability distribution over an observable variable and target variable, and it can generate random instances of an observation. When trained on broad and diverse data, it serves as a foundation model that reduces the time, energy, and labeled samples required for multiple downstream tasks. Due to the recent success of models like the generative pre-trained transformer (GPT), there is ongoing excitement about foundation models. However, these models often struggle with out-of-distribution generalization and may yield unrealistic or physically invalid outputs. A physics-guided foundation model would incorporate scientific knowledge that applies across a wide range of downstream tasks.

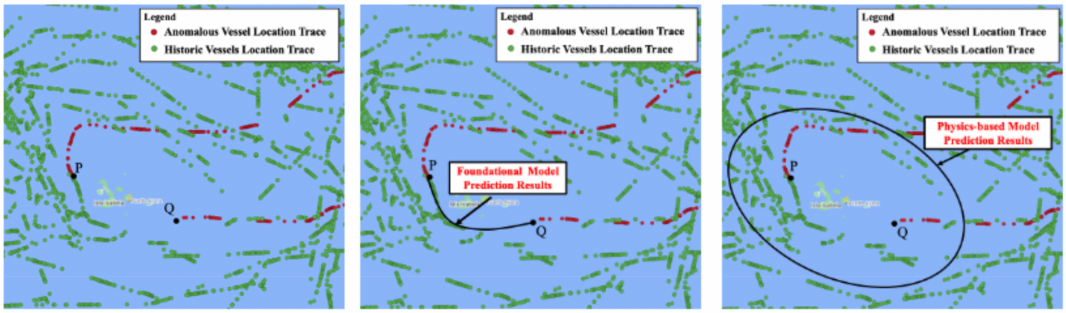


Figure 7.2: An illustrative example of prediction output comparison

For instance, consider the problem of filling trajectory gaps for various downstream public security and safety tasks. In these downstream tasks, a key challenge is denial, where a moving object (e.g., a ship, vehicle, phone, device, or person) temporarily stops reporting its location, creating a gap in its trajectory despite monitoring mandates. Such data gaps largely confound data-driven approaches such as traditional foundation models. However,

broad knowledge, such as the relationship between speed, elapsed time, and location, as well as the laws of motion, may help to bound the location via a space-time prism model. Figure 7.2 shows a foundation model (FM) that trains on historical location data for the entire study area and outputs the most likely path an object would have taken while traveling from P to Q. By contrast, a physics-guided foundation model (PGFM) can use a simple physics-based model (e.g., a space-time prism) to constrain the object's movement in the form of a geo-ellipse, based on maximum speed (but unconstrained acceleration for simplicity) and with no prior knowledge of existing data. If downstream applications are able to specify vehicle-type-specific parameters (e.g., maximum speed), this approach can potentially reduce the number of training samples and time required for fine-tuning for downstream adversarial applications, while also enabling out-of-sample predictions. PGFMs could also handle more sophisticated additive noise distortions, where users deliberately fake their movements from a remote location (e.g., a near-shore region). This means a PGFM could have flagged the tanker "Cathay Phoenix" [5] as anomalous using physically informed models such as the Kinematic Bicycle Model (KBM) [110] as prior knowledge. Figure 7.3 presents the tanker's bearing distribution, reflecting its sudden changes in direction as indicated by the red circle on the right. The angular distribution throughout this phase is depicted in the frequency distribution, outlined in the red box. Moreover, there have been other incidents, such as displacement events, where one ship broadcasts spoofed signals thousands of miles away. Such distortion can be captured using KBM as a regularizer within the PGFM framework, paired with generative models [21] such as diffusion models [111].

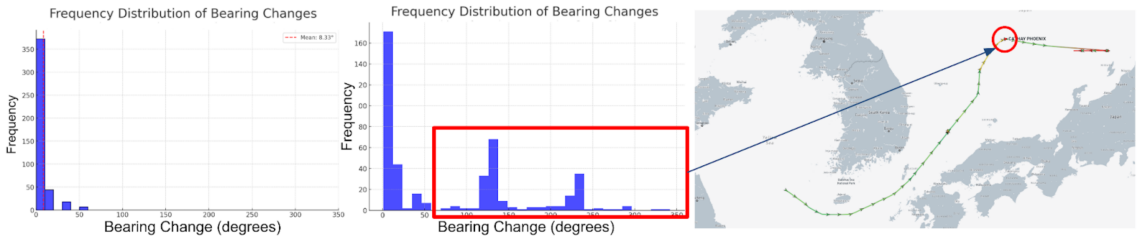
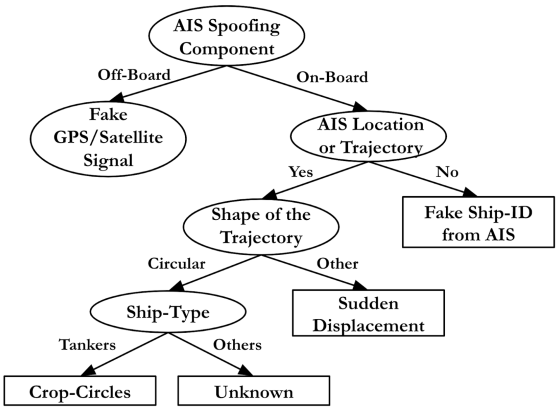


Figure 7.3: Frequency distribution of the bearing change of a tanker "Cathay Phoenix", whose abrupt changes are highlighted in red

LT2: Exploring Other Deception-Based Distortion: While this thesis primarily addressed denial-based distortions, deception-based distortions (e.g., GPS spoofing, falsified AIS data) pose an equally serious threat. As a long-term direction, I plan to explore how substitution-based distortions affect pattern detection and classification pipelines. Initial steps would involve formalizing deceptive patterns, such as trajectory mimicry or deepfake-like location substitution. It would be worthwhile to study how deception differs from noise or dropout and examine how hybrid methods that combine anomaly detection with provenance analysis can distinguish between synthetic and genuine movement traces. This direction is important for establishing trust in location-based systems, particularly in adversarial settings such as military surveillance, sanctions enforcement, or supply chain monitoring.

Criteria	Attack model
Component	Broadcasting a fake location from the transponder
Component of the System	Hardware, software, or transmission in AIS.
AIS Location or Trajectory	Broadcasting fake vessel types
Shape of Trajectory	Abnormal movement patterns indicate possible spoofing.
Signal Strength	Signal anomalies suggest jamming or spoofing activities.
Timing Patterns	Irregular transmission intervals hint at spoofing.
Vessel Operation	Mismatched activity (e.g., fishing) with actual vessel operations.
Vessel Behavior	Unusual maneuvers that may signal unauthorized control.
Communication Frequency	Atypical broadcast rates are a sign of jamming or flooding.
Navigational Status	Status reports do not align with vessel behavior.
Traffic Density	Dense areas may mask or enable more attack attempts.

(a) Criteria and their respective attack models



(b) Classification of spoofing patterns

Figure 7.4: A taxonomy of deception-based distortion

Figure 7.4a lists attack models based on multiple criteria for GPS spoofing scenarios, drawn from case studies. This initial classification lays the foundation for subsequent research steps, including data collection, anomaly detection, model training, and the eventual deployment of systems to monitor and mitigate these threats in maritime traffic, ultimately aiding in the classification of deception-based attacks. Figure 7.4b shows a taxonomy of attack models that can be classified as deception-based distortions. Common attack types include spoofing, where false AIS signals are transmitted to create fake vessels or alter locations; jamming, which disrupts AIS signals, preventing effective communication;

falsification, involving the alteration of a vessel's identity or location details; and replay attacks, where previously captured AIS data is retransmitted to produce misleading historical information.

LT3: Toward 3D and Immersive Settings for Autonomous Driving: The methods in this thesis operate on two-dimensional trajectories and fields. I expect the same ideas to carry into three-dimensional scenes and into the simulated environments used to develop autonomous systems, and I see this as the most promising place to take the work next. Autonomous driving is the clearest case. Self-driving systems are tested heavily in simulation, and the value of a test depends on how realistic the simulated world is. If the synthetic scenes and sensor streams carry the same distortions a car meets on the road, the occlusions that deny part of the view, the sensor noise, and the occasional spoofed or inconsistent signal, then the test measures something close to real reliability. A distortion-aware spatial AI is well suited to generate exactly these conditions, with the space-time prism and the kinematic bicycle model keeping the generated motion physically possible rather than merely plausible-looking.

Concretely, I would couple distortion-aware generative models of motion with three-dimensional scene reconstruction and with virtual and augmented reality. Recent results on reconstructing scenes from ordinary imagery, on open-vocabulary three-dimensional segmentation, and on physics-constrained generation already supply most of the needed components. Brought together, they would let an autonomous-driving team rehearse a planner against rare and adversarial situations, a denied lane marking, a spoofed neighboring vehicle, or a sensor dropout in a tunnel, inside an immersive environment where a human can step in and inspect what the system saw and decided. Moving distortion-aware spatial AI from flat trajectories into immersive, three-dimensional, real-time settings is, in my view, where the work becomes most useful.

LT4: Toward Distortion-Aware World Models: The three-dimensional and immersive direction above connects to a broader conversation about world models, a term now used across computer vision, graphics, and robotics. A recent functional taxonomy organizes these systems into three roles: renderers that output pixels for a person to look at, simulators that output geometrically and physically faithful state, and planners that output actions [112]. That same taxonomy argues simulation is the role that bridges the

other two, since a planner is only as trustworthy as the state it is tested against. This is where distortion-aware spatial AI fits most directly. The generative methods sketched above do not aim for photorealism alone; they aim for state that is physically faithful and faithful in its distortions, carrying the denial, noise, and deception that real sensing suffers. A simulator built this way would let a planner rehearse against failures that matter: a denied lane marking, a spoofed neighboring vehicle, or a sensor dropout in a tunnel, with the space-time prism and the kinematic bicycle model keeping every generated scene physically possible. Distortion-aware world models would give spatial data science a clear role in world-model research by making the realism of failure part of the simulation target.

References

[1] BOEM and NOAA. Marinecadastre. https://marinecadastre.gov/ais/, 2020.

[2] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition, and productivity. *McKinsey Global Institute Report*, 2011.

[3] Paul Tullis. The world economy runs on gps. it needs a backup plan. *Bloomberg Businessweek, July*, 25, 2018.

[4] Riccardo Coppola and Maurizio Morisio. Connected car: technologies, issues, future trends. *ACM Computing Surveys (CSUR)*, 49(3):1–36, 2016.

[5] Christiaan Triebert, Blacki Migliozzi, Alexander Cardia, Muyi Xiao, and David Botti. Fake signals and american insurance: How a dark fleet moves russian oil. *International New York Times*, pages NA–NA, 2023.

[6] Julia Stoyanovich, Bill Howe, and Hosagrahar Visvesvaraya Jagadish. Responsible data management. *Proceedings of the VLDB Endowment*, 13(12), 2020.

[7] James Kraska. Maritime interdiction of north korean ships under un sanctions. *Ecology LQ*, 46:215, 2019.

[8] Kenneth Katzman. *To Evade Sanctions in Iran, Ships Vanish in Plain Sight*. International Herald Tribune, 2013.

[9] Robert A Weber and Kevin M O'Connell. Intelligence agencies pushed to use more commercial satellites. *International New York Times*, pages NA–NA, 2021.

[10] Jahangir Amuzegar. Oil backed up, iranians put it on idled ships. *International New York Times*, pages NA–NA, 2012.

[11] Keith Bradsher. Tariff dodgers stand to profit off us-china trade dispute. *International New York Times*, pages NA–NA, 2018.

[12] Lydia J Treanor. Illegal, unreported, and unregulated fishing and national security: Amending federal law to navigate an ocean of illegality. *Tul. Mar. LJ*, 48:493, 2023.

[13] Sarah Gibbens. How illegal fishing is being tracked from space. *National Geographic*, 2018.

[14] J Baker. Transparent transhipping: detecting illegal fishing with satellite data. *Ship Technology*, 2018.

[15] Virma Simonette and Joel Guinto. Dangerous manoeuvres' in china and philippines' cat-and-mouse sea chase. *Ship Technology*, 2018.

[16] Harvey J Miller. Modelling accessibility using space-time prism concepts within geographical information systems. *International Journal of Geographical Information System*, 5(3):287–301, 1991.

[17] Donald E Stokes. *Pasteur's quadrant: Basic science and technological innovation*. Brookings Institution Press, 2011.

[18] Henry Kautz. The third AI summer: AAAI robert S. engelmore memorial lecture. *AI Magazine*, 43(1):105–125, 2022.

[19] Mingzhou Yang. *Physics-aware Neuro-Symbolic AI for Spatial Computing*. PhD thesis, University of Minnesota, 2026.

[20] Mingzhou Yang, Ruolei Zeng, Arun Sharma, Shunichi Sawamura, William F. Northrop, and Shashi Shekhar. Towards pareto-optimality with multi-level bi-objective routing: A summary of results. In *Proceedings of the 17th ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS)*, pages 36–45, 2024.

[21] Arun Sharma, Mingzhou Yang, Majid Farhadloo, Subhankar Ghosh, Bharat Jayaprakash, and Shashi Shekhar. Towards physics-informed diffusion for anomaly detection in trajectories. *arXiv preprint arXiv:2506.06999*, 2025.

[22] Mingzhou Yang, Arun Sharma, Majid Farhadloo, Bharat Jayaprakash, and Shashi Shekhar. Geo-lucid conditional diffusion models for high physical fidelity trajectory generation. In *Proceedings of the 33rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 382–395, 2025.

[23] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):1–41, 2015.

[24] Somayeh Dodge, Robert Weibel, and Anna-Katharina Lautenschütz. Towards a taxonomy of movement patterns. *Information visualization*, 7(3-4):240–252, 2008.

[25] Emre Eftelioglu, Xun Tang, and Shashi Shekhar. Avoidance region discovery: A summary of results. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 585–593. SIAM, 2018.

[26] Bart Kuijpers, Harvey J Miller, and Walied Othman. Kinetic prisms: incorporating acceleration limits into space–time prisms. *International Journal of Geographical Information Science*, 31(11):2164–2194, 2017.

[27] Jürg Nievergelt and Franco P Preparata. Plane-sweep algorithms for intersecting geometric figures. *Communications of the ACM*, 25(10):739–747, 1982.

[28] Kathleen Hornsby and Max J Egenhofer. Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence*, 36(1-2):177–194, 2002.

[29] Bart Kuijpers, Harvey J Miller, Tijs Neutens, and Walied Othman. Anchor uncertainty and space-time prisms on road networks. *International Journal of Geographical Information Science*, 24(8):1223–1248, 2010.

[30] Mei-Po Kwan. Gis methods in time-geographic research: Geocomputation and geovisualization of human activity patterns. *Geografiska Annaler: Series B, Human Geography*, 86(4):267–280, 2004.

[31] Tijs Neutens, Tim Schwanen, and Frank Witlox. The prism of everyday life: Towards a new research agenda for time geography. *Transport reviews*, 31(1):25–47, 2011.

[32] Maria Riveiro, Giuliana Pallotta, and Michele Vespe. Maritime anomaly detection: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(5):e1266, 2018.

[33] Po-Ruey Lei. A framework for anomaly detection in maritime trajectory behavior. *Knowledge and Information Systems*, 47(1):189–214, 2016.

[34] Elena Camossi, Paola Villa, and Luca Mazzola. Semantic-based anomalous pattern discovery in moving object trajectories. *arXiv preprint arXiv:1305.1946*, 2013.

[35] Hamed Yaghoubi Shahir, Uwe Glässer, Narek Nalbandyan, and Hans Wehn. Maritime situation analysis: A multi-vessel interaction and anomaly detection framework. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 192–199. IEEE, 2014.

[36] Bo Liu, Erico N de Souza, Cassey Hilliard, and Stan Matwin. Ship movement anomaly detection using specialized distance measures. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1113–1120. IEEE, 2015.

[37] Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, Jose Antonio de Macedo, Fabio Porto, and Christelle Vangenot. A conceptual view on trajectories. *Data & knowledge engineering*, 65(1):126–146, 2008.

[38] Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 863–868. ACM, 2008.

[39] Jose Antonio MR Rocha, Valéria C Times, Gabriel Oliveira, Luis O Alvares, and Vania Bogorny. Db-smot: A direction-based spatio-temporal clustering method. In *2010 5th IEEE international conference intelligent systems*, pages 114–119. IEEE, 2010.

[40] Hyun-Mi Kim and Mei-Po Kwan. Space-time accessibility measures: A geocomputational algorithm with a focus on the feasible opportunity set and possible activity duration. *Journal of geographical Systems*, 5(1):71–91, 2003.

[41] Qing Zhu, Jun Gong, and Yeting Zhang. An efficient 3d r-tree spatial index method for virtual geographic environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(3):217–224, 2007.

[42] Dieter Pfoser, Christian S Jensen, and Yannis Theodoridis. Novel approaches to the indexing of moving object trajectories. In *VLDB*, pages 395–406. The VLDB Endowment, 2000.

[43] Ahmed R Mahmood, Walid G Aref, Ahmed M Aly, and Saleh Basalamah. Indexing recent trajectories of moving objects. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 393–396. ACM, 2014.

[44] Bart Kuijpers and Walied Othman. Modeling uncertainty of moving objects on road networks via space–time prisms. *International Journal of Geographical Information Science*, 23(9):1095–1117, 2009.

[45] Bart Kuijpers, Rafael Grimson, and Walied Othman. An analytic solution to the alibi query in the space–time prisms model for moving object data. *International Journal of Geographical Information Science*, 25(2):293–322, 2011.

[46] Edwin H Jacox and Hanan Samet. Spatial join techniques. *ACM Transactions on Database Systems (TODS)*, 32(1):7–es, 2007.

[47] Goce Trajcevski, Alok Choudhary, Ouri Wolfson, Li Ye, and Gang Li. Uncertain range queries for necklaces. In *2010 Eleventh International Conference on Mobile Data Management*, pages 199–208. IEEE, 2010.

[48] Katerina Sofrona. Why cannot i see a vessel on the live map?, Oct 2017.

[49] Shuo Shang, Lisi Chen, Zhewei Wei, Christian S Jensen, Kai Zheng, and Panos Kalnis. Trajectory similarity join in spatial networks. In *Proceedings of the VLDB Endowment*, volume 10. VLDB Endowment, 2017.

[50] Petko Bakalov, Marios Hadjieleftheriou, Eamonn Keogh, and Vassilis J Tsotras. Efficient trajectory joins using symbolic representations. In *Proceedings of the 6th International Conference on Mobile Data Management*, pages 86–93. ACM, 2005.

[51] Petko Bakalov and Vassilis J Tsotras. Continuous spatiotemporal trajectory joins. In *International conference on GeoSensor Networks*, pages 109–128. Springer, 2006.

[52] Hui Ding, Goce Trajcevski, and Peter Scheuermann. Efficient similarity join of large sets of moving object trajectories. In *15th International Symposium on Temporal Representation and Reasoning*, pages 79–87. IEEE, 2008.

[53] Arun Sharma, Xun Tang, Jayant Gupta, Majid Farhadloo, and Shashi Shekhar. Analyzing trajectory gaps for possible rendezvous: A summary of results. In *11th International Conference on Geographic Information Science (GIScience 2021)-Part I*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[54] Lei Chen, M Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2005.

[55] Rake Lin, King-lp Agrawal, Harpreet S Sawhney Shim, and Kyuseok. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceeding of the 21th International Conference on Very Large Data Bases*, pages 490–501. Citeseer, 1995.

[56] Byoung-Kee Yi, Hosagrahar V Jagadish, and Christos Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proceedings 14th International Conference on Data Engineering*, pages 201–208. IEEE, 1998.

[57] Na Ta, Guoliang Li, Yongqing Xie, Changqi Li, Shuang Hao, and Jianhua Feng. Signature-based trajectory similarity join. *IEEE Transactions on Knowledge and Data Engineering*, 29(4):870–883, 2017.

[58] Yun Chen and Jignesh M Patel. Design and evaluation of trajectory join algorithms. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 266–275. ACM, 2009.

[59] Chih-Chieh Hung, Wen-Chih Peng, and Wang-Chien Lee. Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. *The VLDB Journal*, 24(2):169–192, 2015.

[60] Dieter Pfoser and Christian S Jensen. Capturing the uncertainty of moving-object representations. In *International Symposium on Spatial Databases*, pages 111–131. Springer, 1999.

[61] Reynold Cheng, Jinchuan Chen, Mohamed Mokbel, and Chi-Yin Chow. Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data. In *24th International Conference on Data Engineering*, pages 973–982. IEEE, 2008.

[62] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[63] Giuliana Pallotta, Michele Vespe, and Karna Bryan. Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction. *Entropy*, 15(6):2218–2245, 2013.

[64] Chao Chen, Daqing Zhang, Pablo Samuel Castro, Nan Li, Lin Sun, Shijian Li, and Zonghui Wang. iboat: Isolation-based online anomalous trajectory detection. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):806–818, 2013.

[65] Stephan Winter and Zhang-Cai Yin. Directed movements in probabilistic time geography. *International Journal of Geographical Information Science*, 24(9):1349–1365, 2010.

[66] Pengxiang Zhao, David Jonietz, and Martin Raubal. Applying frequent-pattern mining and time geography to impute gaps in smartphone-based human-movement data. *International Journal of Geographical Information Science*, 35(11):2187–2215, 2021.

[67] Marinetraffic. https://www.marinetraffic.com/en/ais/, 2020.

[68] Robert Skulstad, Guoyou Li, Thor I Fossen, Bjørnar Vik, and Houxiang Zhang. Dead reckoning of dynamically positioned ships: Using an efficient recurrent neural network. *IEEE Robotics & Automation Magazine*, 26(3):39–51, 2019.

[69] Eric Rancourt. Estimation with nearest neighbour imputation at statistics canada. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, pages 131–138, 1999.

[70] Arun Sharma, Jayant Gupta, and Shashi Shekhar. Abnormal trajectory-gap detection: A summary (short paper). In *15th International Conference on Spatial Information Theory (COSIT 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[71] Fabio Mazzarella, Michele Vespe, Alfredo Alessandrini, Dario Tarchi, Giuseppe Aulicino, and Antonio Vollero. A novel anomaly detection approach to identify intentional ais on-off switching. *Expert Systems with Applications*, 78:110–123, 2017.

[72] Jiahua Chen and Jun Shao. Jackknife variance estimation for nearest-neighbor imputation. *Journal of the American Statistical Association*, 96(453):260–269, 2001.

[73] Estevam R Hruschka, Eduardo R Hruschka, and Nelson FF Ebecken. Bayesian networks for imputation in classification problems. *Journal of Intelligent Information Systems*, 29:231–252, 2007.

[74] K Vimal Kumar, Divakar Yadav, and Arun Sharma. Graph based technique for hindi text summarization. In *Information Systems Design and Intelligent Applications: Proceedings of Second International Conference INDIA 2015, Volume 1*, pages 301–310. Springer, 2015.

[75] Arun Sharma, Jayant Gupta, and Subhankar Ghosh. Towards a tighter bound on possible-rendezvous areas: preliminary results. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pages 1–11. ACM, 2022.

[76] Arun Sharma and Shashi Shekhar. Analyzing trajectory gaps to find possible rendezvous region. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(3):1–23, 2022.

[77] Arun Sharma, Majid Farhadloo, Yan Li, Jayant Gupta, Aditya Kulkarni, and Shashi Shekhar. Understanding covid-19 effects on mobility: A community-engaged approach. *AGILE: GIScience Series*, 3:14, 2022.

[78] John Fox. *Applied regression analysis and generalized linear models*. Sage Publications, 2015.

[79] Arun Sharma, Zhe Jiang, and Shashi Shekhar. Spatiotemporal data mining: A survey. *arXiv preprint arXiv:2206.12753*, 2022.

[80] Subhankar Ghosh, Jayant Gupta, Arun Sharma, Shuai An, and Shashi Shekhar. Towards geographically robust statistically significant regional colocation pattern detection. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoSpatial Simulation*, pages 11–20. ACM, 2022.

[81] Subhankar Ghosh, Jayant Gupta, Arun Sharma, Shuai An, and Shashi Shekhar. Reducing false discoveries in statistically-significant regional-colocation mining: A summary of results. In *12th International Conference on Geographic Information Science (GIScience 2023)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2023.

[82] Jayant Gupta and Arun Sharma. Mining taxonomy-aware colocations: a summary of results. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pages 1–11. ACM, 2022.

[83] Majid Farhadloo, Arun Sharma, Jayant Gupta, Alexey Leontovich, Svetomir N Markovic, and Shashi Shekhar. Towards spatially-lucid ai classification in non-euclidean space: An application for mxif oncology data. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 616–624. SIAM, 2024.

[84] BBC. North korea: South seizes ship amid row over illegal oil transfer. https://www.bbc.com/news/world-asia-42510783, 2017.

[85] Ugur Demiryurek, Farnoush Banaei-Kashani, Cyrus Shahabi, and Anand Ranganathan. Online computation of fastest path in time-dependent spatial networks. In *International Symposium on Spatial and Temporal Databases*, pages 92–111. Springer, 2011.

[86] Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, and Guang-Zhong Sun. An interactive-voting based map matching algorithm. In *2010 Eleventh international conference on mobile data management*, pages 43–52. IEEE, 2010.

[87] Reaz Uddin, Michael N Rice, Chinya V Ravishankar, and Vassilis J Tsotras. Assembly queries: Planning and discovering assemblies of moving objects using partial information. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10. ACM, 2017.

[88] Venkata Gunturi, Ernesto Nunes, KwangSoo Yang, and Shashi Shekhar. A critical-time-point approach to all-start-time lagrangian shortest paths: A summary of results. In *International Symposium on Spatial and Temporal Databases*, pages 74–91. Springer, 2011.

[89] George Kollios, Dimitrios Gunopulos, and Vassilis J Tsotras. On indexing mobile objects. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 261–272. ACM, 1999.

[90] Su Chen, Beng Chin Ooi, Kian-Lee Tan, and Mario A Nascimento. St2b-tree: a self-tunable spatio-temporal b+-tree index for moving objects. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 29–42. ACM, 2008.

[91] Jignesh M Patel, Yun Chen, and V Prasad Chakka. Stripes: an efficient index for predicted trajectories. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 635–646. ACM, 2004.

[92] Goce Trajcevski, Ouri Wolfson, Klaus Hinrichs, and Sam Chamberlain. Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems (TODS)*, 29(3):463–507, 2004.

[93] Goce Trajcevski. Probabilistic range queries in moving objects databases with uncertainty. In *Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access*, pages 39–45. ACM, 2003.

[94] Joshua S Greenfeld. Matching gps observations to locations on a digital map. In *Transportation Research Board 81st Annual Meeting*, volume 22, 2002.

[95] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM*

SIGSPATIAL international conference on advances in geographic information systems, pages 352–361. ACM, 2009.

[96] Kai Zheng, Goce Trajcevski, Xiaofang Zhou, and Peter Scheuermann. Probabilistic range queries for uncertain trajectories on road networks. In *Proceedings of the 14th International Conference on Extending Database Technology*, pages 283–294. ACM, 2011.

[97] Kai Zheng, Yu Zheng, Xing Xie, and Xiaofang Zhou. Reducing uncertainty of low-sampling-rate trajectories. In *2012 IEEE 28th International Conference on Data Engineering*, pages 1144–1155. IEEE, 2012.

[98] John Krumm and Eric Horvitz. Predestination: Inferring destinations from partial trajectories. In *UbiComp*, pages 243–260. Springer, 2006.

[99] John Krumm, Robert Gruen, and Daniel Delling. From destination prediction to route prediction. *Journal of Location Based Services*, 7(2):98–120, 2013.

[100] Liang Zhao, Tatsuya Ohshima, and Hiroshi Nagamochi. A* algorithm for the time-dependent shortest path problem. In *WAAC08: The 11th Japan-Korea Joint Workshop on Algorithms and Computation*, 2008.

[101] Bolin Ding, Jeffrey Xu Yu, and Lu Qin. Finding time-dependent shortest paths over large graphs. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, pages 205–216. ACM, 2008.

[102] Yu Zheng, Xing Xie, and Wei-Ying Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33(2):32–39, 2010.

[103] C Brunsdon et al. Some notes on parametric significance tests for geographically weighted regression. *Journal of regional science*, 39(3):497–524, 1999.

[104] M Farhadloo et al. Spatial computing opportunities in biomedical decision support: The atlas-ehr vision. *arXiv preprint arXiv:2305.09675*, 2023.

[105] J Gupta et al. Towards spatial variability aware deep neural networks (svann): A summary of results. *1st ACM SIGKDD DeepSpatial Workshop*, 2020.

[106] J Gupta et al. Spatial variability aware deep neural networks (svann): A general approach. *ACM TIST*, 12(6):1–21, 2021.

[107] Y Xie et al. A statistically-guided deep network transformation and moderation framework for data with spatial heterogeneity. In *2021 IEEE ICDM*, pages 767–776, 2021.

[108] Y Xie et al. Spatial-net: A self-adaptive and model-agnostic deep learning framework for spatially heterogeneous datasets. In *Proceedings of ACM GIS*, pages 313–323, 2021.

[109] M Bronstein et al. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[110] Philip Polack, Florent Altché, Brigitte d'Andréa Novel, and Arnaud de La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *2017 IEEE intelligent vehicles symposium (IV)*, pages 812–818. IEEE, 2017.

[111] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[112] Fei-Fei Li and World Labs. A functional taxonomy of world models, 2026. World Labs. Available at https://drfeifei.substack.com/p/a-functional-taxonomy-of-world-models.