

Ricardo Daniel Cardoso Pereira

AIS Data Visualization applied to the identification of anomalous vessels' movements on the Portuguese maritime territory

Dissertation
Master in Informatics Engineering
advised by Professor Pedro H. Abreu, PhD and Professor Fernando P. Machado, PhD
and presented to the Department of Informatics Engineering
of the Faculty of Sciences and Technology of the University of Coimbra

July 2018



UNIVERSIDADE DE COIMBRA

This page is intentionally left blank.



FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Masters' Degree in Informatics Engineering
Final Dissertation

**AIS Data Visualization applied to the identification of
anomalous vessels' movements on the Portuguese mar-
itime territory**

Author:

Ricardo Daniel Cardoso Pereira

rdpereira@student.dei.uc.pt

Advisors:

Professor **Pedro H. Abreu**, PhD

Professor **Fernando P. Machado**, PhD

Coimbra, 2018

This page is intentionally left blank.

Acknowledgements

I would first like to thank my advisor, Professor Pedro Henriques Abreu, for all the support that he gave me during this year of work, which was essential for the success of this thesis.

I would also like to thank my co-advisor, Professor Fernando Penousal Machado, for the technical assistance that helped the work to evolve on the right direction.

To my lab partners I want to thank for the opportunity to work in such a great environment.

To my closest ones, thank you for all the support, patience and good times during this journey.

Finally, to my parents and sister, I want to thank all the efforts they made during my life that made possible for me to achieve this goal.

This page is intentionally left blank.

Abstract

A few years ago the Automatic Identification System (AIS) was introduced as the international communication standard for vessels with the propose of improving maritime safety, but nowadays it is used for more proposes mainly because its data has the potential of mapping with detail the entire maritime traffic of an area. One of this new proposes is assisting law enforcement in detecting abnormal behaviors through movement analysis of the vessels. Because of that, several scientific works addressing AIS data have been published based on machine learning and data visualization approaches, in distinct areas such as trajectory mining, traffic visualization and anomaly detection. However, considering this last area, only machine learning approaches have been proposed, while the data visualization works tend to be focused on representing the vessel's traffic without any consideration for the anomalous behaviors. Therefore, this thesis is focused in developing visualization strategies that are able to identify these behaviors, with the assistance of data analysis, and in testing them with AIS data from the Portuguese maritime zone. These strategies were implemented on a platform and they include approaches for a general analysis of the data and for detecting specific types of anomalous behaviors. The validation, made through case studies, showed that the approaches are effective and can be used as a support tool for the domain experts.

Keywords

Automatic Identification System, Portuguese Maritime Territory, Traffic Visualization, Trajectory Mining, Anomaly Detection, Data Visualization, Data Analysis

This page is intentionally left blank.

Resumo

Há poucos anos atrás o Sistema de Identificação Automática (AIS) foi definido como o standard internacional para a comunicação entre navios com o objetivo de melhorar a segurança marítima, mas hoje em dia é utilizado para muitos mais fins porque os seus dados têm o potencial de conseguirem mapear todo o tráfego marítimo de uma determinada zona. Um desses fins é ajudar as autoridades a detetarem comportamentos anómalos através da análise dos movimentos dos navios. Desta forma, vários trabalhos científicos relacionados com dados do AIS têm sido publicados, apresentando abordagens de aprendizagem computacional e de visualização de informação, em áreas tão distintas como a extração de trajetórias, visualização de tráfego e deteção de anomalias. No entanto, considerando esta última área, apenas abordagens de aprendizagem computacional foram propostas, enquanto os trabalhos na área da visualização de informação tendem a propor representações do tráfego dos navios sem qualquer destaque aos comportamentos anómalos. Assim sendo, a presente tese tem como objetivo o desenvolvimento de estratégias de visualização capazes de identificar comportamentos anómalos, com a assistência de técnicas de análise de dados, e o teste dessas estratégias com dados AIS da zona marítima Portuguesa. Estas estratégias foram implementadas numa plataforma e incluem abordagens para uma análise geral dos dados e para a deteção de tipos específicos de comportamentos anómalos. A validação, feita através de casos de estudo, mostrou que as abordagens funcionam e que podem ser utilizadas como ferramenta de suporte aos peritos da área.

Palavras-Chave

Sistema de Identificação Automática, Território Marítimo Português, Visualização de Tráfego, Extração de Trajetórias, Deteção de Anomalias, Visualização de Informação, Análise de Dados

This page is intentionally left blank.

Contents

1	Introduction	1
1.1	Contextualization	2
1.2	Research Goals	3
1.3	Work Plan	4
1.4	Research Contributions	7
1.5	Document Structure	7
2	Background Knowledge	9
2.1	Data Mining	9
2.2	Data Visualization	15
2.2.1	Jacques Bertin’s Principles	15
2.2.2	Edward Tufte’s Principles	19
2.2.3	Interactive Visualization	23
2.2.4	Visualization Validation	25
2.3	Conclusion	26
3	Related Work	27
3.1	Trajectory Mining	27
3.2	Traffic Visualization	38
3.3	Anomaly Detection	46
3.4	Future Directions	52
4	Visualization and Implementation Choices	55
4.1	Dataset Characterization	55
4.2	Visual Variables	58
4.3	Navigation	60
4.4	Data Filters	62
4.5	Trajectories Animation	63
4.6	Analytics	65
4.7	Rendering Optimizations	67
4.8	Interface Evolution	70
4.9	Implementation	72
5	Anomalous Behavior Analysis	77
5.1	Intersections Based Behaviors	77
5.1.1	Data Processing Tasks	77
5.1.2	Visual Search Through a Magnified Fish-Eye Lens	80
5.1.3	High Density Areas with an Abnormality Level	86
5.1.4	Overtime Analysis with Small Multiples	92
5.2	Speed Outlier Behaviors	93
5.3	3D Explorations for Time Evolution	97

6	Case Studies	103
6.1	Hidden Intersection	103
6.2	Fishing Vessels Intersections	105
6.3	Low Speed Intersections	107
7	Conclusions and Future Work	111
	Appendices	119
A	Density-Based Clustering Results	120
A.1	Silhouette Coefficient Results	120
A.2	Number of Extracted Clusters	123

Acronyms

- AIS** Automatic Identification System. xi–xiii, 1–4, 28–33, 38–42, 44, 46–49, 52, 53, 55–58, 60–62, 64, 65, 67, 68, 72, 77, 78, 80, 84, 87, 88, 97, 98, 103, 105, 108, 111
- API** Application Programming Interface. 55, 56, 67, 68, 70, 73
- COG** Course Over Ground. 1, 31, 35, 38, 47, 56
- DBSCAN** Density Based Spatial Clustering of Applications with Noise. xi, 13, 14, 26, 27, 29, 31, 35, 36, 38, 47, 50, 52, 72, 73, 87
- DBSCANSD** Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction. 31, 36, 49, 52
- GUI** Graphical User Interface. 55, 70
- HDBSCAN** Hierarchical Density Based Spatial Clustering of Applications with Noise. 14, 26, 87, 88
- IMO** International Maritime Organization. 1
- KDD** Knowledge Discovery in Databases. 9
- KDE** Kernel Density Estimation. 94
- MMSI** Maritime Mobile Satellite Identity. 1, 47, 48, 56, 57, 66
- PDF** Probability Density Function. 94, 95, 97
- SOG** Speed Over Ground. 1, 31, 35, 38, 47, 49, 56
- TREAD** Traffic Route Extraction and Anomaly Detection. 29, 34, 52
- VTS** Vessel Traffic Services. 1

This page is intentionally left blank.

List of Figures

1.1	Example of how the Automatic Identification System (AIS) works.	2
1.2	Expected and real work plan of the 1st semester.	5
1.3	Expected and real work plan of the 2nd semester.	6
2.1	KDD process (Fayyad et al., 1996).	10
2.2	Classification example (Fayyad et al., 1996).	11
2.3	Regression example (Fayyad et al., 1996).	12
2.4	Clustering example (Fayyad et al., 1996).	12
2.5	Density Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm example (Lutins, 2017).	14
2.6	Levels of the visual variables (Bertin, 1983).	17
2.7	Examples of diagrams (Bertin, 1983).	18
2.8	Examples of networks (Bertin, 1983).	18
2.9	Example of a map (Bertin, 1983).	19
2.10	Examples of symbols (Bertin, 1983).	19
2.11	Groups and types of imposition (Bertin, 1983).	20
2.12	Example of a graphic with a high Lie Factor (Tufte, 1986).	21
2.13	Example of a graphic with a good Lie Factor (Tufte, 1986).	21
2.14	Example of a graphic with a lot of chart junk (Tufte, 1986).	22
2.15	Example of a graphic with the chart junk problem fixed (Tufte, 1986).	22
2.16	Example of small multiples (Tufte, 1986).	23
2.17	Example of a filtering operation by removing data records and dimensions (Ward et al., 2010). The original visualization is presented in the right image and the filtered visualization in the left image.	25
3.1	Visualization model of the experiment from the article of Riveiro and Falkman (2009).	39
3.2	Density visualization comparasion with and without speed (right and left images, respectively) from the article of Willems et al. (2009).	40
3.3	Index visualization of the MeiZhou Wan area from the article of Jiakai et al. (2012). The 3 potentially danger areas are identified on the map.	40
3.4	2D density visualization from the article of Gao and Shiotani (2013) for two hours of the day. High and low densities are represented by red and white, respectively.	41
3.5	3D visualization from the article of Gao and Shiotani (2013). The vessel being followed is in the green circle.	41
3.6	Pipeline of actions from the article of Fiorini et al. (2016).	42
3.7	Direct visualization (left) and summary visualization (right) of the experiment from the article of Chen et al. (2016).	43
4.1	Color palette for vessel type representation.	59

4.2	Usage example of the visual variable value.	59
4.3	Portugal, Spain and Africa boundaries for data contextualization.	61
4.4	An example of several selected trajectories.	62
4.5	The controls of the data filters.	63
4.6	An example of an animation with vessels sailing with different speeds.	65
4.7	An example of the statistics for the first day of data.	66
4.8	An example of the line distance tool.	67
4.9	Initial interface.	70
4.10	Final version of the interface.	72
4.11	General architecture of the implementation.	72
4.12	More important classes implemented on the AIS Viz Platform.	74
4.13	More important classes implemented on the data processing logic component.	75
4.14	Usage relation between the more important classes of both components.	76
5.1	Proposed approach for intersections' analysis.	78
5.2	Comparison between different mapping functions for the fish-eye effect using different focal length values and $r_x = 5$ for all the examples.	82
5.2	Comparison between different mapping functions for the fish-eye effect using different focal length values and $r_x = 5$ for all the examples.	83
5.3	Example of the fish-eye effect applied and the intersections identified.	84
5.4	Example of the fish-eye lens with different levels of magnification.	85
5.5	Example of the detail lens usage.	85
5.6	Color sub-palette for the representation of two vessels of the same type.	86
5.7	Proposed steps to identify areas of intersections with particular interest.	87
5.8	Example of the problem, raised by the usage of a rectangle to represent the cluster, when calculating its area.	90
5.9	Example of the problem, raised by the usage of a circle to represent the cluster, when calculating its area.	90
5.10	Example of a convex hull.	91
5.11	Color palette for the representation of the abnormality levels.	92
5.12	An example with two abnormal areas drawn on the platform.	92
5.13	An example of the small multiples approach with the monthly granularity.	93
5.14	An example of the small multiples approach with the daily granularity.	93
5.15	Distribution of the speed variable from each type of vessel.	95
5.15	Distribution of the speed variable from each type of vessel.	96
5.16	An example of fishing vessels sailing inside an illegal fishing area.	97
5.17	ECEF projection of 4 different vessels. Each image corresponds to the same data with different perspectives.	99
5.18	Projection using z axis to represent the date, applied to 4 different vessels. Each image corresponds to the same data with different perspectives.	100
5.19	Projection using z axis to represent the time, applied to 4 different vessels. Each image corresponds to the same data with different perspectives.	100
5.20	Projection representing the positions through a 24 hour clock shape. Each image corresponds to the same data with different perspectives.	102
6.1	Visualization of several AIS trajectories.	103
6.2	Identification and individual analysis of the intersection.	104
6.3	Animation of the trajectories from the intersection.	104
6.4	The small multiples approach for this intersection with the monthly gran- ularity.	105
6.5	Visualization of AIS trajectories.	105
6.6	Intersections and high density area of the fishing trajectories.	106

6.7	Detection and isolation of the first intersection.	106
6.8	Three frames of the animation from the isolated intersection.	107
6.9	The small multiples approach, with the daily granularity, for the intersection trajectories.	107
6.10	Visualization of low speed AIS trajectories.	108
6.11	Identification and isolation of the first intersection with low speed.	108
6.12	Animation of the trajectories from the intersection.	109

This page is intentionally left blank.

List of Tables

3.1	Summary of trajectory mining related work.	34
3.2	Summary of traffic visualization related work.	44
3.3	Summary of anomaly detection related work.	50
4.1	Number and percentage of positions by vessel type.	57
5.1	Silhouette coefficients averages for density-based clustering algorithms. . . .	88
A.1	Silhouette coefficients for density-based clustering algorithms with $MinPts = 25$	120
A.2	Silhouette coefficients for density-based clustering algorithms with $MinPts = 50$	121
A.3	Silhouette coefficients for density-based clustering algorithms with $MinPts = 75$	121
A.4	Silhouette coefficients for density-based clustering algorithms with $MinPts = 100$	122
A.5	Number of clusters extracted from density-based clustering algorithms with $MinPts = 25$	123
A.6	Number of clusters extracted from density-based clustering algorithms with $MinPts = 50$	124
A.7	Number of clusters extracted from density-based clustering algorithms with $MinPts = 75$	124
A.8	Number of clusters extracted from density-based clustering algorithms with $MinPts = 100$	125
A.9	Average number of clusters extracted from density-based clustering algorithms.	126

This page is intentionally left blank.

Chapter 1

Introduction

The Automatic Identification System (AIS) is an international standard for communication between vessels and terrestrial stations developed to improve maritime safety. It achieves this goal by helping vessels avoiding collisions and by assisting Vessel Traffic Services (VTS) in the control of the vessels sailing near the coast and specific ports (Tetreault, 2005). AIS equipment transmits periodical messages to other vessels and to terrestrial stations that are within its range through very high frequencies. These messages contain static information from a vessel like the antenna position, the Maritime Mobile Satellite Identity (MMSI) number, the vessel name and type, the International Maritime Organization (IMO) number, among other attributes. They also contain dynamic information like the current position of the vessel (in the form of a latitude and longitude), the time-stamp of the transmission, the Speed Over Ground (SOG), the Course Over Ground (COG), among other attributes (Bošnjak et al., 2012).

This system was initially developed for military usage but was later adopted by the IMO for civil traffic. This organization is responsible for the international maritime traffic regulations and in 2004 made the AIS system mandatory for vessels with a volume of 300GT¹ or more operating in international waters, for vessels with a volume of 500GT or more operating in local waters and for all passenger vessels (Bošnjak et al., 2012). AIS messages are transmitted through very high frequency between vessels and terrestrial stations, which are controlled by the VTS. However, when considering the distances on the sea, a vessel may be in a position to far from other vessel or a station, and in those scenarios the messages would not be received by these parties of interest. To fix this issue, the AIS equipment installed in each vessel is able of repeating the messages received from other vessels, serving as a middleman on the transmission protocol. This scenario is presented on Figure 1.1 where, for example, the messages from A would not reach D without B working as the middleman.

AIS data contains all the necessary information for mapping the trajectories followed by each vessel and the general maritime traffic of any sea, and for that reason it has been the

¹Gross Tonnage is a unit for measuring the volume of a vessel internal spaces.

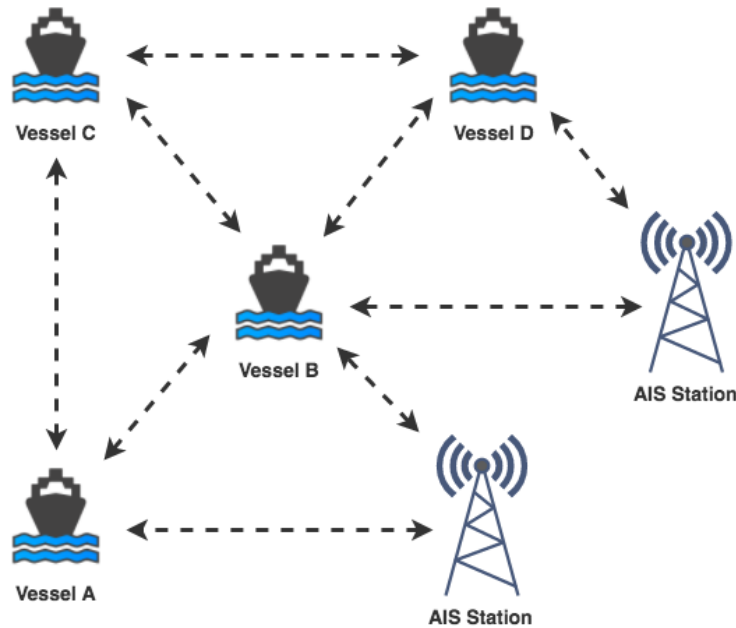


Figure 1.1: Example of how the AIS works.

target of several studies that try to apply statistical and data mining techniques to it. The majority of these studies are focused particularly in traffic analysis and forecasting (Mazzarella et al. (2015), Sang et al. (2016)), pollution control (Busler et al. (2015), Liu et al. (2015)), fusion of different maritime data sources (Xu et al. (2015), Yang et al. (2013)) or identification of vessels' anomalous behaviors (Handayani et al. (2013), Soleimani et al. (2015)). Concerning the data visualization field (the central theme of this thesis), published works have been focused in new representations of the traffic situation from specific areas of interest (Willems et al. (2009), Gao and Shiotani (2013), Chen et al. (2016)), with minor or none emphasis on anomalous behaviors, which constitutes a direction yet to be explored.

1.1 Contextualization

Portugal has one of the biggest coastlines in Europe and an exclusive economic zone with 1.727.408 km² of extension (Jacobs, 2016). This makes the Portuguese sea a mandatory passage point not only for a huge number of vessels that transport cargo between countries but also for fishing vessels that operate in the same waters. Considering the described scenario, with so many vessels in such a big maritime area the existence of illegal activities tends to be very high if a rigorous control is not enforced. The Portuguese navy is responsible for this control and, among other tools, uses a software developed in a partnership with the company Critical Software, called *Oversee*², to perform this control. This tool works with AIS data that is periodically received by aggregating the reported positions into trajectories and allowing a simple visualization of the maritime traffic in the area of

²Available at <https://oversee.criticalsoftware.com/en/home>

interest. The data is presented to the navy operators in a way that they are able to control the coastline in real time. With this approach the navy is able to detect problems very quickly and act accordingly. *Oversee* helps the navy particularly in three fields: search and rescue, law enforcement and environment protection (Software, 2017).

Focusing on the law enforcement field, detecting abnormal patterns in the vessels trajectories is a major feature to detect illegal behavior. Currently this is a grey area in the *Oversee* software. Critical Software has been working in several machine learning approaches to implement this feature, but none has reached production. Regarding the anomalous behaviors, there are a set of common ones that were identified by the domain experts, which are:

- Drifting, which could be an indicator of a failure that was not reported;
- Sailing in low speed, which could also be an indicator of a failure. If maintained for a long time, this could even be an indicator of illegal fishing if the vessel is of the fishing type and is on a forbidden zone³;
- A vessel sailing in a different trajectory from the expected one, which could be an indicator of an illegal action;
- Fishing vessels stopping or sailing in low speed on a zone where fishing is forbidden, which could be an indicator of illegal fishing;
- Two vessels sailing very close to each other, which could be an indicator that an illegal trade is happening;
- Two vessels crossing trajectories, especially if one of them goes from Portugal to the intersection zone and comes back after a short period of time, which could be an indicator that this vessel went to the zone to pick up some illegal goods left by the other one.

Although some machine learning approaches have been introduced to address this problem, there is still work to be done in this area. Besides, new approaches and scientific fields can be explored, like the data visualization one.

1.2 Research Goals

The main goal of this thesis was to study and develop data visualization approaches, assisted by data analysis techniques, to detect anomalous behaviors, on the Portuguese maritime area, using AIS data. These approaches were implemented on a visualization platform that can be used by a domain expert. Having in mind this main goal, the following intermediate goals were defined:

³According to the Portuguese laws it is illegal to fish in the first 6 nautical miles of the sea and in specific zones after that point.

- Study state-of-the-art concepts and models of data visualization and data mining techniques, specially in the context of AIS, including (but not limited to) trajectory mining, traffic visualization and anomaly detection;
- Define and implement the visualization choices and data processing tasks, which are necessary for an analysis of the data without focusing on any type of anomalous behaviors;
- Define and implement the visualization and data analysis strategies to address each category of anomalous behaviors, exploring the 2D and 3D plane;
- Test and validate the implemented strategies.

1.3 Work Plan

For each semester of this work, the considered tasks and respective scheduling are presented through gantt charts on the Figures 1.2 and 1.3 for the 1st and 2nd semester, respectively. For each task the expected effort and schedule are represented with grey bars, and the real values are represented with blue bars.

Regarding the 2nd semester, the gantt chart presents some deviations between the expected and the real time spent with each task. These deviations are justified by an additional effort related with the implementation tasks. The platform, with the respective visual strategies and data analysis, was implemented from the scratch and several problems that were not expected arose, which required more effort to be fixed. The necessary adjustments were made to the remaining tasks in order to finish the thesis within the required time.

In order to properly manage the time spent on each task, an agile methodology was used during the 2nd semester. The usage of this type of methodology improved the control of the time spent in each task, ensuring that even small deviations were detected in an early stage, which allowed the application of corrective measures without consequences (or with minimal ones) in the results of the thesis. Also, the iterative and incremental approach followed by agile methodologies allowed a more constant validation of the work developed, which was important to understand gradually if the selected approach for each task of the work was presenting the expected results.

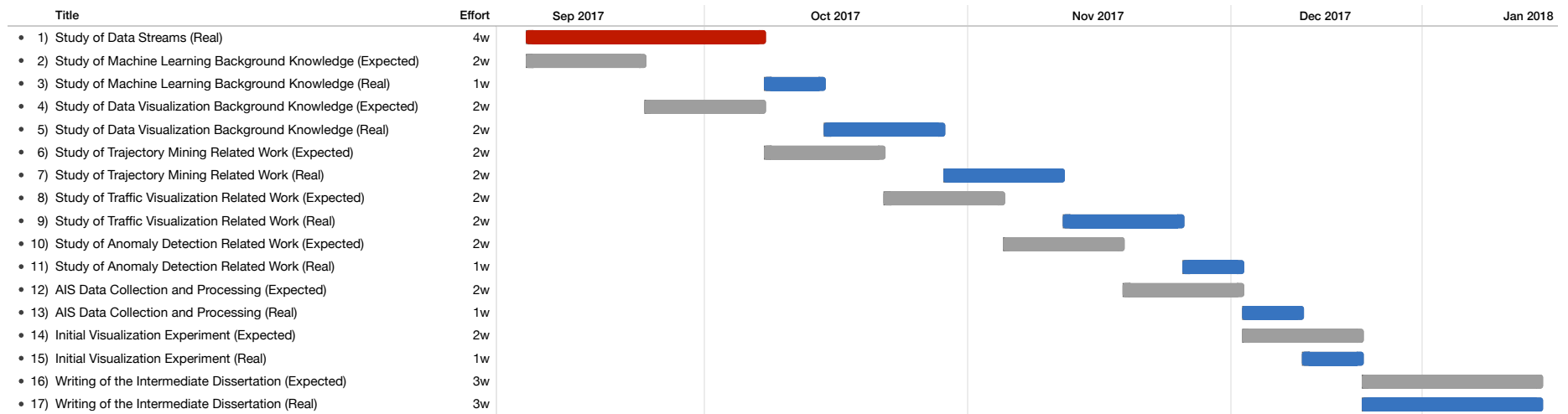


Figure 1.2: Expected and real work plan of the 1st semester.

6

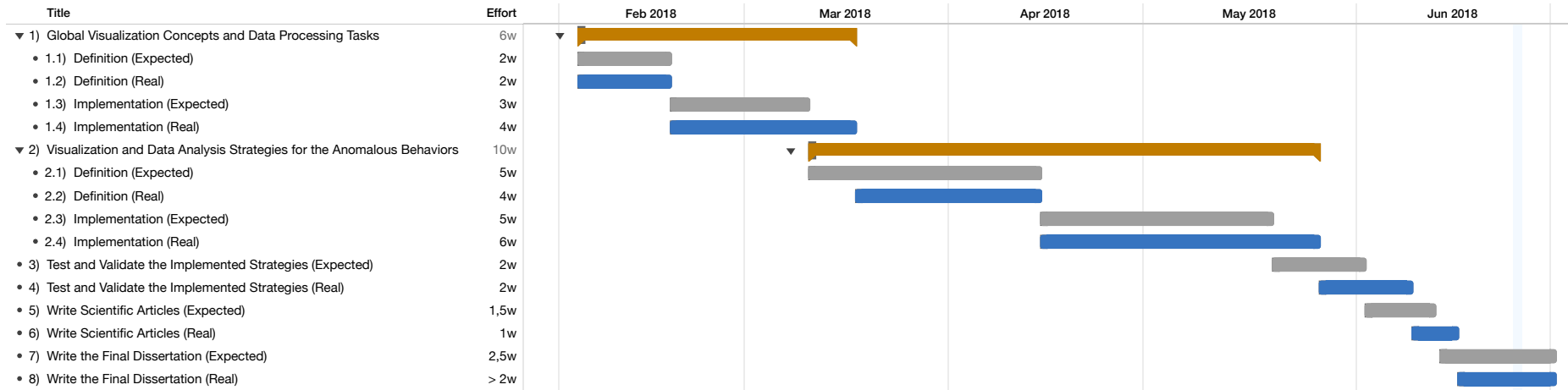


Figure 1.3: Expected and real work plan of the 2nd semester.

1.4 Research Contributions

From the work presented on this thesis, two major research contributions were made:

- Ricardo Cardoso Pereira, Pedro Henriques Abreu e Penousal Machado. *A Survey of AIS Data Analysis Techniques: Trajectory Mining, Traffic Visualization and Anomaly Detection* (2018). *IEEE Transactions on Intelligent Transportation Systems* (the article is awaiting review);
- Ricardo Cardoso Pereira, Pedro Henriques Abreu e Penousal Machado. *AIS Intersections Analysis Through Data Visualization: a Visual Search Approach Using a Magnified Fish-Eye Lens* (2018). *Seventeenth International Symposium on Intelligent Data Analysis (IDA 2018)* (the article is awaiting review).

1.5 Document Structure

The remaining of this document is organized in the following way:

- Background knowledge, where the theoretical foundations necessary for this thesis are exposed, namely the necessary fundamentals of data mining (section 2.1) and data visualization (section 2.2);
- Related work, where some of the recent work published in the fields approached by this thesis are introduced, which are trajectory mining (section 3.1), traffic visualization (section 3.2) and anomaly detection in vessels traffic (section 3.3), and finally some future directions are identified (section 3.4);
- Visualization and implementation choices, where all the base concepts for the visualization and for the data processing aspects, that are necessary for a general analysis and for the platform to function, are defined;
- Anomalous behavior analysis, where the data visualization strategies, assisted by data analysis, developed to analyze and detect the different categories of anomalous behaviors are defined. This chapter also includes the explorations made of the 3D plane to emphasize the time variable;
- Validation of the implemented strategies through three case studies of real scenarios;
- Conclusions, where an evaluation of the past and future work is presented.

This page is intentionally left blank.

Chapter 2

Background Knowledge

This section presents the theoretical foundations of the fields of study approached by this thesis. These fields are data mining process (section 2.1) and data visualization (section 2.2).

2.1 Data Mining

Knowledge Discovery in Databases (KDD) (Fayyad et al., 1996) is a process focused on the development of techniques and methods to extract knowledge from data. This process mainly tries to map raw data, that usually is too voluminous to be processed in its original condition, into models capable of approximating a representation of the entire data.

The KDD is an iterative process and requires several steps, some of them that depend on user made decisions (Fayyad et al., 1996):

- The first one is to understand the domain of the data that is being treated, usually studying all the background knowledge of this domain, and to identify the goals that the expected outcome of the process is trying to achieve;
- The second one is to select a dataset containing all the necessary variables and records where the process is going to be applied;
- The third one is to clean and preprocess the data, where the more common operations are noise removal, defining how to model and treat noise, deciding what to do with missing data fields and deciding how to deal with time-sequence data;
- The fourth one is to reduce and project data, through operations like feature selection and feature extraction, where the most relevant features to represent the data and achieve the goals are identified and filtered, reducing the dimensionality of the dataset;
- The fifth one is to map the goals of the first step to data-mining methods (classification, clustering, regression, and others);

- The sixth one is to make an exploratory analysis and define a hypothesis to test, which in fact is the selection of the specific data mining algorithms to be applied, with the respective parameter tuning (when required), taking in consideration that the selected algorithms must create a model that fulfills the expected goals;
- The seventh is to apply the data mining algorithms in order to search and identify the hidden patterns of interest in the data;
- The eighth is to interpret the results obtained from the previous step, very often through the visualization of the extracted patterns and models, and to decide if the results are satisfying or if the steps should be repeated with different decisions in between to achieve different (and eventually better) results;
- The ninth and final one is to apply the discovered knowledge, using it directly or integrating it in another system, or even simply publishing it to be used by parties of interest.

The basic flow of the described process is shown in Figure 2.1. To be notice that iteration can happen in between any of the steps until the outcome of the step is the expected one.

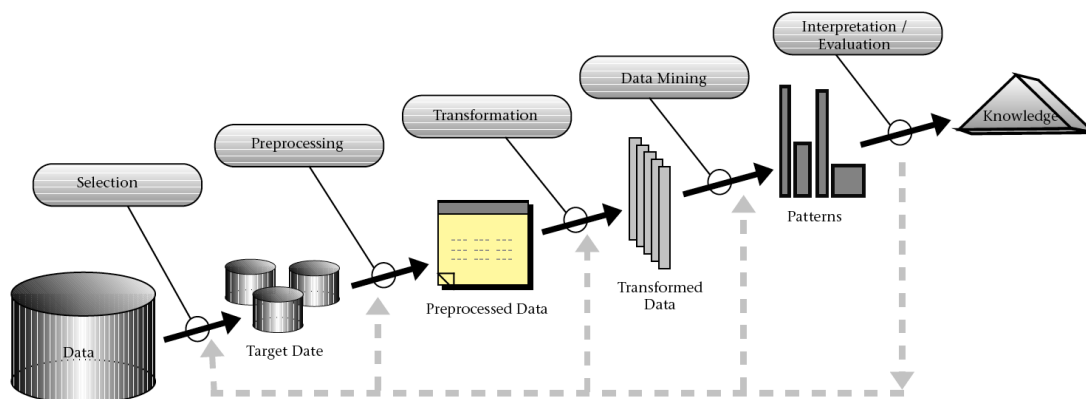


Figure 2.1: KDD process (Fayyad et al., 1996).

Focusing on the data mining step, and particularly on the more important available methods, they can be assigned to two different categories related with the expected goals. These two categories are prediction and description. Prediction is the action of predict unknown of future values for specific variables using all or some of the available variables and data. Description is the action of discovering interpretable patterns from the existing data. It is common to find models that fit in both categories because sometimes both actions are required to achieve specific goals.

To achieve prediction and description several data mining methods exist that can be grouped in a few approaches. The more important approaches are described below (Fayyad

et al., 1996). Notice that for demonstration proposes a simple graphic example from Fayyad et al. (1996) was used, where each point represents a person who has received a loan from a bank, being the horizontal axis the incoming and the vertical axis the debt of each person. Also, the x's represent people who failed their payments and the o's people who are in compliance with their obligations. Enumerating the approaches:

- Classification, which consists in creating a function that is able to classify a given data record into a specific class from an existing set. This type of approach fits into the prediction category because is trying to predict a type (label) for a data item based on known information. An example of a classification can be seen in Figure 2.2, where people are divided into two classes (bad status and good status). Notice that the obtained classification model does not fit perfectly all the points because a single linear boundary is not enough for the dispersion of the given example;

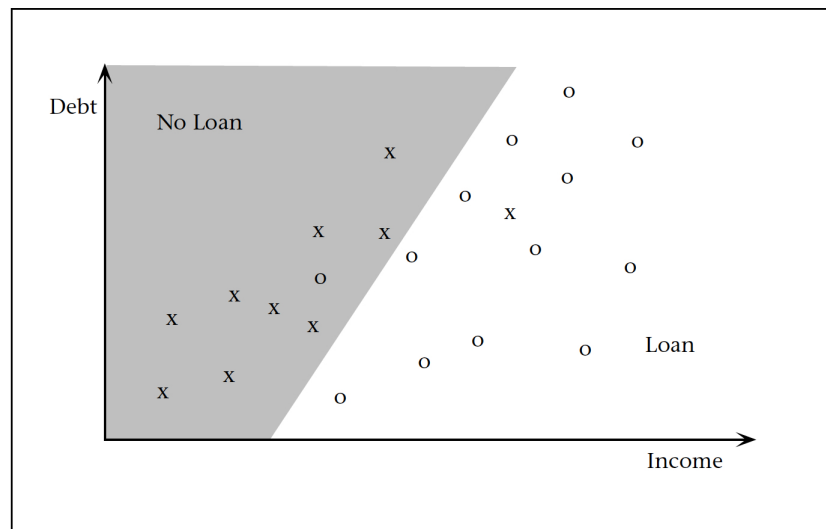


Figure 2.2: Classification example (Fayyad et al., 1996).

- Regression, which consists in creating a function that is able to map a data record into a real number variable. The process is actually very similar to classification, being the major different the output of the method because classification method's output is a class from a predefined set and regression method's output is a real number. This approach also fits into the prediction category because it is trying to predict a number for a data item. An example of a regression can be seen in Figure 2.3, where a simple linear regression is presented by fitting the total debt as a linear function of the income. To be notice that the fitting does not model the data very well because the correlation between the two variables is very weak;
- Clustering, which consists in identifying a finite set of categories (commonly called clusters) that are able to describe the data. Each category should contain data records with some level of similarity in specific features. Depending on the used algorithms, categories can be mutually exclusive or overlapped. This approach fits

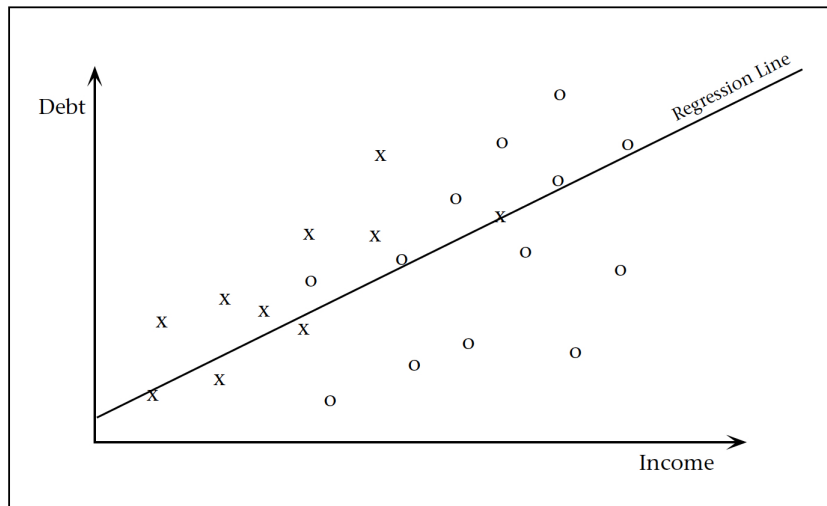


Figure 2.3: Regression example (Fayyad et al., 1996).

into the description category because it is trying to describe (find) data records with similar features into categories. An example of clustering can be seen in Figure 2.4, where all the people are clustered into 3 categories (overlapping is allowed and some people belong to more than one cluster). Notice that the original labels (x and o) were replaced by +.

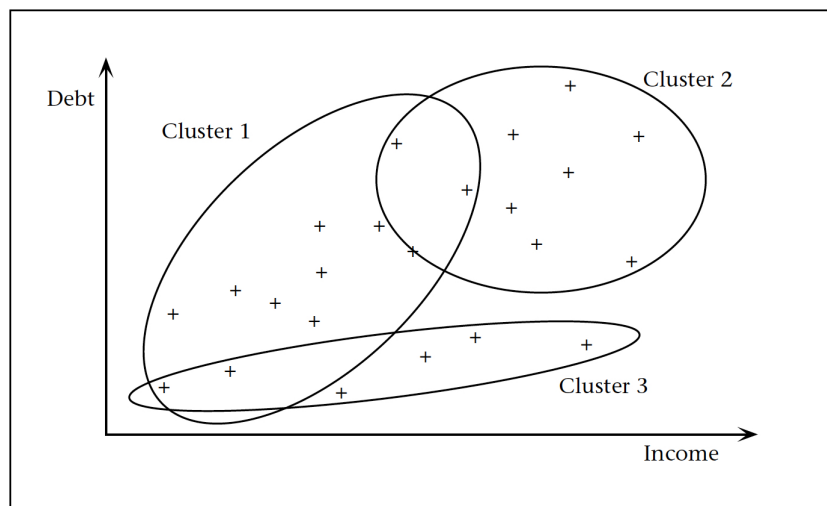


Figure 2.4: Clustering example (Fayyad et al., 1996).

Focusing on clustering, there are 3 types of algorithms that use this approach, namely (Ester et al., 1996):

- Partitioning algorithms, where k clusters are created (k is typically a given parameter) and each cluster is commonly represented by its gravity center or by an object

close to its center. Each new object is assigned to the closest cluster, which normally requires that each cluster representation is updated after a new assignment;

- Hierarchical algorithms, where the clusters are build as a hierarchy commonly represented by a tree, and two types of approach for the hierarchy exist:
 - Agglomerative approach, where each object starts in its own cluster (each one is a leave of the tree) and the clusters are merged while going up in the hierarchy until arriving at the root node;
 - Divisive approach, where all objects start in one cluster (the root node of the tree) and new clusters are created by splitting the existing one(s) going down in the hierarchy until arriving at the leaves nodes.

This type of algorithms require a termination condition to define when to stop merging or devising, which normally is a minimum distance between all the clusters, but the number of clusters is not a required parameter;

- Density-based algorithms, where each cluster represents an area with a higher density of objects compared to others, which means that each cluster is created by determining some kind of distance between objects and grouping the ones where this distance is small. This type of algorithm does not require the number of clusters as a parameter.

Density-based clustering has been highly used in different contexts and the most common algorithm that follows this approach is the Density Based Spatial Clustering of Applications with Noise (DBSCAN). Given a set of points (for this algorithm each object is treated as a point in the space), this algorithm starts by labeling each of the points into one of three categories (Ester et al., 1996):

- Core point, meaning that this point has a set of points within a given distance called *Eps* (this distance is a parameter of the algorithm) and the cardinality of this set is greater than a given threshold called *MinPts* (this threshold is also a parameter of the algorithm). Notice that the points in this set are said to be in the neighborhood of the core point and are density-reachable from it (but the opposite may not be true). Also, the distance function commonly used is the Euclidean distance, but any function is supported;
- Border point, meaning that this point did not meet the criteria for becoming a core point but is in the neighborhood of at least one core point;
- Noise point, meaning that this point did not meet the criteria for becoming neither a core or border point.

Then, the algorithm picks a random core point and creates a cluster containing all the core and border points in its neighborhood and, for each core point in this neighborhood,

it expands the cluster by doing the same described process in a recursive way. A core point can only be "expanded" once and the algorithm stops when all the core points were "expanded", returning the discovered clusters.

This algorithm has the advantages of not requiring the number of clusters to be passed as a parameter, but the required parameters (*Eps* and *MinPts*) are very sensitive, which means that minor variations on these values can generate very different results. The authors of the algorithm (Ester et al., 1996) proposed the usage of a k-distance graph to estimate the better values for these parameters, but many approaches can be used to perform this fine tuning.

Figure 2.5 presents a simple example that illustrates all the algorithm process described above.

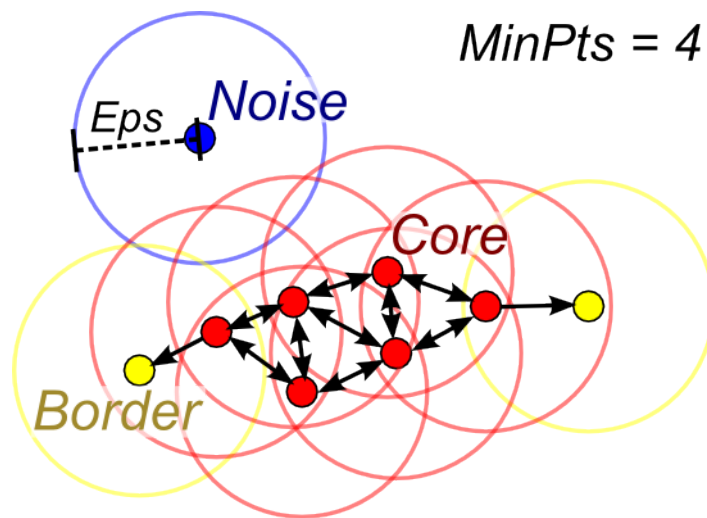


Figure 2.5: DBSCAN algorithm example (Lutins, 2017).

Regarding the sensitivity problems of the DBSCAN parameters, a new approach called Hierarchical Density Based Spatial Clustering of Applications with Noise (HDBSCAN) (Campello et al., 2013) was introduced. The idea of this algorithm is to use an hierarchical approach to find the clusters with a higher density, using a strategy similar to the DBSCAN for the density evaluation but varying different epsilon values, which removes the epsilon from the parameters list and also allows the creation of clusters with different densities. It then uses the *MinPts* parameter (the only required) to mark and merge the clusters that do not have this minimum size into their parents, and uses this information to calculate the stability of each final cluster. The final cluster extraction to flat mode uses this stability to decide which clusters to keep from the hierarchy between the parent and the descendants.

In order to evaluate the quality of the clusters obtained from any clustering algorithm several metrics exist, one being the Silhouette Coefficient (Rousseeuw, 1987). This metric indicates the cohesion of a point to its cluster by measuring how well it fits when compared to the remaining points of the same cluster and to the 2nd closest one. The resulting value is between $[-1, 1]$, where 1 indicates that the position fits perfectly into the cluster and -1

the opposite. Considering x_1 as the average distance between point p and the remaining ones from its cluster and x_2 as the average distance between the same point and the ones from the 2nd closest cluster, the formula to calculate the coefficient is the one presented on Equation 2.1. To evaluate an entire cluster through this metric the average coefficient from all points that are within it is considered.

$$SC_p = \frac{x_2 - x_1}{\max(x_1, x_2)} \quad (2.1)$$

2.2 Data Visualization

Delivering information for the masses is not always an easy task. People are used to communicate with words, but words are not an efficient way of presenting big quantities of data. To understand the meaning of data a visual representation of the information is required, otherwise people will not understand the message. In this section the principles of information visualization introduced by Jacques Bertin are presented (section 2.2.1), but also some of the more important principles of Edward Tufte are introduced (section 2.2.2). These authors are two of the main references in this field, being their principles the foundations of data visualization. The general principles of interactive visualization are also presented (section 2.2.3) and, finally, the common approaches for validating visualizations are described (section 2.2.4).

2.2.1 Jacques Bertin's Principles

Jacques Bertin was a French cartographer and one of the first researchers to work on this field. He proposed through his book "Semiology of Graphics" (Bertin, 1983) a set of visual (or retinal) variables that combined together are able to present information in an intuitive way. These variables can be represented in a normal two-dimensional plane through the visual marks, which are points, lines and areas. Points can only represent a location on the plane because they don't have length or area. Lines can represent connections and trajectories because they have length but don't have area. Areas can represent anything on the plane that has a measurable size. Bertin introduces the following 7 visual variables:

- Position, which reflects a change on the location of the mark on the plane through the increase or decrease of the x or y values;
- Size, which reflects a change in the height or area of a mark, or eventually on the number of repetitions of a mark;
- Value, which reflects a change of the color steps from white to black;
- Texture, which reflects a change in the fineness or coarseness of an area, originating a new pattern associated with a different value;

- Color, which reflects a change on the hue of a mark associated with a new value;
- Orientation, which reflects a change on the alignment of a mark from vertical to horizontal;
- Shape, which reflects a change on the actual representation of a mark.

Each of these variables have a level associated depending on the type of information that each one is able to represent. Bertin defines 4 main characteristics that each variable may or may not have, being these characteristics what defines the variable level. A visual variable can be:

- Selective, which is the ability to immediately identify all the marks that belong to the same category of the given variable;
- Associative, which is the ability to immediately group all the marks that are somehow distinguished (associated) by the given variable;
- Ordered, which is the ability to assign an immediate and universal order to the values of the given variable;
- Quantitative, which is the ability to assign a numerical value to the "distance" between to values of the given variable.

Bertin also introduces a fifth characteristic, the length, that defines the number of possible values that a visual variable can take. In several scenarios this characteristic has no value because there are an infinite number of values for a variable. An example could be a shape variable, that theoretically can assume an infinite number of forms.

The level of each visual variable is presented in Figure 2.6. The variables are displayed as rows and the characteristics as columns, and for each valid characteristic of each variable is presented an example in the respective cell. The position variable is not displayed because it is considered a special variable, meaning that it is always used and is valid for all characteristics.

The main conclusions about the valid characteristics of each visual variable are:

- The association characteristic is valid for all variables, but for size and value it has the opposite effect, which means that these variables are dissociative;
- The selection characteristic is valid only for size, value, texture, color and orientation (for the last one it is only valid when the visual mark is a point or a line);
- The order characteristic is valid only for size, value and texture;
- The quantity characteristic is valid only for size.

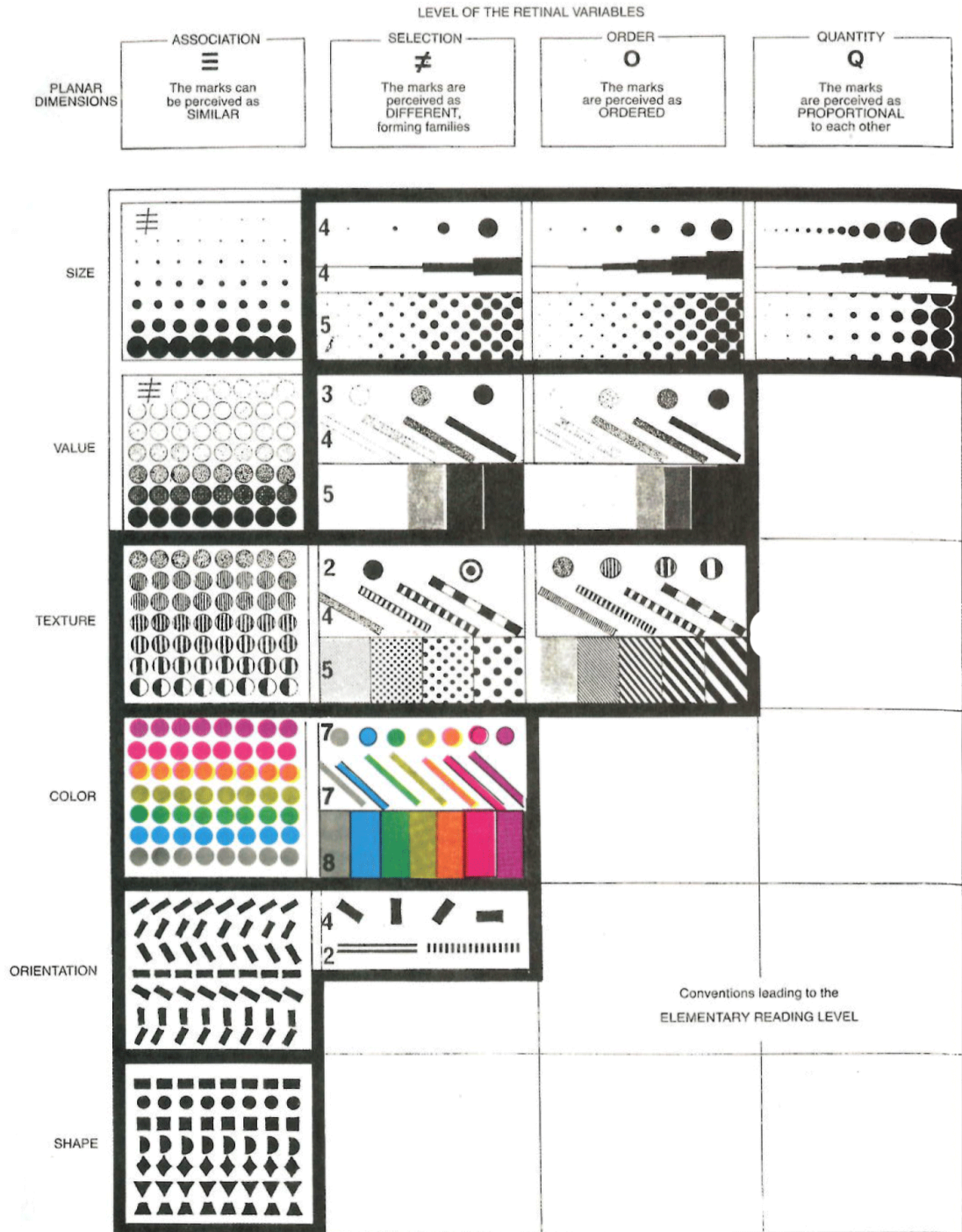


Figure 2.6: Levels of the visual variables (Bertin, 1983).

Bertin also introduces the concept of "imposition" as the utilization of the plane and divides this utilization into 4 groups: diagrams, networks, maps and symbols.

Diagrams are created when correspondences in the plane can be established between all the values (Bertin calls them divisions) of one component and all the values of another component (Bertin introduces component as a variation concept, which means some con-

cept where the values change over time). An example is a simple scenario where for each date a correspondent price exists, and the representation of these correspondences creates a diagram. The example is presented in Figure 2.7.

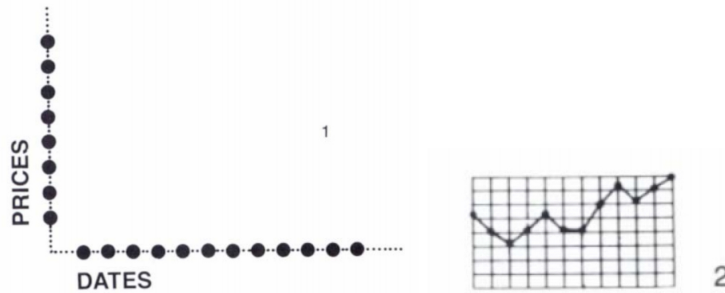


Figure 2.7: Examples of diagrams (Bertin, 1983).

Networks are created when correspondences in the plane can be established between all the values of the same component. An example is a conversation between several individuals, where the component is the different individuals. Some possible networks for this example are presented in Figure 2.8.

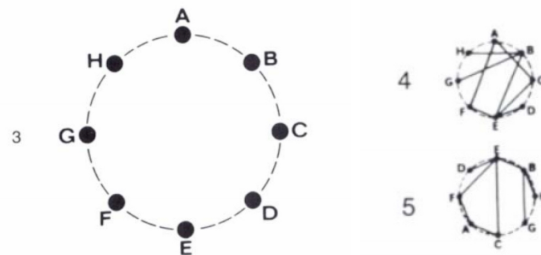


Figure 2.8: Examples of networks (Bertin, 1983).

Maps are created when the correspondences in the plane can be established in the same way as a network, but can be presented according to a geographic order. An example is a representation of the highways of an area where the correspondences are established between several locations distributed in a geographic order. The example is presented in Figure 2.9.

Symbols are created when the correspondence is not established in the plane, but instead is established with the reader through the universal meaning of the presented element. Some examples are universal signs associated with road traffic, agriculture, geology, industry, among others. Some of these examples are presented in Figure 2.10.

These groups of imposition can be drawn in the plane according to a specific arrangement or according to a construction that can be rectilinear, circular, orthogonal or polar. This is called the type of imposition. Examples of arrangements and constructions of each type for each group of imposition are presented in Figure 2.11. The groups are displayed

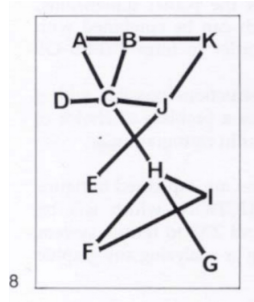


Figure 2.9: Example of a map (Bertin, 1983).

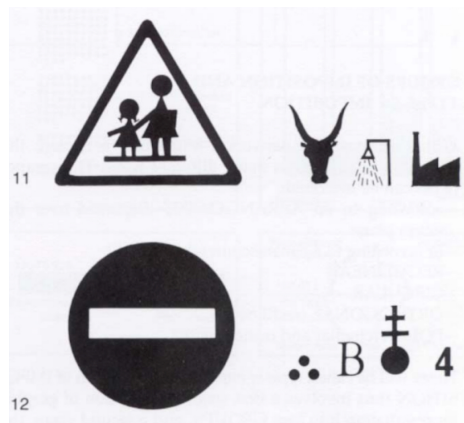


Figure 2.10: Examples of symbols (Bertin, 1983).

as rows and the types as columns, and in each cell one or more examples of the type of imposition applied to the group is presented. An empty cell means that the type of that column does not apply to the respective group.

2.2.2 Edward Tufte's Principles

Edward Tufte, one of the major contributors for this field after Bertin, introduced several principles in his book "The Visual Display of Quantitative Information" (Tufte, 1986) with the propose of improving the quality of graphics that represent data. Quoting two of these principles "The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities represented." and "Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data.". Considering the first principle, Tufte introduces a simple measure called Lie Factor which is calculated with the formula $\text{Lie Factor} = \frac{\text{size of effect shown in graphic}}{\text{size of effect in data}}$. When applying the Lie Factor to a graphic, if the value is equal to 1 then the graphic is representing the numerical quantities in a proportional way, but if the value is less than 0.95 or greater than 1.05 then the representation is distorting the real values. Tufte presents the example from Figure 2.12, a graphic published by the New York Times in




















IMPOSITION		TYPES OF IMPOSITION				
		ARRANGEMENT	RECTILINEAR	CIRCULAR	ORTHOGONAL	POLAR
GROUPS OF IMPOSITION	DIAGRAMS		 	 	 	 
	NETWORKS	 	 	 	 	
	MAPS	 GEO 				
	SYMBOLS					

Figure 2.11: Groups and types of imposition (Bertin, 1983).

1978 where a series of fuel economy standards to be met by automobile manufactures are presented, starting with 18 miles per gallon in 1978 and moving step by step until 27.5 miles per gallon in 1985. Looking to the values in percentage, from 1978 to 1985 an increase of 53% is presented, and this is considered the size of the effect in the data. To obtain the size of the effect in the graphic the relative length of the two lines associated with the two years is calculated, which results in a value of 783%. Calculating the Lie Factor with these two values a value of 14.8 is obtained. This is clearly a factor too big, which means the data is represented in a disproportional and deceitful way. A much simpler graphic that displays the same data but with a good lie factor is present in Figure 2.13.

Another major principle from Tufte is "Above all else show the data.". This means that a graphic should use the majority of its space and ink to effectively display data. To test this principle a measure called Data-ink ratio is introduced, which is calculated with the formula $\text{Data-ink ratio} = \frac{\text{data-ink}}{\text{total ink used to print the graphic}}$. This is actually a simple ratio that

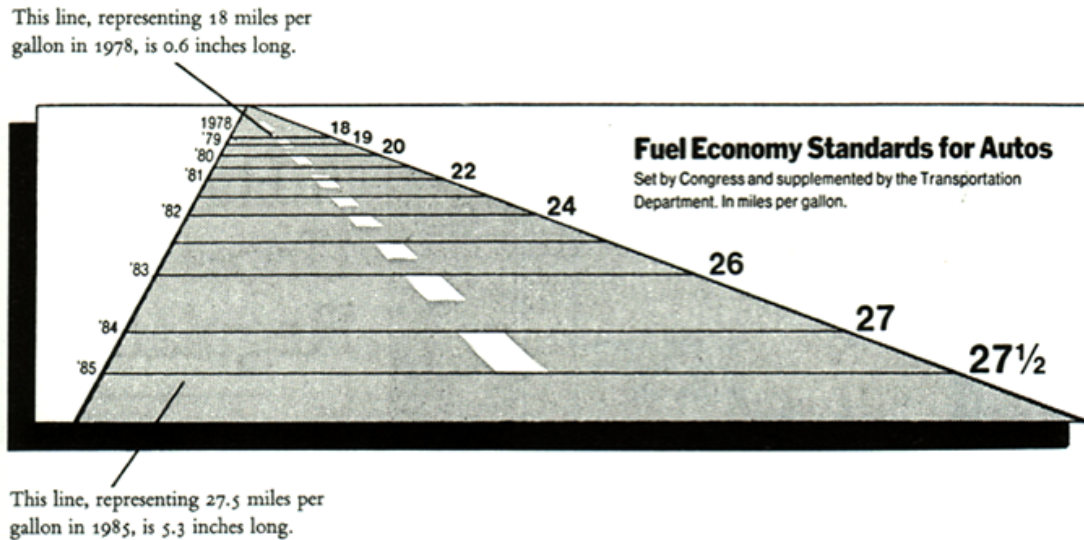


Figure 2.12: Example of a graphic with a high Lie Factor (Tufte, 1986).

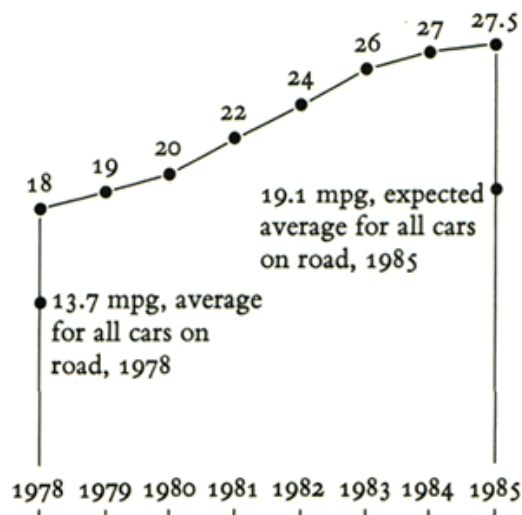


Figure 2.13: Example of a graphic with a good Lie Factor (Tufte, 1986).

measures in an objective way how much of the graphic can be erased without losing data information. An ideal graphic would have a data-ink ratio of 1, which means that the entire graphic is filled with non-redundant information. Sometimes a ratio of 1 is impossible to achieve, mainly because some visual elements auxiliary to the data are required to make the graphic comprehensible. The most common example is the grid. A lot of graphics require a grid for scale purposes and it can not be deleted. But also a lot of times this grid is a powerful source of distraction to the reader. Tufte calls this phenomenon the chart junk, and it is everything that distracts the reader from the real message of the graphic. All the chart junk should be deleted or, at least, minimized. Considering the grid, when

it is not possible to eliminate it then some simple actions can help a lot, like reducing the thickness of the grid lines and using the color grey instead of black for those lines. The example of Figure 2.14 shows a graphic with a lot of chart junk. The same example is presented in Figure 2.15 with the junk problem fixed through the application of the actions mentioned above.

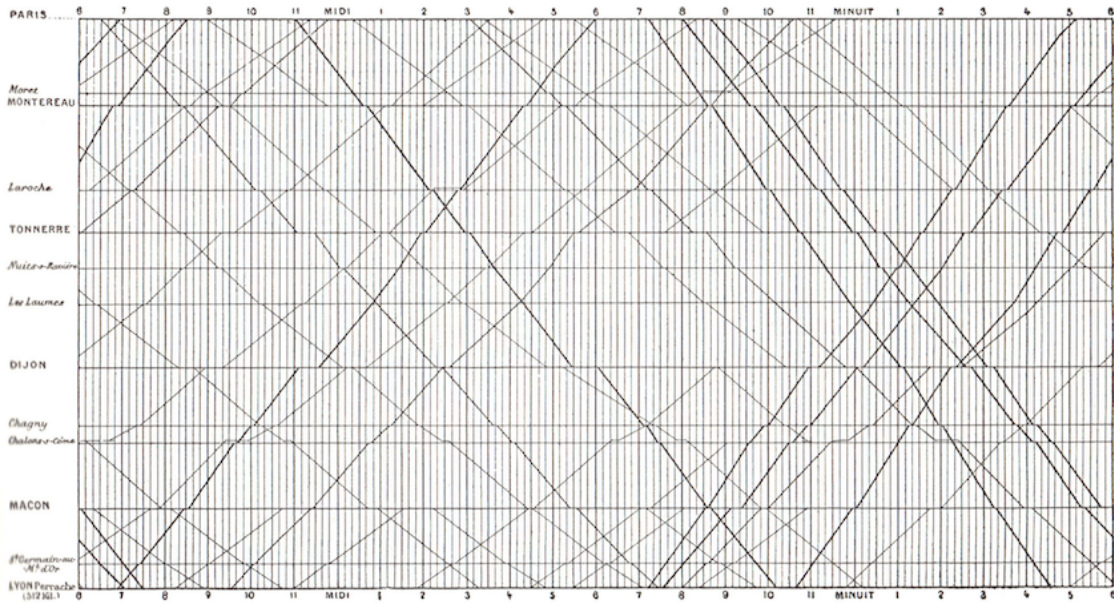


Figure 2.14: Example of a graphic with a lot of chart junk (Tufte, 1986).

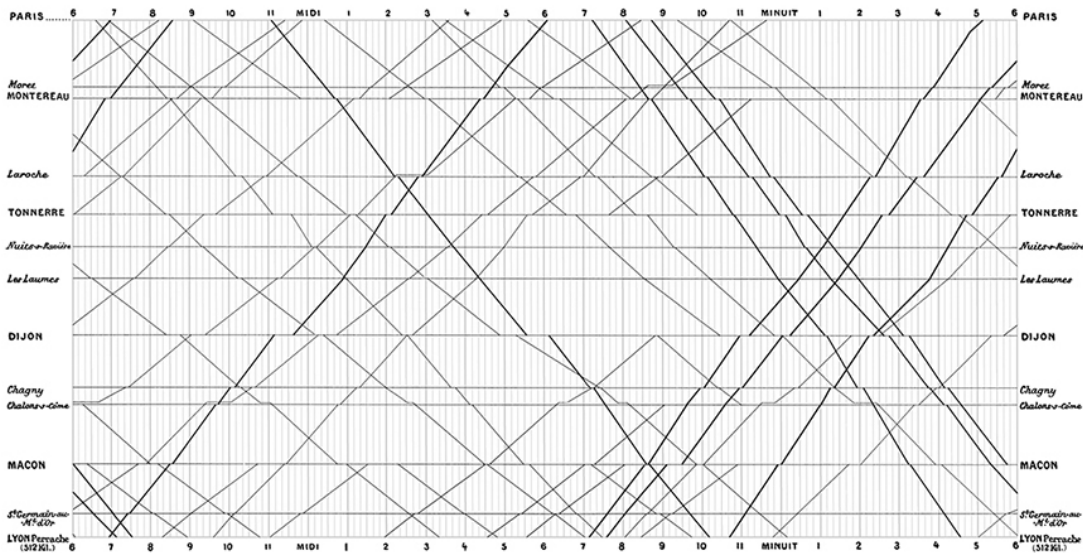


Figure 2.15: Example of a graphic with the chart junk problem fixed (Tufte, 1986).

Within the visual concepts proposed by Tufte, one that is widely used in modern visualization is the small multiples. He describes this concept as series of images where all

variables values are indexed by the changes from a specific variable through the sequence. In terms of looks Tufte compares this concept to a set of frames from a movie, because they are in fact a set of sequential images. The presented example, visible on Figure 2.16, shows the average distribution of the reactive hydrocarbon emissions from 23 hours of Los Angeles. To display the levels of emissions the visual variable color is used through a scale. The independent variable that changes over the multiples is the hour of the day.

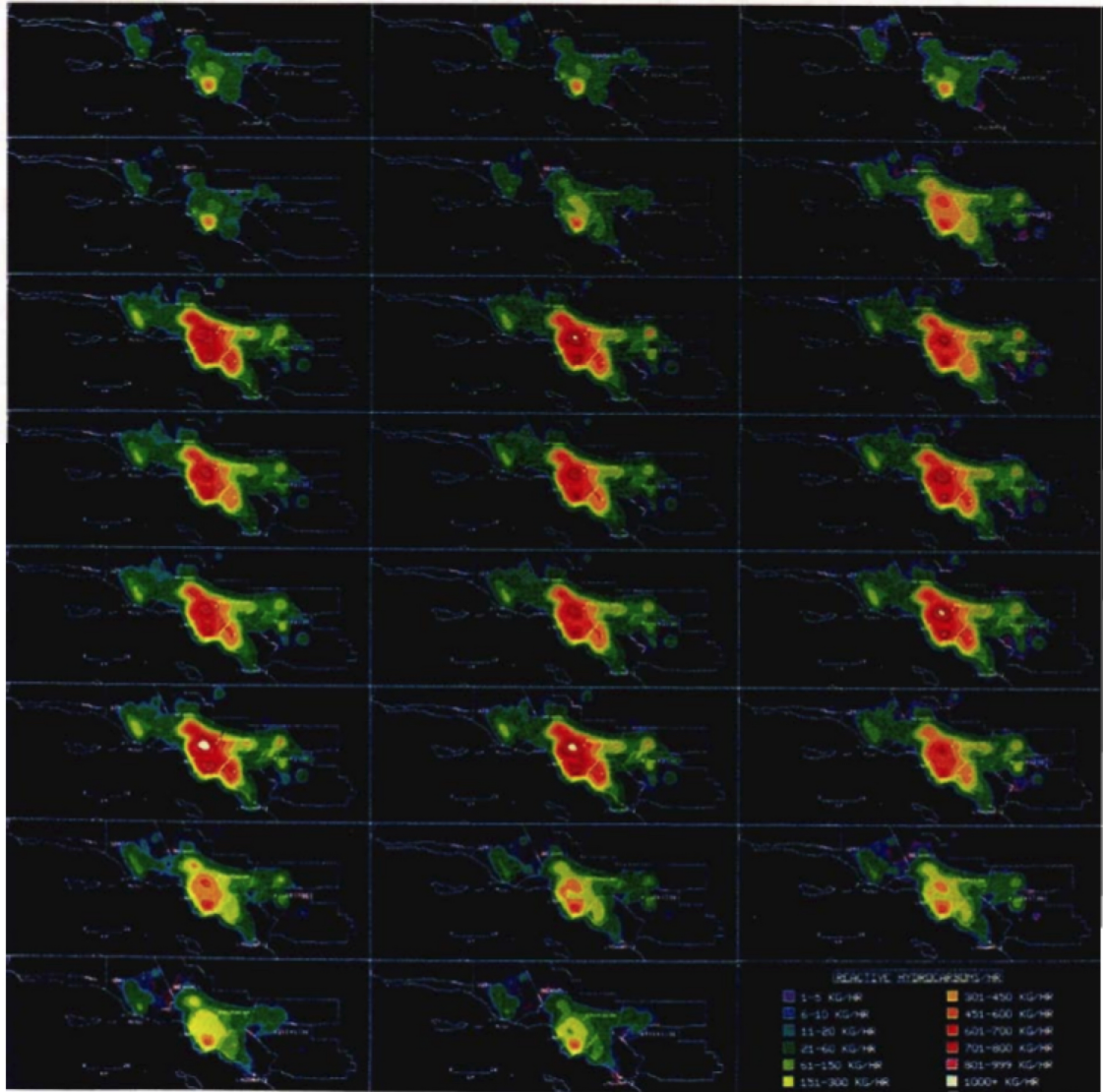


Figure 2.16: Example of small multiples (Tufte, 1986).

2.2.3 Interactive Visualization

With the digital era and the massification of the computer some new concepts appeared in the context of data visualization, namely the concept of interactive visualization. Until there, data was always displayed in a static way, typically through images. But with the computer a new kind of interaction was available, which allowed the creation of more

dynamic ways of displaying the data. This dynamic is obtained through the interaction of the user that effectively adjusts the visualization to his needs.

One of the novelties introduced with interactive visualization was a new visual variable, the motion (Ward et al., 2010). Motion is actually a variable that is always associated with one of the other variables because it simply introduces a way to display changes over time on the values of the other variable. Typically these changes can be obtained through variations of the speed or direction, being these variations what triggers the idea of motion to the user and also the change of values in the other variable.

Another novelty is the ability that the user has to modify the visualization when interacting with it. Therefore, several types of interaction techniques (also called interaction operators) have been introduced (Ward et al., 2010), namely:

- Navigation, which gives the user the possibility of changing the position of the camera and also scaling the content displayed in the screen through rotations, zooming, and other actions;
- Selection, which gives the user the possibility of selecting a specific object, group of objects or area with the propose of highlighting it/them or execute some operation;
- Filtering, which gives the user the possibility of reducing the information displayed by removing data records or dimensions (or eventually, both);
- Reconfiguring, which gives the user the possibility of changing the graphical mappings of the data (i.e., the visual variables used for each attribute or dimension of the data), which in fact means that different visualizations can be applied to the same data;
- Encoding, which gives the user the possibility of changing the used visual variables to explore different features;
- Connecting, which gives the user the possibility of creating links between related content;
- Abstracting/Elaborating, which gives the user the possibility of changing the level of detail of the visualization.

It is common to combine and apply several of the mentioned techniques at once, creating a hybrid approach that fulfills the interaction requirements for the respective context.

An example of a filtering operation is presented in Figure 2.17, where some data records and dimensions are removed, which gives special focus to a part of the data more important to the user.

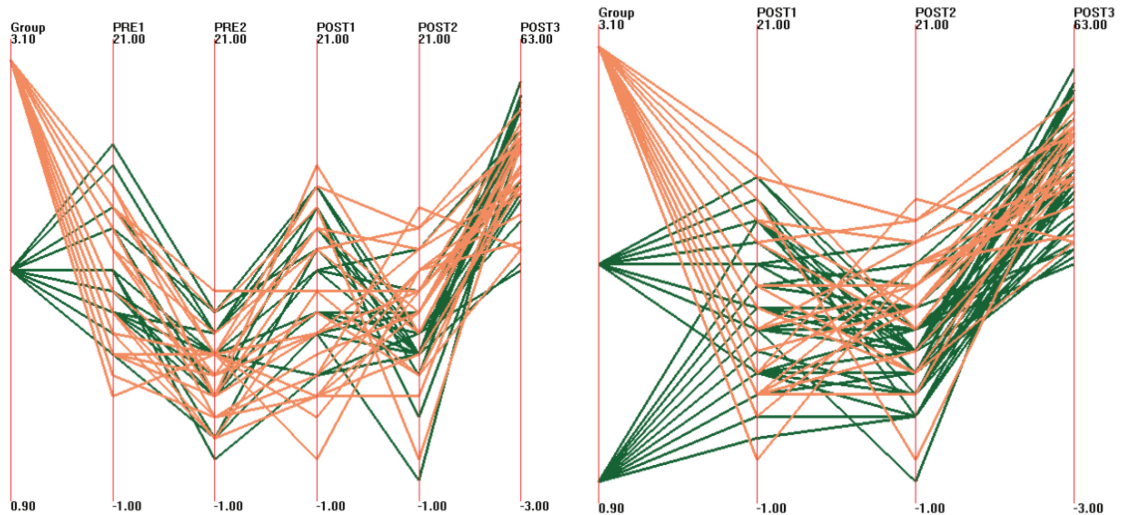


Figure 2.17: Example of a filtering operation by removing data records and dimensions (Ward et al., 2010). The original visualization is presented in the right image and the filtered visualization in the left image.

2.2.4 Visualization Validation

Most of the data mining approaches produce results that can be submitted to different types of metrics, and these metrics give numeric results that can be compared objectively through a mathematical analysis. The results produced by a visualization are graphical and not suitable for the same type of metrics. Therefore, this type of mathematical analysis can not be applied on this context. A survey produced by Seriai et al. (2014) analyzed this problematic and concluded that 78.16% of the articles included on their study used case studies to validate visualizations. These case studies are a qualitative analysis and follow into two approaches: the exemplification of scenarios where the visualization does what it is supposed to do, proving that it works; human-computer interaction studies with real users where their feedback is analyzed. The first approach is more focused on proving that the visualization is in fact a solution to the problem that it tried to solve. The second approach is more focused on evaluating the usability and the interaction of the visualization through quantitative measures (ask the user to do some task and measure the necessary time to do it) or qualitative ones (questionnaires or interviews). Therefore, depending on the type of tasks that the users have to do, it may in fact just evaluate the "user features" of the visualization and not its effectiveness. The survey also states that 16.09% of the analyzed articles use experiments with statistical tests, but these are in fact case studies used as hypothesis. The remaining 5,75% are interviews to the users, which are again a human-computer interaction strategy but less focused on specific tasks and more focused on generic aspects of the visualization and its contexts.

2.3 Conclusion

From the concepts described on this background knowledge, the following conclusions can be drawn:

- DBSCAN is the more common algorithm for density-based clustering;
- HDBSCAN is a DBSCAN extension that automatically chooses the *Eps* parameter;
- The Silhouette Coefficient metric is a very common evaluation approach for clustering algorithms and it is based on the clusters cohesion;
- Jacques Bertin's principles introduce the fundamental concepts of the visualization field, like the visual variables definition and its levels;
- Edward Tufte's principles propose concepts that are able to improve visualizations, like the data-ink ratio to reduce the chart junk problem and the small multiples strategy;
- The interactive visualization concepts are mandatory to develop modern computational visualizations;
- The case studies have proved to be a good approach to validate visualizations.

For these reasons the concepts presented above were used in the developed work.

Chapter 3

Related Work

This section presents some of the recent work published in the fields approached by this thesis, which are trajectory mining (section 3.1), traffic visualization (section 3.2) and anomaly detection in vessels traffic (section 3.3). For each of these subjects an extensive search of highly cited and/or recently published papers was made, being the summary of these papers the content of this section. In the end some future directions of work are identified (section 3.4).

3.1 Trajectory Mining

Mazimpaka and Timpf (2016) introduce in their survey the concept of trajectory as a set of points where each point is represented by a spatial location (typically a latitude and a longitude for GPS-based data, but other sources are used, like GSM-based data), the timestamp at which the point occurred and, eventually, other informative data related with the point and its context. The survey proposes a division of the trajectory mining methods in two categories, primary and secondary methods. Primary methods try to categorized the trajectories, while secondary methods try to analyze the trajectories based on the results of the primary methods. The authors enforce that the data must be pre-processed before the mining techniques are applied through tasks like data cleaning, trajectory compression, map matching and trajectory segmentation. Primary methods fall in two common types of machine learning algorithms, clustering and classification. Clustering algorithms, being part of the unsupervised learning algorithms, have the advantage of not requiring labeled data. The article gives a special focus to algorithms like the ST-DBSCAN (Birant and Kut, 2007) and T-OPTICS (Nanni and Pedreschi, 2006), which are an extension of the well known clustering algorithms Density Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996) and OPTICS (Kriegel et al., 2011), respectively. The TraClus clustering algorithm (Lee et al., 2007), which operates on sections of trajectories instead of the entire trajectories, is another important alternative mentioned in the article. In terms of classification the article mentions that the classic algorithms

are used for trajectory mining, and gives some examples of algorithms that were already tested, like decisions trees and support vector machines (SVMs). The TraClass framework (Lee et al., 2008b) is also presented in this context, but it actually uses clustering in a first step and a SVM in a second step. Secondary methods fall in three types, being these pattern mining, outlier detection and prediction. Pattern mining tries to discover movement patterns in trajectories and is subdivided in three categories. One is repetitive pattern mining, which tries to discover patterns that are repeated periodically in several trajectories, and the authors give special focus to the Periodica algorithm (Li et al., 2010) for this purpose. Another one is frequent pattern mining, which tries to discover routes that are frequently followed in the trajectories based on time and location, and the article presents T-Patterns as one of the best ways to represent these routes. The last one is group pattern mining, which tries to discover patterns that several objects follow together as a group, and typically is applied through density-based clustering, where the result patterns can be of three types: flock, which consists of a group of objects that are seen together in several consecutive time-stamps inside a circular area; convoy, which is similar to flock but the circular area is relaxed to a neighborhood area found by density-based clustering algorithm; and swarm, which is similar to convoy but relaxes the need for consecutive time-stamps to just a minimum number of time-stamps where the objects must be seen together, consecutive or not. Outlier detection tries to discover trajectories that do not comply with the expected routes, which requires previous knowledge of what is the expected behavior. The authors present several algorithms for this method, like the TRAOD algorithm (Lee et al., 2008a), the iBAT anomaly detection framework (Zhang et al., 2011), and in general algorithms that apply distance measurement between trajectories or two-label classifiers for trajectories classification as normal/abnormal. Prediction tries to discover the future location of objects based on already seen trajectories of them, which is out of the scope of this thesis. The survey also concludes that, with the increasing usage of location-aware devices, trajectory mining has become a very attractive research topic with a lot of problems to be explored. Moreover, it identifies one of the current trends in movement data analysis that needs to be explored with trajectories in a near future that is to relate the movement to its context, creating the concept of context-aware mining of trajectories.

Cazzanti and Pallotta (2015) introduce in their survey a distinction between trajectory mining approaches for Automatic Identification System (AIS) data, namely point-based approaches, where complete independence is assumed between the spatial points (in the AIS context, between messages), and trajectory-based approaches, where each object trajectory is first estimated (in the AIS context, the vessels trajectories are estimated from the spatio-temporal distribution of the messages stream) and the algorithms work with the estimated trajectories. The first approach has several benefits in terms of performance but has the drawback of not accounting for important correlations between the spatial locations (points) of the vessel through the time, which is the main benefit of the second approach. For the point-based approaches the survey appoints some other articles that applied kernel density estimation (KDE) and association rule mining, and for the

trajectory-based approaches it gives a special focus on the Traffic Route Extraction and Anomaly Detection (TREAD) algorithm (later presented in this section), being this one the reference for this approach. The article also discusses the applicability of the TREAD algorithm in discovering and detecting vessel stationary areas, and includes an experimental part with AIS data from the Persian Gulf, collected between February 3 and May 7 of 2013, which refers to 12051 vessels. The results show that the main stationary areas were detected but no quality assessment was provided.

Sang et al. (2012) introduce an approach based on interpolation for restoring missing points from AIS trajectories, and presents an experiment where three mathematical models are tested and compared. The existence of missing points is very common with AIS data because the period of transmission is different for each vessel since it depends on its geographical position and speed. The three models used in the experiment are all piecewise-defined, which means that they are composed by several small functions that together form the final function. The first one is a linear interpolation, which means that it can only map data in a straight line format. The second one is a cubic interpolation, and for that reason it is able of mapping data with a curve format. The third and final one is a cubic spline interpolation, which is very similar to the last one but uses a spline approach, meaning that it can use lower-degree polynomials in the different small functions, choosing the degrees that are more suitable for achieving a final function with the maximum smoothness. The three models were tested with data from an unknown location collected on January 10 of 2012. The results show that all methods are able of mapping correctly trajectories when the vessel does not change its direction, but when the vessel makes a turn and the trajectory takes the form of a curve the cubic spline interpolation is the method that maps the data more accurately. The article from Zhang et al. (2017) presents a similar study but uses AIS data from a specific vessel collected on October 10 of 2016 from Wuhan in China. The results were coherent with the ones mentioned before, but the authors claim that the cubic spline method only performs well when no more than five points are missing from the trajectory.

Pallotta et al. (2013b) introduce a new algorithm called Traffic Route Extraction and Anomaly Detection (TREAD), which is able of discovering, in a unsupervised and incremental way, waypoint objects that can be stationary, entry or exit points, and cluster these waypoint objects into routes through an incremental DBSCAN algorithm. With the routes extracted, simple anomaly detection based on deviation of the normality is applied, using historical data and a minimum threshold. The algorithm was applied to AIS data from the North Adriatic Sea collected between March 1 and May 15 of 2012 and to AIS data in the proximity of the Strait of Gibraltar collected over two months. The results show that the main routes that were expected based on previous knowledge of the nautical charts where in fact detected, but some other not known or expected routes where also detected. In terms of the anomaly detection approach, it was tested with a specific vessel that was located in the Port of Livorno, in the Ligurian Sea area, and the anomalous behavior was successfully detected. The authors applied the same algorithm in another article (Pallotta et al., 2013a) with AIS data from the Northern Tyrrhenian Sea collected

between January 1 and February 20 of 2013, and used the concept of entropy to evaluate the results considering that a route with a higher entropy was more likely to be correct than others with a lower value, and some examples of routes with an entropy over 0.95 were presented.

Hadzagic et al. (2013) introduce the usage of the R software for data mining proposes trying explicitly to answer several questions, being one the identification of all vessels that go from port A to port B in a time period T. To perform this task first the authors created a grid that covered the region of interest, because this is a very simple and fast way of computing the distances between port's and vessel's locations, considering that each vessel and port position is associated with a cell of the grid and if a vessel visited a cell associated with a port then it visited the port. Secondly, the association rule mining algorithm APRIORI was applied. This algorithm identifies frequent items on a dataset and extends them to groups, considering only as valid the groups that appear frequently in the dataset. These groups are then derived to association rules which are able to map the trends of the data. The approach was tested with AIS data from an area limited by latitude between 35°N and 53°N and longitude between 80°W and 48°W (an area between the Atlantic coast of Newfoundland, Labrador in Canada and Virginia in USA) collected between 3 and 10 of November of 2011, containing 93000 positions. The results present some rules with a very good lift value (between 1 and 10), but no rationale is presented for the quality of these results.

Liu and Chen (2013) propose a new method to recreate vessels' routes from AIS incomplete data. The concept of incomplete data in this paper is actually normal AIS data but, considering that a vessel can spend a lot of time without communicating the current position, huge gaps can exist between two consecutive collected positions. The authors introduce a new algorithm capable of creating new simulated points to fill these gaps. The new points are created based on a calculated distance that takes in consideration the direction of the vessel and this process is executed in a loop until the gap is filled in. The authors experimented the technique with AIS data from north of Australia collected between May 13 and October 7 of 2012, recreating the routes from 4 specific vessels. The results show that the created positions of the vessels adjusted well to the routes, with the exception of when the inference was made at turning corners.

Gonzalez et al. (2014) introduce two methods for the extraction of vessels lanes (called ship lanes), the first one based on point clustering and the second one on segment clustering. The point clustering approach detects entry, exit and turning points, groups them based on density and connects the clusters with common trajectories to create the final paths. The segment clustering approach partitions the vessels trajectories by the turning points, clusters these segments with a density based approach and calculates the final path with the adjustment of regressions. These methods were experimented with data from an area around Algarve (Portugal) collected from 1 week, with approximately 28000 points. The results show that both methods detected the main vessels lanes of the area, but the segment clustering approach gives paths with more breaks in between.

Sun et al. (2015) introduce an extension of the DBSCAN algorithm particularly for trajectory clustering and proposes a few steps of data preparation on AIS data, like cleaning the data with noise and re-sampling the data to periods of 15 minutes. The algorithm was experimented with AIS data from an unknown location, collected between November and December, with 506884 points. The sum of squared errors is proposed as an evaluation criteria of the work but the results only show that some trajectories are found and no rational is presented for the quality of those results.

Wu et al. (2015) introduce an hierarchical approach based on fusion to detect trajectories and stopping points. First the positions from the same vessel are aggregated into sub-trajectories by computing the angle of every two consecutive points and a new sub-trajectory is detected when the difference of the maximum angle value and the minimum angle value is greater than a threshold (defined as 30 in the article). Secondly, the sub-trajectories are aggregated into route segments and stops. To perform such task, the authors observed through the data that when a vessel is moored or anchored the length of the sub-trajectories is typically under 30 meters and the angles are above 60° , and based on this info they estimate the degree of membership of each sub-trajectory to a so called stopped state. If this degree is above a given threshold (0.2 in this context) it is considered a stop. The remaining sub-trajectories are aggregated in route segments. The authors experimented their approach with AIS data from the coast of China of one specific vessel, using 32176 records collected between 2 and 11 of September of 2013. The initial data was aggregated into 1297 sub-trajectories and these ones were aggregated into 250 route segments and 16 stops. The authors compared the results with a combination of the algorithms CB-SMoT and DB-SMoT and say that the proposed one is more precise, but no proves besides a visual comparison are presented to justify this affirmation.

Liu (2015) introduces a new trajectory clustering algorithm that is an extension of the DBSCAN algorithm called Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction (DBSCANSD). This extension uses the exact same approach of the DBSCAN but when it is looking for neighbors it also considers the Course Over Ground (COG) and the Speed Over Ground (SOG), being that two points are considered neighbors if the distance between them is less than a given radius and their Course Over Ground (COG) and Speed Over Ground (SOG) absolute differences are less than a given threshold for each of them. There for, the algorithm also requires these thresholds as an input, which in fact are the maximum direction variance (COG) and the maximum speed variance (SOG). The author also proposes a technique to reduce the number of elements in the cluster, called Gravity Vectors, where the cluster is partitioned into a grid (the grid split criteria is defined by the domain knowledge or multiple experiments) and for each cell the average of the COG, SOG, latitude and longitude values and the median distance are calculated, merging these values into a gravity vector. This means that each cluster will have as many gravity vectors as grid cells. The author also addresses the detection of stopping areas through the usage of the normal DBSCAN without any extension. The algorithms were tested with AIS data from the strait of Juan de Fuca and the Los Angeles Long Beach, collected between November 1 and December 31 of 2012, containing

67850 and 327694 points respectively. In both sources the main trajectories and stopping areas that were expected accordingly to the navigation rules defined for both areas were in fact found. The article from Yan et al. (2016) followed the exact same approach and experimented it with two weeks of AIS data from the Singapore strait that includes over 4000 vessels of different types and 340000 points. The results also look promising but no point of comparison is provided.

Zhang et al. (2016) introduce a technique to detect and simplify trajectories from AIS data through an adaptation of the Douglas-Peucker algorithm. This algorithm creates line segments that approximate the original points (in this case, the original positions of the vessels) by comparing the distance between the existing line segments and each original point with a given threshold, and when the distance between the line and one or more points is higher than the threshold a new segment is created that connects to that point. This process is executed in a loop until no distances higher than the given threshold exist. The technique was experimented with AIS data collected between 2 and 12 of July of 2011 from the Qiongzhou Strait AIS base station, considering the trajectories from 962 vessels created with 5902840 points. The results were compared through a visual approach and also by measuring traffic flow statistics, namely the number of vessels crossing specific areas and their average length and speed. The visual approach show that the generated trajectories match the original ones with just a few representative points. The traffic flow statistics measured with the original and simplified data show minor differences in the values, with a maximum difference of 8 for the number of vessels. Based on the same approach, the article from Li et al. (2016) introduces the usage of a density visualization for comparing the original trajectories with the ones compressed with the Douglas-Peucker algorithm with the propose of discovering an ideal threshold that provides a good representation of the original routes with a high reduction of the necessary points. The authors experimented the approach with data from the Wuhan section of the Yangtze River containing 29015 points and 187 vessels, and obtained the best results with a threshold of 2.5×10^{-6} , producing a compression rate of 44.84% without losing the characteristics of the original trajectory.

Dobrkovic et al. (2016) introduce a new approach to detect trajectory patterns using a genetic algorithm. Each gene of the chromosome is composed by a latitude, a longitude and a radius and the algorithm tries to maximize a fitness function that sums the number of vessel points inside each geographical circle generated by each gene. To improve the genetic algorithm the authors also propose the application of several features, with major emphasis on the usage of a quad tree structure to store all the points subdivided in smaller regions, which makes the queries to these points must faster because only the points adjacent to the circle of the gene under test are considered. The extracted routes are direct graphs that have as nodes the circles obtained from the best chromosome given by the genetic algorithm. This method was experimented with data from two Dutch provinces (South Holland and Overijssel) and the extracted routes were compared with a map of rivers and canals of them. The results show that the extracted patterns match the main water routes.

Lei et al. (2016) propose a new route extraction method called Maritime Traffic Route Discovery (MTRD) that is able to extract routes from AIS data. It starts by mining frequent regions through the creation of a grid that maps the trajectory data and classifying each cell (region) as frequent if the number of trajectories that cross that cell is higher than a given threshold. These regions are then converted to frequent patterns through the application of the Prefixspan algorithm (Han et al., 2001) and these patterns are summarized by eliminating the ones that are included in another pattern and by concatenating the ones that have similarities in the beginning or the end parts. Finally the routes are extracted from the patterns by calculating the average direction and position of each cell. The method was experimented with an AIS dataset of an unknown location that covered 20639 trajectories and 21202212 positions. The results were analyzed in terms of average coverage rate and, with variations between 200 and 400 for the number of necessary sequence patterns, this rate is up to 76%, which is a good indicator of the effectiveness of the method.

A summary of the related work analyzed in the trajectory mining context is presented on Table 3.1.

Table 3.1: Summary of trajectory mining related work.

Publication	Goals	AIS Data	Strategy	Evaluation Metrics	Results
Sang et al. (2012)	Restore vessel's trajectories	Unknown location data collected on January 10 of 2012	Piecewise Linear, Cubic and Cubic Spline Interpolation	Visual analysis	Piecewise Cubic Spline Interpolation is the method that better models trajectories with a curve format
Pallotta et al. (2013b)	Extract main vessels' routes	North Adriatic Sea data collected between March 1 and May 15 of 2012; Strait of Gibraltar data collected over two months	Traffic Route Extraction and Anomaly Detection (TREAD)	Visual comparison	Main routes are coherent with the nautical charts
Pallotta et al. (2013a)	Extract main vessels' routes	Northern Tyrrhenian Sea data collected between January 1 and February 20 of 2013	Traffic Route Extraction and Anomaly Detection (TREAD)	Route entropy	Some routes with an entropy over 0.95
Hadzagic et al. (2013)	Identify vessels that go from port A to port B in a time period T	Data of an area between the Atlantic coast of Newfoundland in Canada and Virginia in USA, collected between 3 and 10 of November of 2011, containing 93000 points	Association rule mining with APRIORI algorithm	Lift	Rules with high lift values between 1 and 10

Continues in next page.

Continued from previous page.

Publication	Goals	AIS Data	Strategy	Evaluation Metrics	Results
Liu and Chen (2013)	Complete trajectories with gaps between the points	North of Australia data collected between May 13 and October 7 of 2012	Trajectory interpolation based on position, COG and SOG	Visual comparison	No quality assessment; The generated points adjusted well to the trajectories
Gonzalez et al. (2014)	Extract vessels' routes and lanes	Algarve (Portugal) data, collected over one week and containing 28000 points	Density-based clustering over points and segments	Visual analysis	No quality assessment; Segment clustering gives routes with more breaks
Sun et al. (2015)	Extract maritime routes from spatio-temporal data	Unknown location data, collected between November and December and containing 506884 points	Density Based Spatial Clustering of Applications with Noise (DBSCAN)	Sum of squared errors	No quality assessment; Some trajectories were found
Wu et al. (2015)	Extract vessels' trajectories and stopping points	Coast of China data for one specific vessel, collected between 2 and 11 of September of 2013 and containing 32176 points	Hierarchical fusion of points and sub-trajectories	Visual comparison	Precision outperforms similar algorithms

Continues in next page.

Continued from previous page.

Publication	Goals	AIS Data	Strategy	Evaluation Metrics	Results
Liu (2015)	Extract vessels' trajectories and stopping points	Strait of Juan de Fuca and the Los Angeles Long Beach data, collected between November 1 and December 31 of 2012 and containing 67850 and 327694 points, respectively	Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction (DBSCANSD)	Visual comparison	Trajectories are coherent with the navigation rules
Yan et al. (2016)	Extract vessels' trajectories and stopping points	Singapore strait data, collected over two weeks and containing 340000 points for more than 4000 vessels	Density Based Spatial Clustering of Applications with Noise (DBSCAN)	Visual analysis	No quality assessment; Some trajectories are presented visually
Zhang et al. (2016)	Detect and simplify trajectories	Qiongzhou Strait data, collected between 2 and 12 of July of 2011 and containing 5902840 points for 962 vessels	Douglas-Peucker algorithm	Visual comparison; Comparison of the number of vessels crossing specific areas and their average length and speed	Visually the simplified and the original trajectories match; The statistical values are almost equal, with a maximum difference of 8 for the number of vessels

Continues in next page.

Continued from previous page.

Publication	Goals	AIS Data	Strategy	Evaluation Metrics	Results
Li et al. (2016)	Simplify trajectories and find an ideal threshold for the compression	Wuhan section of the Yangtze River data containing 29015 points for 187 vessels	Douglas-Peucker algorithm; Density visualization	Visual density comparison	With a threshold of $2.5 * 10^{-6}$ a compression rate of 44.84% is obtained without losing the characteristics of the original trajectory
Dobrkovic et al. (2016)	Extract vessels' routes	Dutch provinces South Holland and Overijssel data	Genetic algorithm	Visual comparison	The trajectories match the map of rivers and canals of the provinces
Lei et al. (2016)	Extract vessels' routes	Unknown location data containing 21202212 points	Maritime Traffic Route Discovery (MTRD)	Average coverage rate	With variations between 200 and 400 for the number of necessary sequence patterns, the rate is up to 76%
Zhang et al. (2017)	Restore vessel's trajectories	Wuhan (China) data collected on October 10 of 2016	Piecewise Cubic Hermite and Spline Interpolation	Accuracy; Visual analysis	Piecewise Cubic Spline Interpolation is the method that better restores trajectories; The method only performs well when the number of missing points is below 5

In conclusion, the following key aspects can be described:

- A lot of work exists in the trajectory mining field, but when considering only AIS data the quantity of available work is considerable reduced, possibly because AIS data is difficult to obtain;
- With the exception of one more common trend, the authors follow very different approaches in their work, from simple frequency analyses to rule mining and genetic computing approaches;
- The only recognized trend in the field is the usage of density-based clustering, particularly the DBSCAN algorithm and extensions of it;
- Concerning the techniques' evaluation, there is no global metric followed by the authors but a lot works rely on visual comparison.

Focusing on the reasons why density-based clustering is a trend in the field, if we look at the sea as a big space with no physical barriers we can conclude that a vessel can follow any trajectory. This does not happen because some traffic corridors are defined for the vessels to sail. Therefore the vessels must sail on those corridors, which in fact means that the majority of the vessels follow the same trajectories (at least if they are in compliance with the law). With this rational we can conclude that some areas of the sea have the majority of the traffic, and therefore these areas have a much higher density of vessels sailing on them. Density-based clustering serves this exact propose, detecting and describing areas (clusters) where the traffic (density) of vessels (points) is much higher. The reasons why DBSCAN is the more common are not only because it is the more known and used density cluster algorithm but also because it supports any distance function, which in the case of the AIS data gives the possibility of including in this distance calculation other (possibly) important attributes, like the COG and the SOG.

3.2 Traffic Visualization

Riveiro and Falkman (2009) introduce a visualization model that is able to present the normal behavior of vessels through the density probability of each combination of the more relevant kinematic values (SOG, COG, heading, latitude and longitude), using a grid of 6x6 where each column and row represent one of the kinematic values. The model allows the visualization of the density probability of each pair of kinematic values in a 3D scatter plot by selecting two columns of the same row. After the selection, three scatter plots are presented with the expected information, each one with a different scale and perspective. A 2D plot is also presented showing only the kinematic values and the dangerous zones, that are specific areas of the plot where the values are considered anomalous by one or more rules defined by expert's knowledge. Notice that the density probabilities mentioned above are obtained through a probability density function created from the application of

the methods Self Organizing Maps (SOMs) and Gaussian Mixture Models (GMMs). The visualization model was experimented with AIS data from the Swedish coast, collected from 17 days of January and containing 3.5 million points and 600 vessels per day. The first 5 days of data were used to train the methods that create the probability density function. The results look promising but no quality justification is provided for them. The visualization model of this experiment is presented in Figure 3.1.

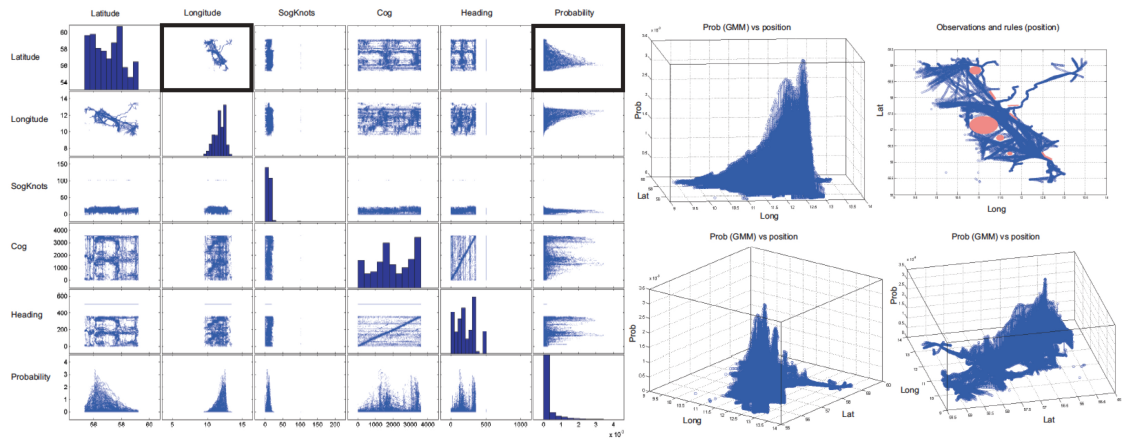


Figure 3.1: Visualization model of the experiment from the article of Riveiro and Falkman (2009).

Willems et al. (2009) introduce a new kernel density method based on the speed of the vessels that is able to measure the contribution of each vessel in each point of the map over time, being this contribution modeled with a convolution. In terms of visualization these densities are displayed through continuous or discrete color mappings, and to get both an overview and a detail view of the data two densities are calculated for each vessel, one with a larger kernel for the overview and the other one with a smaller kernel for the detail view, being both displayed in the same area simultaneously but applying a different shading to the density of the larger kernel. The method was experimented with AIS data from the port of Rotterdam in the Netherlands and the entire Dutch coast. The results show that, comparing with other density approaches that do not take the speed into account, this method highlights some of the speed patterns, specially vessels that are moving slower than expected. This comparison is presented in Figure 3.2. Notice that the Douglas-Peucker algorithm was applied to the AIS data in order to reduce the number of points necessary for the representation of each trajectory.

Jiacai et al. (2012) introduce a new data visualization model that divides the region of interest into a grid and calculates an index of maritime traffic situation for each cell of the grid. This index combines three features that are the rate of encounter, the rate of turn and the vessel acceleration. Each of these features is multiplied by a weight (in the study all features have the same weight, therefore $\frac{1}{3}$), and the final value of the index is the sum of the values. A region (cell of the grid) with a higher index has a more complicated and danger traffic situation than a region with a lower index, but no threshold values are

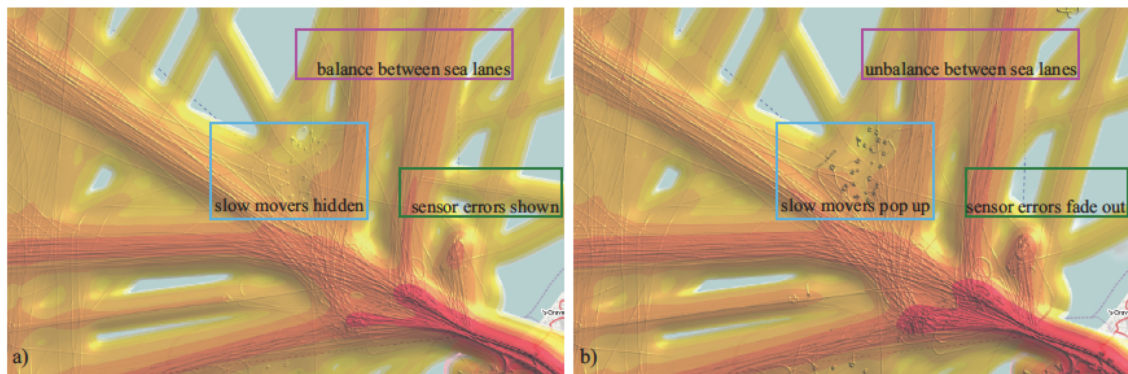


Figure 3.2: Density visualization comparison with and without speed (right and left images, respectively) from the article of Willems et al. (2009).

proposed in the article. The method was experimented with two AIS datasets from the Chinese ports of Xiamen Bay and MeiZhou Wan, containing the first one 865295 points and 649942 vessels and the second one 1669 points and 1546 vessels. The index values were displayed in the Electronic Chart Display and Information System. The results show that 4 areas in Xiamen Bay and 3 areas in MeiZhou Wan were identified as potentially more danger. The visualization from this last area is present in Figure 3.3.

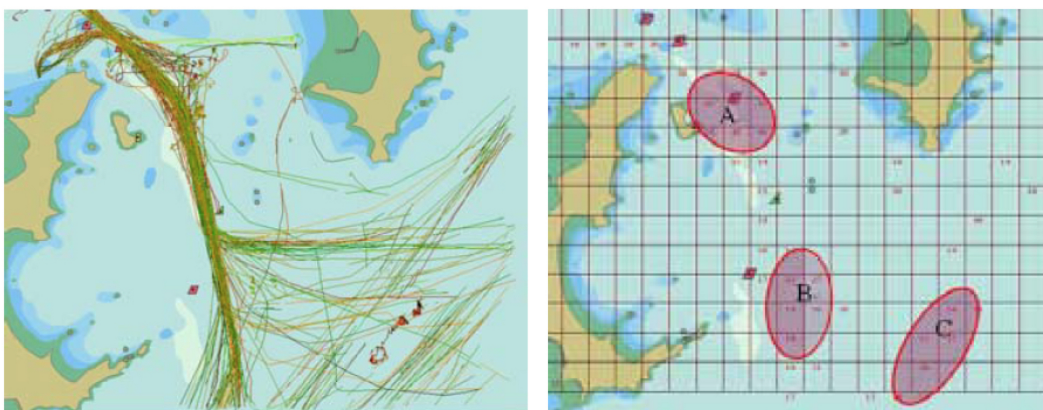


Figure 3.3: Index visualization of the MeiZhou Wan area from the article of Jiakai et al. (2012). The 3 potentially danger areas are identified on the map.

Gao and Shiotani (2013) propose 2D and 3D views more suitable for the presentation of AIS data. The views are introduced through several case studies, all of them using AIS data collected from the Osaka Bay, located east of the Seto Inland Sea in Japan, on March 7 of 2012 and containing 333 vessels. In the 2D approach, the authors divided the area of interest in a grid with cells of 5x5 kms and calculated the number of vessels that crossed each cell. These values are then used to create a visual density distribution of the vessels in the area through a color gradient. This 2D approach is presented in Figure 3.4. Considering that the AIS data may be transmitted over different periods for different vessels there is the risk of calculating wrong density values because of the missing positions

from the vessels. To solve this problem the position data of each vessel is interpolated in a per-second basis. In the 3D approach, the authors created a simulation where a specific container vessel was followed in a 3D map containing detailed information about the surrounding environment like water depth and other vessels. This view creates a very trustful representation of the environment where the vessel is sailing and can even integrate other AIS data (e.g., the direction of travel). This 3D approach is presented in Figure 3.5.

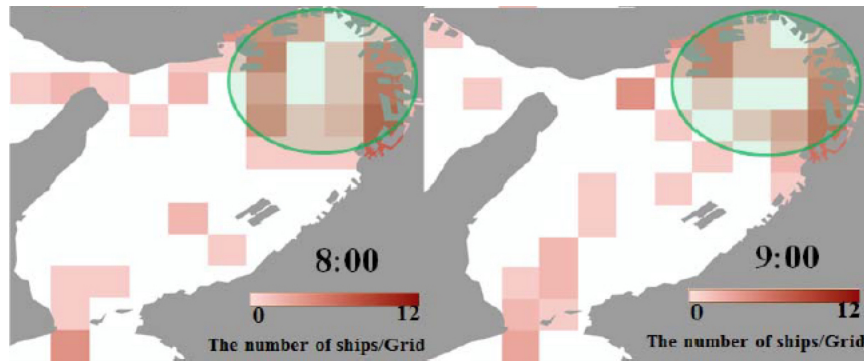


Figure 3.4: 2D density visualization from the article of Gao and Shiotani (2013) for two hours of the day. High and low densities are represented by red and white, respectively.

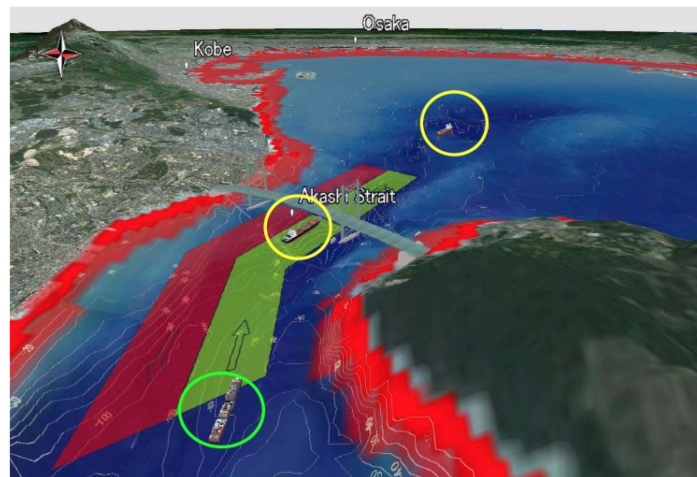


Figure 3.5: 3D visualization from the article of Gao and Shiotani (2013). The vessel being followed is in the green circle.

Fiorini et al. (2016) introduce a pipeline of actions to go from raw AIS data to a proper visualization of the vessels routes. This pipeline follows a sequential process starting with raw data being aggregated and reduced in terms of number of attributes (only location, time-stamp and kinematic information is kept) and in terms of proximity (if several points are within the same location only a representative one is kept). The resulting points are then aggregated into segments using a format that is ready for geographical visualization and that is compliant with the specifications of the Open Geospatial Consortium. Finally, the generated trajectories are then presented in a interactive web-like page with several types of filtering, like selecting specific routes or attributes. The described pipeline is

presented in Figure 3.6 with the sequence of actions. The approach was experimented with a large AIS dataset with over 90 millions rows from the entire world collected in October of 2015. To perform all the steps of the pipeline only open-source tools were used, namely PostGIS ¹ for the raw AIS data aggregation and reduction, GeoServer ² for the points' aggregation into segments in the desired format and OpenLayers ³ to present the trajectories in a web-like page. The resulting application allows for a fast and complete exploitation of the trajectories with the possibility of querying specific routes or attributes.

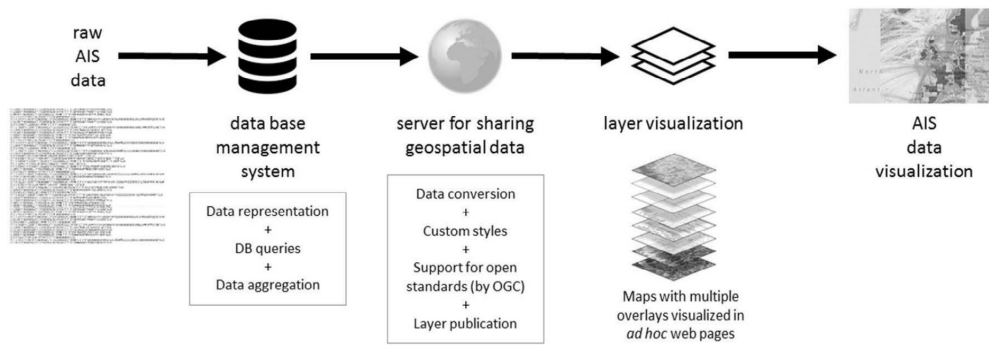


Figure 3.6: Pipeline of actions from the article of Fiorini et al. (2016).

Chen et al. (2016) introduce the concepts of direct and summary visualization in the context of AIS data. In a direct visualization approach the data is displayed as is without any kind of previous processing, which can perform well with small quantities of data but when this quantity increases the analysis can become very difficult. In a summary visualization approach the displayed data is an abstract representation obtained from the original data, which is more suitable for big data contexts and for patterns representation. In this second approach the authors give special emphases in density-based representations like the heat map, and propose an extension of the heat map concept for the AIS context that works with relative values instead of absolute values like the basic heat maps, which is required to ensure a bigger continuity in the density representation. This extension also divides the region to be presented in cells and calculates the heat contribution value of each pixel point by summing the distances of that point to the vessels and, after that, calculates the relative heat by normalizing the value with the minimum and maximum heat contribution values of the region. This approach can also be applied to kinematic information like the velocity. Both the direct and the summary approaches were experimented with data from the Wuhan Yangtze River, collected between 11 and 17 of August of 2015 and containing 167865 points. The results from the direct visualization show that the general trends of the patterns are displayed but the view is very confuse and a lot of data is overlapped, and the results from the summary visualization show that the relevant patterns are properly displayed through dense, high-speed and extreme slow-down areas. These results are presented in Figure 3.7.

¹Available at <https://postgis.net/>

²Available at <http://geoserver.org/>

³Available at <https://openlayers.org/>

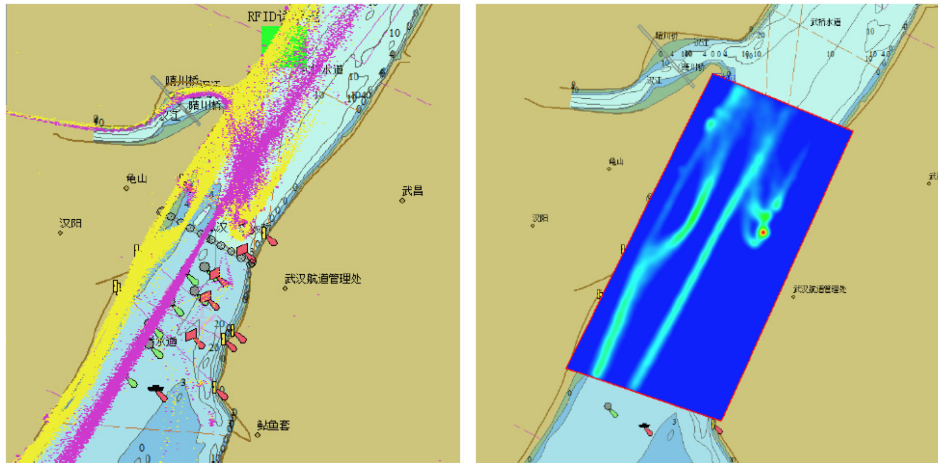


Figure 3.7: Direct visualization (left) and summary visualization (right) of the experiment from the article of Chen et al. (2016).

A summary of the related work analyzed in the traffic visualization context is presented on Table 3.2.

Table 3.2: Summary of traffic visualization related work.

Publication	Goals	AIS Data	Strategy	Evaluation Metrics	Results
Riveiro and Falkman (2009)	Interactively visualize vessels' normal behavior	Swedish coast data, collected from 17 days of January and containing 3.5 million points and 600 vessels per day	Interactive visualization with connecting technique	Visual analysis	No quality assessment
Willems et al. (2009)	Visualize vessels' movements	Port of Rotterdam in the Netherlands and Dutch coast data	Density-based visualization	Visual analysis	No quality assessment; Taking speed into account highlights speed patterns
Jiacai et al. (2012)	Visualize maritime traffic situation	Chinese ports of Xiamen Bay and MeiZhou Wan data, containing the first one 865295 points and 649942 vessels and the second one 1669 points and 1546 vessels	Index calculation for each cell of a grid mapping of the area	Visual analysis	4 areas in Xiamen Bay and 3 areas in MeiZhou Wan were identified as potentially danger
Gao and Shiotani (2013)	Visualize AIS data in 2D and 3D	Osaka Bay data collected on March 7 of 2012 and containing 333 vessels	2D density-based visualizations; 3D modeling	Visual analysis	2D visualization provides traffic density evaluation; 3D visualization provides a trustful representation of the environment

Continues in next page.

Continued from previous page.

Publication	Goals	AIS Data	Strategy	Evaluation Metrics	Results
Fiorini et al. (2016)	Create an interactive visualization for maritime traffic	Data from all over the world, collected in October of 2015 and containing 90 million points	Data aggregation and reduction; Interactive web-like geographical visualization	Visual analysis	An application that allows the visualization and exploitation of maritime trajectories with interactive techniques
Chen et al. (2016)	Visualize maritime trajectories	Wuhan Yangtze River data, collected between 11 and 17 of August of 2015 and containing 167865 points	Direct visualization; Summary visualization through density (heat map)	Visual analysis	Direct visualization has overlapping issues; Summary visualization shows the relevant patterns displayed through dense, high-speed and extreme slow-down areas

In conclusion, the following key aspects can be described:

- A lot of work exists for the traffic visualization field but for the AIS data context the quantity of work available is once again very reduced, possibly because of the same reason as in trajectory mining;
- With the exception of one paper that explores the three-dimensional plane, the remaining ones explore the two-dimensional plane through the usage of maps divided into grids and explore the usage of visual variables to display information on each cell;
- The general trend of the works is to display the traffic information through density-based visualizations, using visual variables like the value to show different densities;
- Concerning the techniques' evaluation, all the works are evaluated through visual analysis.

Focusing on the reasons why density-based visualizations is a trend in the field, an area with more traffic is in fact where more vessels sail frequently. Therefore, this concept of frequency can easily be transformed into a density. From a visualization perspective, density is actually a very good way of displaying different frequencies because it introduces the concept of an ordered gradient, which is highly informative when displayed through the right visual variables (in this case, the ones that are able to display order).

3.3 Anomaly Detection

Tu et al. (2017) present in their survey anomaly detection algorithms based on trajectory mining, and distinguishes the vessels anomalies in three types: position anomalies, where the vessel is located in a forbidden or unexpected position; speed anomalies, where the vessel speed is above or below the normal speed; and time anomalies, where the vessel is sailing during a period that was not expected. Focusing on the first two types of anomalies, the survey also categorizes the anomaly detection algorithms in two types: geographical model based methods, where the models are built for a specific area and trained with local data from that area; and parametrical model based methods, where the built models are independent of the areas. For the first type the survey presents some example models like the Normalcy Box (Rhodes et al., 2005), the Fuzzy ARTMAP (FAM) (Bomberger et al., 2006), the Holst model (Laxhammar, 2008) and the Potential Field Method (PFM) (Osekowska et al., 2015), and for the second type it also presents some examples like the Trajectory Cluster Modeling, the Gaussian Process and Bayesian Networks. The survey concludes that the geographical methods are more intuitive but usually do not make use of all the available information, while the parametrical methods are less intuitive but can easily include all information of expert knowledge.

Handayani et al. (2013) introduce the usage of support vector machines (SVMs) for anomaly detection in trajectories. SVMs are supervised classification models that can be trained with labeled nonlinear data and are able to map this data into an higher dimensional feature space using a kernel function. When presented with new data, based on the previous training, SVMs are able to classify this new data with known labels. The raw AIS data is not labeled and in order to create the training set labeling of part the data is required. Therefore the authors propose the cleaning and splitting of the data by trajectories, and these resulting routes are displayed to a domain expert that classifies them as normal or abnormal. The training set is then created with part of the trajectories of each type. The article also proposes the interpolation of the positions in a 3 minutes interval to avoid gaps. The method was experimented with AIS data from Port Klang, collected between July and September of 2013 and containing 9845 points. Three different divisions of the data into training and testing sets were experimented, with the respective percentages of 60-40, 70-30 and 80-20. Also, each of these experiments was executed with and without interpolation of the data. The results show that the best combination is the division of 70%-30% for the training and testing sets using interpolated data, which resulted in 99.81% of accuracy.

Mazzarella et al. (2014) introduce a process to discover fishing areas that can easily be applied to anomaly detection, namely the detection of fishing activities in illegal areas. The process starts by cleaning the data, removing reported positions for the same Maritime Mobile Satellite Identity (MMSI) that are very close to each other, and by aggregating the positions of the same vessels in trajectories. Then an algorithm to cluster positions of a vessel where fishing behavior is probable is applied to each trajectory. This algorithm takes in consideration if the COG variation between two points is greater then a given threshold and if the SOG is between a given interval. Also, a cluster is only considered valid if it has a minimum of points and if its duration time interval is between given thresholds. Finally the DBSCAN algorithm is applied to obtain the fishing areas from the fishing points (this step is not relevant for anomaly detection proposes). The entire process was experimented with AIS data from the Icelandic waters, collected between January 1 and January 31 of 2014 and containing 1055 vessels. The results were compared with trajectories followed by a specific fishing vessel and the fishing points and areas detected by the process matched with the ones where the vessel actually fished.

Mascaro et al. (2014) introduce an anomaly detection approach using bayesian networks (BNs) that are able to measure how probable is the occurrence of a given trajectory. Based on these probabilities, and considering that they are usually very low, an anomaly score is created from each probability applying the expression $-\log_2(x)$, being x the probability. In order to create the bayesian networks the software CaMML was used. CaMML uses a stochastic search and a score approach to create causal BNs from training data. The authors propose two learning approaches for the models generation. One is to take advantage of the fact that the trajectories have a time series format, which allows the creation of dynamic bayesian networks where the attributes change over time for each trajectory. The other is to obtain a summary of each trajectory and create

static bayesian networks with it. Both approaches were experimented with AIS data from the NSW coast of the Sydney port collected between May 1 and July 31 of 2009. The raw AIS data (near 9.2 million of positions for 544 vessels) was divided into trajectories based on the MMSI and transmission gaps bigger than 6 hours, and to ensure an equal distribution of the trajectories the data was interpolated with intervals of 10 seconds. Then, to effectively test the anomaly detection, several anomalies were injected in the data by three approaches, namely changing the vessel type, splicing trajectories together and adding random anomalous trajectories. The change of vessel type produced an increase on the anomaly scores, with a positive detection around 87.2% in the time series model and 69.4% in the summary model. Considering the spliced trajectories, 70 random ones originated by vessels of the same type and another 70 from vessels with different types were altered, and the summary model responded well increasing the anomaly score from 89 to 115.4 and 121.3 respectively for each of the 70 trajectories mentioned above, while the time series model responded badly with a decrease in the scores to 45.6 and 48.9 respectively. Injecting random anomalous trajectories resulted in different results from both models because they detected well trajectories with close interactions, with an increase of 49.1 and 30.1 in the scores from a base of 90.8 and 45.7 for the summary and the time series model respectively, but the time series model detected very short trajectories better with an increase of 17 in the score compared with only 4.7 for the summary model, and this last one detected the unusual stops better increasing the score by 28.3 compared with an increase of only 2.9 for the time series model. In average the score always increased after the injection, which proves that the anomaly detection works as expected, and in general the summary model outperformed the time series model. The authors also suggest a combined usage of both models to complement their strengths.

Osekowska et al. (2015) introduce a novel method called Seafaring Transportation Anomaly Detection (STRAND) based on the potential fields theory where the idea is that when a vessel passes in a position it assigns a charge to that position. Then, for each position the total charge is calculated by summing the local charges assigned by the vessels. This means that a location has a higher potential when more vessels visit it. The anomaly detection is then made through a binary classification, such that a vessel position is considered as (possible) anomalous if the potential of this position is below a given threshold. The confirmation of the vessel position as anomalous is made explicitly by a human operator through an information visualization platform platform that presents the locations potential in the form of a heat-map. The method was experimented with AIS data from the Piast Canal and the Gdansk bay area, collected over 20 days and containing 2263 vessels. The results show that the traffic patterns created for each area match the expected routes and the anomaly detection accuracy was calculated using the real values of the datasets and using altered values of speed and course. The results show that the method detected more anomalies with the altered values, which is a good indicator for the quality of the algorithm.

Liu (2015) introduces a new anomaly detection method based on distances. It starts by devising the AIS dataset in two, one with the stopping points an the other one with the

moving points, using a threshold of 0.5 knots for the SOG value as the division criteria. Then, based on the results obtained from the DBSCANSD algorithm that are the stopping points and the gravity vectors, the proposed algorithm calculates the absolute division distance for each of the new stopping points and the relative and cosine division distances for each of the new moving points. When the minimum of the absolute and relative distances are above a given threshold or the maximum of the cosine distance is below a given threshold (depending on the type of the point) the point is labeled as abnormal. These three thresholds can be estimated through several experiments until the more suitable ones are found. The method was experimented with an AIS dataset from the strait of Juan de Fuca, collected between November 1 and December 31 of 2012 and containing 67850 points, that was previously labeled by a domain expert. The used thresholds were 97.290 for the absolute distance, 5.938 for the relative distance and 0.485 for the cosine distance. The results show an overall accuracy of 90.49% on the anomalies detection.

Soleimani et al. (2015) introduce a new anomaly detection framework that is able to calculate an abnormality score for each trajectory. The method starts by devising the region of interest into a fixed resolution grid and setting each cell of the grid with the value 1 if at least one vessel crosses the cell or 0 if no vessels cross it. Then, for each trajectory, the well known A* algorithm is applied in order to extract the shortest possible path for each one, using the created grid to build a graph corresponding to the optimal trajectory. These optimal trajectories are considered the normal behavior. Therefore, for each of these trajectories and also for the real ones four features are extracted, namely the trajectory length, the area under the curve of the trajectory and the gradients of the trajectory with respect to latitude and longitude. These features are then normalized with the length of the optimal trajectory, being this step required to make the features' values scale-independent. The abnormality score is then calculated by adding all the differences between the optimal path's features and the real trajectories' features and normalizing all the units of the features to meters. This score is actually a measure of the deviation of the real trajectories from the optimal trajectories, based on the four discussed features. A score of 0 means that the trajectories are coincident, a score greater than 0 means that the real trajectory actually outperforms the optimal one, and a score less than 0 means that the trajectory may be anomalous, being the level of abnormality the absolute value of the score. The framework was experimented with AIS data from the North Pacific region collected between June and August of 2013 from the exactEarth database and for evaluation proposes 100 trajectories were randomly selected and labeled by an expert. The results show that for the 100 labeled trajectories the framework accuracy was 94%.

A summary of the related work analyzed in the anomaly detection context is presented on Table 3.3.

Table 3.3: Summary of anomaly detection related work.

Publication	Goals	AIS Data	Strategy	Evaluation Metrics	Results
Handayani et al. (2013)	Detect anomalies in trajectories	Port Klang data, collected between July and September of 2013 and containing 9845 points	Support Vector Machines (SVMs)	Accuracy	An accuracy of 99.81% with a division of 70%-30% for the training and testing sets
Mazzarella et al. (2014)	Discover areas with fishing activities	Icelandic waters data, collected between January 1 and January 31 of 2014 and containing 1055 vessels	Rules with speed and course thresholds; DB-SCAN	Visual comparison	The detected fishing activities for a specific vessel match the real ones
Mascaro et al. (2014)	Detect vessel's abnormal trajectories	NSW coast of the Sydney port data, collected between May 1 and July 31 of 2009 and containing 9.2 million points and 544 vessels	Bayesian Networks (BNs)	Difference between the abnormality scores when injecting anomalies	Summary model (static network) presented better scores in all injected anomalies except in the altered the vessel types, when compared with the time series model (dynamic network)
Osekowska et al. (2015)	Detect vessels with abnormal positions	Piast Canal and Gdansk bay area data containing 2263 vessels	Seafaring Transportation Anomaly Detection (STRAND)	Number of anomalies detected	Injecting anomalies like invalid values of speed and course increases the number of detected anomalies

Continues in next page.

Continued from previous page.

Publication	Goals	AIS Data	Strategy	Evaluation Metrics	Results
Liu (2015)	Detect vessels with abnormal positions	Strait of Juan de Fuca data, collected between November 1 and December 31 of 2012 and containing 67850 points	Distance calculation	Accuracy compared with a dataset labeled by a domain expert	Overall accuracy of 90.49%
Soleimani et al. (2015)	Detect abnormal trajectories	North Pacific region data collected between June and August of 2013	A* algorithm	Abnormality scores; Accuracy	For 100 labeled trajectories the abnormality scores were obtained with an accuracy of 94%

In conclusion, the following key aspects can be described:

- There is no trend in the field besides the fact that some works use classification methods, which means that different works follow different approaches;
- In terms of results, some works opt for displaying the result as a probability of abnormality and others as a binary classification (normal or abnormal);
- The majority of the works do not use a single algorithm or method to define a behavior as anomalous, but instead use a pipeline of steps (some of them simple preprocessing tasks, others application of specific algorithms) to achieve the results;
- The works that presented objective accuracy results actually show very promising values, typically above 90%.

3.4 Future Directions

Based on the related work presented above, the following directions could be explored in this thesis:

- In the trajectory mining context, considering that the trend is the usage of density-based clustering, this is a direction to take in consideration with the following approaches:
 - Direct application of published algorithms like TREAD or DBSCANSD;
 - Extend one of the studied algorithms (the more obvious ones are the DBSCAN or DBSCANSD) and consider other attributes in the distance function used by the algorithm.
- In the traffic visualization context, considering that the trend is the usage of density-based visualizations, this is a direction to explore, particularly by studying the possibility of including more AIS attributes in the density calculation;
- Also in the traffic visualization context, exploitation of 3D visualizations is a direction that makes sense considering that with more dimensions on the plane more data attributes can be represented and, also, the visualizations can become more realistic;
- In the anomaly detection context, the usage of abnormality scores or probabilities instead of binary classifications makes more sense when considering an integration of this information in a data visualization context, because it gives the possibility of exploring more information when using the right visual variables, and for that reason this is a direction to explore. Also, the possibility of this score/probability include information from more AIS attributes is something to take in consideration;

- Also in the anomaly detection context, considering that the quantity of labeled AIS data is very limited because this information requires a domain expert to label the data manually, some semi-supervised learning approaches could be considered, particularly with classification because some of the works prove that classification methods produce good results.

This page is intentionally left blank.

Chapter 4

Visualization and Implementation Choices

Considering the different abnormal behaviors previously mentioned on chapter 1, distinct approaches were developed for detecting each one. However, there are several implementation and visualization aspects of the developed platform that are common for all the approaches. These aspects are detailed in this section.

4.1 Dataset Characterization

On the basis of the developed platform was a proprietary dataset, supplied by the Critical Software company ¹, that contains 2852679 real Automatic Identification System (AIS) messages from the Portuguese maritime zone, corresponding to 20 days of data (between February 22 and March 12 of 2012). The AIS positions and static information from the vessels were stored in messages through a compact format used for transmission. This format needed to be processed and converted into a human readable format. Therefore, a tool for this propose was developed in Java and it uses a public library called AisLib ² to process the messages and create POJOs (Plain Old Java Objects) with their information. This library already implements the necessary decoding operations of the messages, and that is the main reason for its usage. The data was then extracted from the POJOs and written to CSV files.

To enlarge the dataset some other sources of AIS data were considered, namely:

- The MarineTraffic web site ³, which offers a Graphical User Interface (GUI) with AIS data on real time and has a proprietary RESTful Application Programming Interface (API) where the data from the web GUI is obtained. However, this API uses also proprietary and non-standard parameters to obtain the data for a specific

¹Available at <https://www.criticalsoftware.com>

²Available at <https://github.com/dma-ais/AisLib>

³Available at <https://www.marinetraffic.com/>

area (possibly to avoid that external entities can obtain that information, considering that they offer premium paid services), and for this reason this source was ignored;

- The VT Explorer RESTful API ⁴, which offers the AIS data through a web service with the possibility of filtering the data request by area. However, this is a paid service and for that reason this source was ignored;
- The AISHub RESTful API ⁵, which offers the same as the VT Explorer RESTful API but is free. However, to register in this service an AIS data feed needs to be supplied, because the service lives from this exchange of data between users. Considering that there was no feed to supply, a request was made to the service administrators asking for access without supplying any feed because the collected data was going to be used in an academic context. The administrators of the service created a temporary account that was active until January 31 of 2018.

Therefore, a new data collection tool was developed in Java to collect data from the AISHub RESTful API each minute (the web service blocks accesses more frequent than once per minute) and store the returned data. No additional processing was required as the data was already retrieved in the required format. Notice that only data from the Portuguese maritime zone was being collected. However, when the collection process ended, the data was analyzed and the conclusion was that it contained only positions from specific and small areas near some ports. Considering the high level of incompleteness, the data was discarded and only the original dataset supplied by the Critical Software company was used.

The AIS system has several types of messages for different purposes, but the dataset contained only two types: the ones with the reported positions and the ones with static information from the vessels. The remaining messages that were not available are not important for this usage context of the data, because the majority are related with aspects of the AIS communication protocol and not with the vessels. Regarding the messages with the positions, a CSV file was created that stores the following fields:

- Maritime Mobile Satellite Identity (MMSI) - the vessel unique identifier within the AIS communication system;
- Timestamp - the date and time of the position on the UNIX epoch time format;
- Longitude - the longitude of the position in decimal degrees;
- Latitude - the latitude of the position in decimal degrees;
- Speed Over Ground (SOG) - the speed of the vessel, in knots, considering the wind and current forces;
- Course Over Ground (COG) - the direction of the vessel, in degrees.

⁴Available at <http://www.vtexplorer.com/>

⁵Available at <http://www.aishub.net/>

Regarding the static messages, a different CSV file was created that stores the following fields:

- MMSI - has the same meaning as the one described above, and it is used to correlated the positions to the static information;
- Name - the name of vessel defined by the maritime operators (sometimes this field is empty or contains invalid information);
- Type - an integer representing the type of the vessel. There are 9 base types, represented by the first digit of the integer, and they are:
 1. Reserved
 2. Wing In Ground
 3. Special Category
 4. High-Speed Craft
 5. Special Category
 6. Passenger
 7. Cargo
 8. Tanker
 9. Other

Notice that the used library has a set of useful features, particularly one called message filters, that are a way of introducing conditions on the data processing mechanism, and when a AIS message does not meet those conditions it is ignored. Therefore, a location filter was used with the propose of ignoring any messages outside of the Portuguese maritime zone.

As stated before, the extracted data was collected between February 22 and March 12 of 2012. It contains positions for the 9 types of vessels available but with very unbalanced quantities for each type, being the statistics presented on Table 4.1.

Table 4.1: Number and percentage of positions by vessel type.

Vessel Type	# Positions	% Positions
Reserved	11327	0,40%
Wing In Ground	7707	0,27%
Special Category	129394	4,54%
High-Speed Craft	50283	1,76%
Special Category	233405	8,18%
Passenger	123210	4,32%

Continues in next page.

Continued from previous page.

Vessel Type	# Positions	% Positions
Cargo	1495404	52,42%
Tanker	611904	21,45%
Other	190045	6,66%

Analyzing the trajectories of the vessels, the following statistics can also be obtained:

- The dataset contains a total of 9394 trajectories;
- The average duration of a trajectory is 1.5 days;
- The average number of trajectories by day is 1085.

4.2 Visual Variables

AIS data contains different features that need to be visualized simultaneously, which requires the definition of specific visual variables for these features in order to display them. For the developed visual platform, the following variables were considered:

- The position is used to represent the latitude and longitude coordinates of each vessel. This requires a projection that converts the original geographic coordinates to cartesian ones (described in section 4.3);
- The color is used to identify the type of vessels, creating an associative effect. As Figure 4.1 shows, the following colors are used for each type:
 - Light blue is for general and unspecified vessels (commonly called reserved);
 - Red is for wing in ground, search and rescue vessels;
 - Green is for a series of special vessels like fishing ones, tugs, dredgers, military operations, sailing, among others;
 - Purple is for high-speed craft vessels;
 - Orange is for another series of special vessels like law enforcement ones, anti-pollution, medical transport, among others;
 - Yellow is for all types of passenger vessels;
 - Dark blue is for all types of cargo vessels;
 - Pink is for all types of tanker vessels;
 - Cyan is for all types of vessels that were not mentioned previously.

This color palette was chosen in compliance with two restrictions: the hue values of each color could not be confused with any of the remaining ones and the levels

of saturation and brightness needed to be balanced between all the colors, in order to avoid the idea that some types of vessels are more important than others. For the remaining components of the platform (controls, map, and others), grey colors, with different levels of intensity on the spectrum that goes from white to black, were used;



Figure 4.1: Color palette for vessel type representation.

- The value (lightness) is used to emphasize the areas where the vessels navigate more often. Each position is drawn with an opacity between 0.05 and 0.5 (the value is adjustable through a slider but the default is 0.05), and when several positions overlap their color is added and the opacity increases, which creates the notion that the density of traffic is higher in those positions. An example of this concept is visible in Figure 4.2, where one can see that the majority of the traffic is created by cargo and tanker vessels and passes through the two main corridors of the Portuguese maritime area.

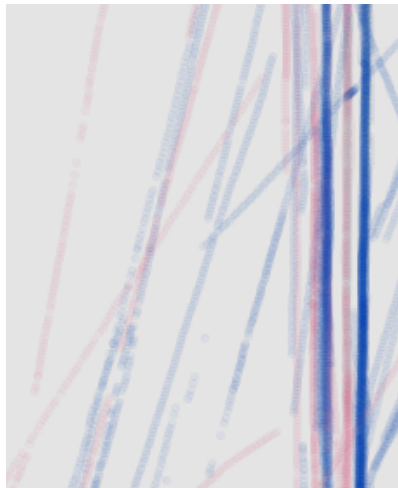


Figure 4.2: Usage example of the visual variable value.

- The motion, combined with other variables, is used to represent the trajectories of the vessels over time through animation. This approach is described with detail on section 4.5, but the effect caused in each of the used variables is the following:
 - The position changes according to the geographic position of the vessel on the respective time frame;

- The orientation is inferred as a consequence of the position changes over the time frames;
- The value is used to display the speed of the vessels over time through different levels of the color black.
- The shape is not used for differentiation proposes because it would not be perceptible when several vessels overlap on the same position. Therefore a constant shape of an ellipse is used to represent the positions;
- The size is constant between all the positions because otherwise it would create the notion that some positions are more important than others. The circle of each position has a radius of 2 pixels, that is increased to 7 pixels in the animation mode.
- The texture is not used for any propose.

4.3 Navigation

Each AIS position has several features and is represented geographically by a longitude and latitude. These values are the ones that dictate where in the screen is the visual mark of the respective position going to appear. However, the visual mark position is defined through a pair of values (x, y) on the cartesian system, which requires a conversion between the original pair $(longitude, latitude)$ to these final values. Several projection models are available for this propose, but one of the most used is the Mercator projection (Maling, 2013). This projection is cylindrical and conformal, which means that the Earth angles are preserved in the generated positions, and, because of that, it presents a low level of distortion (in general) that increases in positions far from the equator (Maling, 2013). The more important interactive maps platforms, like Google Maps ⁶ and OpenStreetMap ⁷, use a variant of this projection called Spherical Mercator projection (or Web Mercator), which assumes that the Earth is spherical instead on ellipsoidal, mainly due to calculations simplification. For these reasons, this projection was used to convert the geographical coordinates into cartesian ones.

Another important aspect of this projection is that it allows an easy integration of the zoom concept, which is a mandatory feature to navigate on the AIS data. The zoom level is defined by an integer that is multiplied by the cartesian coordinates. This approach is simple but requires an adjustment of the positions. When the zoom increases the generated cartesian coordinates are too big to fit the screen and need to be shifted back to the display area. Therefore, when the zoom changes, the amount of shifting required needs to be recalculated. This adjustment is made by selecting the first position on the map, calculating the distance between the position before and after the zoom update, and adding that distance to the required shifting. These shifting values are also used for screen navigation in terms of moving to the left, right, up and down. When a movement is made

⁶Available at <https://www.google.com/maps>

⁷Available at <https://www.openstreetmap.org>

in any of these directions the shift values are adjusted 100 pixels. For example, when a left movement is made the horizontal shift is incremented 100 units. The zoom in, zoom out and movement actions can be performed through buttons or keyboard shortcuts.

Considering that the AIS positions are displayed in a 2D plane, its important to contextualize the data regarding its location on Earth. This can be achieved by displaying points of interest that are geographical recognized by the general population. Therefore, the boundaries of Portugal, Spain and Africa are presented and the respective shapes are filled with a dark grey color, which indicates that these areas are land. These boundaries were obtained as polygons on the shapefile format from the GADM ⁸ and GeoTech ⁹ websites. Shapelines are a common way to represent geographic data as points, lines and polygons, and are widely used by systems that deal with this type of data. However, to integrate these boundaries into the developed platform they needed to be represented as a set of geographical positions with a longitude and latitude. Therefore, a tool called QGIS ¹⁰ was used to import the shapefiles, convert the polygons to a set of points representative of the shape and save them into CSV files. These files are then loaded to the platform and the same projection, zoom and movement strategies described above are applied to these positions. This approach is shown in Figure 4.3. No more areas were considered because the target of the platform is the Portuguese maritime zone.

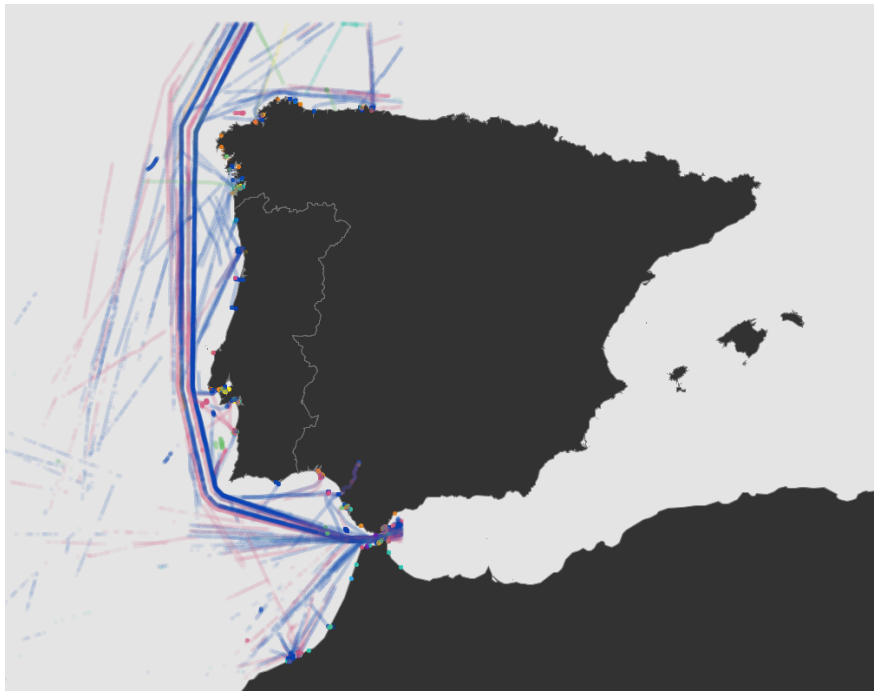


Figure 4.3: Portugal, Spain and Africa boundaries for data contextualization.

Each position drawn on the screen is often part of a line of points from the same vessel. This line represents a trajectory of that vessel and shows the entire route followed by it

⁸ Available at <https://gadm.org/maps.html>

⁹ Available at <http://techcenter.jefferson.kctcs.edu/data>

¹⁰ Available at <https://www.qgis.org>

during the selected time period. Several analysis and operations are done at this level and, for that reason, an approach for selecting trajectories was developed. When a visible point is selected through a mouse click, it is converted back to a latitude and longitude. Based on these values, each antecedent and subsequent position of the same vessel are analyzed and if the time gap between each of them and the selected point is below a given threshold (by default 30 minutes, a value obtained through experimentation) they are considered part of the trajectory. This process is repeated for the remaining positions of the vessel in both directions and stops on the first point where the time gap is above the threshold. In terms of visualization, the selected trajectory is displayed as before but an additional black line connecting all of its points is drawn. However, the remaining points are faded out by changing their color to dark grey with the general level of opacity. Notice that multiple trajectories can be selected simultaneously. An example of multiple selected trajectories is presented on Figure 4.4, where 3 cargo and 3 tanker trajectories are selected while the remaining are faded out.

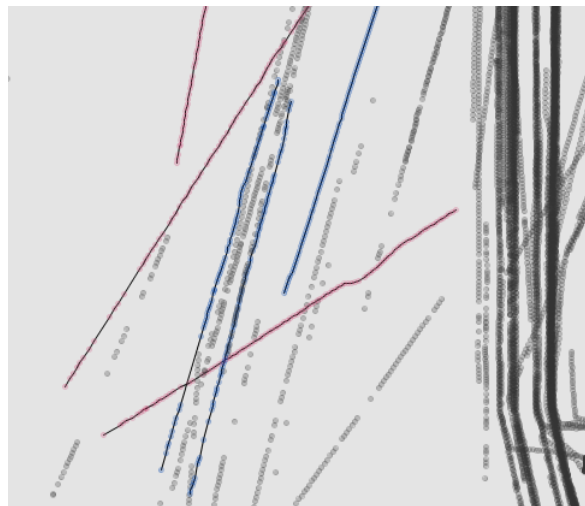


Figure 4.4: An example of several selected trajectories.

4.4 Data Filters

Depending on the goal of the analysis made to the data, there is a need to ignore and remove from the screen unnecessary data. To fulfill this need several data filters were implemented that cover the more important features of AIS positions. These filters are:

- A date range selector, which allows the selection of the start and end date for the displayed data. The default range is the first day with data;
- A hour range selector, which allows the selection of the start and end hours, within the previously selected dates, for the displayed data. The default range is the entire day (24 hours);

- A multiple selector for the vessels types, which allows the selection of the ones that are displayed on the screen. By default all types are visible;
- A slider for the minimum number of points by vessel, which ensures that vessels with less reported positions than the defined value, within the date and time period in analysis, are not displayed. The default value is 0, making the filter inactive;
- A slider for the maximum time gap between points of the same vessels, used to define the time gap required for the trajectories selection approach described on section 4.3. The default value is 30 minutes.

Figure 4.5 presents the controls of the data filters described above. After the adjustment of the controls, the new filters are applied explicitly. The two small arrow buttons on the top allow a navigation on the data, day by day, and their effect is applied immediately.

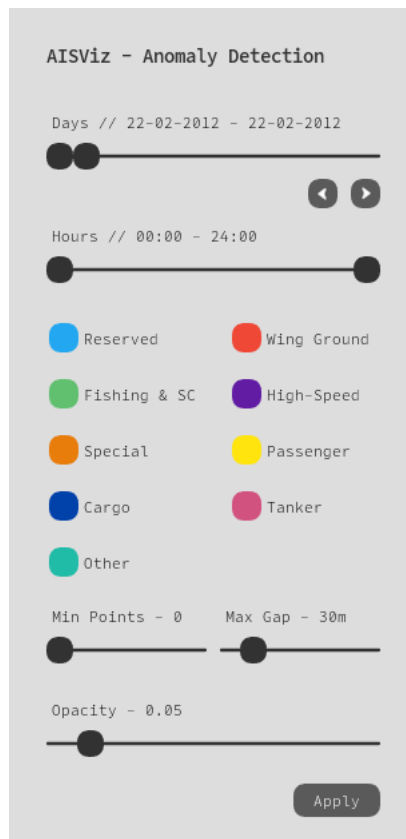


Figure 4.5: The controls of the data filters.

4.5 Trajectories Animation

Visualizing the vessels trajectories with a static approach has several limitations regarding the amount of features that can be displayed. Considering the visual variables already described, in a static environment only the positions of the vessel and its type can be

displayed. This means that two very important features for behavior analysis are ignored: the speed and the direction. To present these two features the motion variable, combined with the position and the orientation, was introduced and implemented through an animation approach that allows the visualization of the vessels moving over time.

For the animation to be performed, the period in analysis is divided into 15 minutes frames. This interval was defined because it includes more than one position by vessel (often 2 to 4 positions), which allows a better understanding of the trajectories evolution in terms of direction. Each frame contains the vessels positions from its time interval and the ones from the previous frames, and is displayed a quarter of a second after the previous one, creating the desired motion effect. Each vessel position is drawn through an ellipse with a radius of 7 pixels and using the color pallet already presented. The positions from the current time interval are drawn with an opacity of 100% but the ones that are from previous frames are drawn black with an opacity of only 7.5%. This creates a trace effect that allows the perception of how the vessels are moving over time without losing track of their position in the current frame.

The vessels new positions and directions changes are automatically displayed through the animation as a consequence of the motion effect. The points added in each frame are enough to understand the direction of a vessel because these new points will change the orientation of the trajectory. However, the speed attribute requires further efforts to be visible. An approach was developed where the speed is represented by the accumulative opacity of the vessels trace over time. Assuming that a vessel reports its positions in a fixed period of time (for instance, each 5 minutes), if this vessel is moving slowly the reported positions will be very close to each other, or even the same if it is stopped. However, if the vessel is moving fast these positions will be far from each other. When all the positions are drawn at the same time on the trace of the vessel, the ones that are overlaid will generate a higher opacity because their individual colors are blended. This means that the areas of the trajectories where the transparency of the trace is lower are the ones where the vessels are moving slower, because more points were overlaid for this effect to happen. On the contrary, a trace with a higher transparency corresponds to a vessel moving faster.

As stated before, the speed visualization approach only works if the time-span between each position of a vessel is fixed, which is a problem because the AIS communication periods are not consistent. To fix this issue a cubic spline interpolation is applied to every trajectory to generate the missing positions. As stated in the literature (Sang et al., 2012, Zhang et al., 2017), this type of interpolation is the one that adjusts better to the reconstruction of AIS trajectories, offering just some limitations in the presence of very tight curves. This interpolation creates a piecewise function, which means that it defines several small sub-intervals through the domain of x and has an individual polynomial of degree 3 for each one. This aspect is important to make the final function more smooth and better suitable for curves. The interpolation is made individually for the latitude and the longitude, being these variables the output y of the generated functions and the timestamp in seconds the input x . The polynomial coefficients of both functions are calculated with

all the AIS positions of the respective vessel. Each of these positions is then compared with the one immediately after and, if the time gap between them is over 5 minutes, new positions are generated through the interpolated functions, with intervals of also 5 minutes, until the gap is filled. This 5 minutes period was supported by existent works from the literature. Figure 4.6 shows an example of an animation with 4 vessels sailing with different speeds. It is visible that the tanker vessel (the pink one) in the middle is sailing with a low speed, maintaining a route in a very small area, while the other 2 tankers sailed faster. The cargo (dark blue one) started slow but increased the speed roughly in the middle of the trajectory.

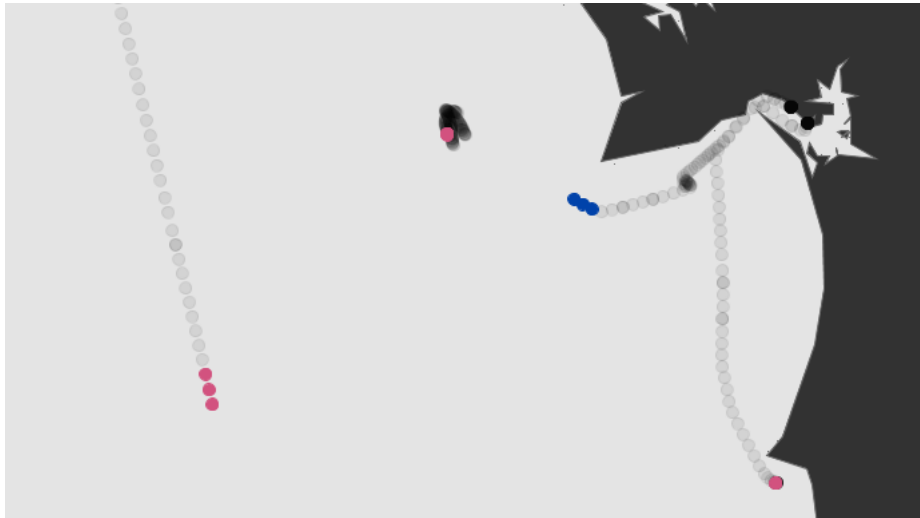


Figure 4.6: An example of an animation with vessels sailing with different speeds.

The animation can be considered a video that is rendered on the fly. Therefore, the common video controls are presented, namely a play/pause button and a time bar to navigate on the frames.

4.6 Analytics

When considering the AIS data displayed, there are some statistics that are helpful for analysis proposes but can not be easily integrated in a visual scheme. For that reason, this statistics are calculated and presented as text through a toggle button. The more important ones are:

- Number of trajectories;
- Number of vessels;
- Mean trajectory time of all vessels;
- Number of trajectories by vessel type;

- Number of vessels by type.

Figure 4.7 shows an example of the statistics described above for the first day of data.

```
AISViz - Anomaly Detection
Statistics

929 Trajectories // 929 Vessels
Mean Trajectory Time // 09H29m
Vessels & Trajectories by Type
- Reserved // 3 & 3
- Wing In Grnd // 2 & 2
- Fishing & SC // 36 & 36
- High-Speed // 11 & 11
- Special // 81 & 81
- Passenger // 37 & 37
- Cargo // 485 & 485
- Tanker // 207 & 207
- Other // 67 & 67
```

Figure 4.7: An example of the statistics for the first day of data.

These statistics take in consideration only the vessels and respective positions that are displayed on screen, which means that they need to be updated each time the data filters change. To avoid an impact on the performance of the platform, and considering that this information is typically secondary for the analysis, this update is made on the background through a different thread. During the update time, a message of loading is presented on the same place of the statistics.

Other information that sometimes may be helpful is the details of the selected trajectories. As described before, the selected trajectories have a visual demarcation from the remaining positions, but the visual representation is only able to present some of the features from the data. Therefore, when a trajectory is selected, the following information is presented after the general statistics:

- MMSI of the vessel;
- Name of the vessel (this information is inserted manually by an operator and can be wrong or incomplete in some cases);
- General type of the vessel, followed by the complete type identification;
- Start date and time of the trajectory;
- Start day of the week of the trajectory;
- End date and time of the trajectory;

- End day of the week of the trajectory;
- Number of reported positions that constitute the trajectory.

In the scenario where several trajectories are selected, only the information from the last one is displayed.

Finally, a measure tool, that is able to calculate the line distance between two positions on the map, was also included in the platform. When the tool is enabled, the first point of the line is fixed through a mouse click. After that, by moving the mouse the final point of the line is adjusted and the distance is recalculated. To fix the final point another mouse click on the wanted position is required. The line distance is calculated and presented in two ways:

- Using the Haversine formula, which is able to calculate the real distance in kilometers from two geographical positions defined by the longitude and latitude values;
- Using the Euclidean formula, which calculates the distance between the two (x, y) points of the screen considering an Euclidean space.

The usage of these distances, particularly the Haversine one, may be useful to understand the real distance between two positions or events of interest, considering that the notion of proportion between the projected points and the real positions may sometimes be lost. Figure 4.8 shows an usage example of this tool.

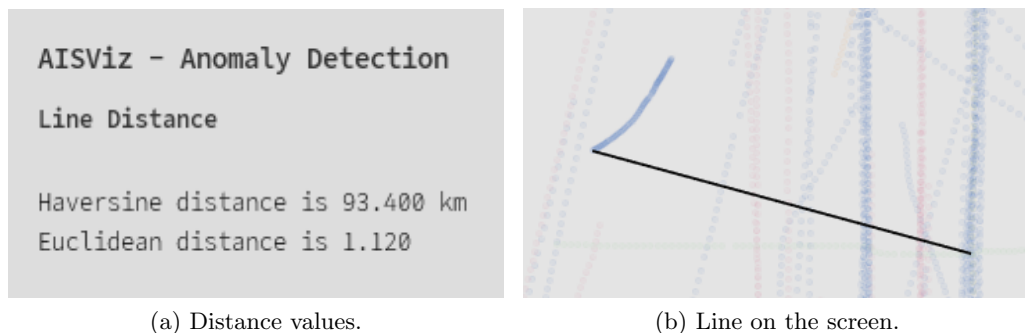


Figure 4.8: An example of the line distance tool.

4.7 Rendering Optimizations

AIS positions are reported by each vessel very frequently¹¹ and, when analyzing the data of several days from an entire maritime area, performance issues arise because of the quantity of data that needs to be processed. The developed platform and respective visualization strategies were implemented with the graphics API Processing 3 and suffered from these

¹¹Depending on the type of the vessel, its speed and course, the period may vary between 2 seconds and 3 minutes, assuming that all messages are sent and received.

performance issues in several occasions. Effective measures to solve these issues were implemented and the more important ones are:

- A double buffering approach with cache to avoid multiple drawing of the same data. The used graphics API already implements a double buffering mechanism but it only avoids screen flickering, because the API forces each shape to be drawn in each new frame. This default behavior does not work well when big quantities of shapes are drawn simultaneously because the amount of time needed for the process to be completed creates a lag effect that eventually freezes the application. To fix this issue the complete drawing process is made in background through a different thread. The result is saved in a "virtual image" that has the same width and length of the screen and holds the color for each rendered pixel. The resulting image is then swapped with the main one that is currently being displayed by the graphics API in the screen. This swapping process is synchronized with a mutex in order to avoid the usage of the main image for reading proposes while it is being updated. This approach fixes the issue because the main image is kept in cache and displayed as-is on each new frame by the graphics API, which removes the lag effect and consequent freeze problem. Also, the background thread is only executed when the data to be displayed changes (a new data filter is applied, a zoom action is performed, etc.), which avoids unnecessary renderings of the same data;
- A partial load of the AIS data through daily chunks and a time window of 1 day, in order to avoid large periods of time spent by the data load and filtering tasks. With all the dataset on a CSV file, a script was developed in Python to divide it into daily chunks where each new CSV file contains the entire data for a specific day. The implemented approach on the platform takes advantage of these chunks by loading only the ones needed for the date and time period defined through the data filter, with the addition of 1 day both before and after the period. This addition is important because the platform favors a daily analysis of the data, for performance reasons and also because an analysis made with date from more than 1 day may be negatively influenced by the fact that the date is only visible through the animation and, eventually, by some visual clutter. Therefore, if the data of the day x is being displayed, the most probable days to be analyzed next are the $x + 1$ or $x - 1$. With the loading of 1 day before and after the current period, the data from these two days is already in the memory, and the loading task is therefore skipped for these scenarios. This will increase the update speed because only the rendering actions are required for these cases. When these actions terminate and the data is being displayed, the loading task for the new period that includes the new additional days starts executing on the background;
- A caching mechanism by different types of rendering, which avoids the execution of partial tasks of the rendering process that are unnecessary. When a new rendering of the data is required, depending on the action that originated it, different intermediate tasks may be necessary:

- When the data filters change a complete render (type 1) is necessary, and this process consists on filtering the data, projecting the pair (*longitude, latitude*) of each position to the respective cartesian coordinates (x, y) and drawing the shapes of these positions on the new image;
- When a change in the zoom of the map occurs only the projection and drawing steps are required (type 2), because no changes on the filtered data occurred;
- When a movement on map occurs or any other action that does not change the level of zoom and the data filters (i.e. toggling the visibility of a specific component), only the drawing step is required (type 3) because there are no changes on the filtered data and neither on the positions. As described before, the movement updates are made through a horizontal and vertical shift that are not dependent of the map projection.

Therefore, these 3 scenarios were implemented and when a new render is required the type must be specified. The first render is always the type 1, and a state is saved with the filtered positions and the projected points. When another type 1 render occurs this state is updated, but if a type 2 is executed only the projected points are. This ensures that the cached state is always up to date. Considering that the majority of the renderings are from the type 3, this approach has a good impact on the performance of the platform;

- A caching mechanism for the rendered frames of the animation of trajectories, avoiding multiple and unnecessary renderings of the same frames. The animation works in loop mode, which means that it restarts after the last frame. Consequently each frame may be displayed more than once, depending on the number of times that the animation is repeated. To avoid multiple rendering of the same data, each frame is rendered once and saved for future usage. This behavior is also applied when the visible frame is chosen by the time bar control;
- A point reduction approach for the map boundaries, in order to optimize the drawing time of these shapes without losing quality. Each of the map boundaries extracted from the shapefiles is composed by a big quantity of points in order to create the shape of the polygon as accurately as possible. However, this makes the drawing process slower because of the high number of vertexes that the shapes contain. To avoid this problem the Douglas–Peucker algorithm is applied to all the boundaries points with the propose of reducing them for the minimum quantity that is able to represent the shapes accurately. Before the application of the algorithm, all boundaries combined in a total of 554101 points, and after the reduction this number was reduced to just 5642, which is approximately 1% of the original quantity. This process reduced the drawing time without compromising the original shapes;
- Regarding the developed 3D strategies (described on section 5.3), an approach to reuse the shapes was implemented to avoid performance issues that caused a lag effect. Each position on the 3D plane is represented through a sphere, and the

creation of 3D shapes is a particularly heavy task because they are composed by more vertices when compared with 2D shapes. With this issue, the creation of multiple spheres (one for each visible position) was creating a lag effect on the screen. To fix this problem only 9 spheres are created in the beginning of the rendering process, each one filled with the color associated with a vessel type. Then, to draw a position, the sphere with the color that matched the type of the vessel that reported it was selected, translated to the correct coordinates and drawn. The lag effect was fixed with this reuse approach and the rendering time was also improved.

4.8 Interface Evolution

The focus of the developed platform are the visualization strategies, but without a good user experience it may be difficult to exploit such strategies. The platform was developed in Java using the graphics API Processing 3 for the visualization components. The initial approach was to use a library called ControlP5¹² for the GUI controls and the color palette was obtained from the ColorBrewer¹³ web site. The remaining options were kept with the default values. A print screen of the platform on this initial state is presented on Figure 4.9.

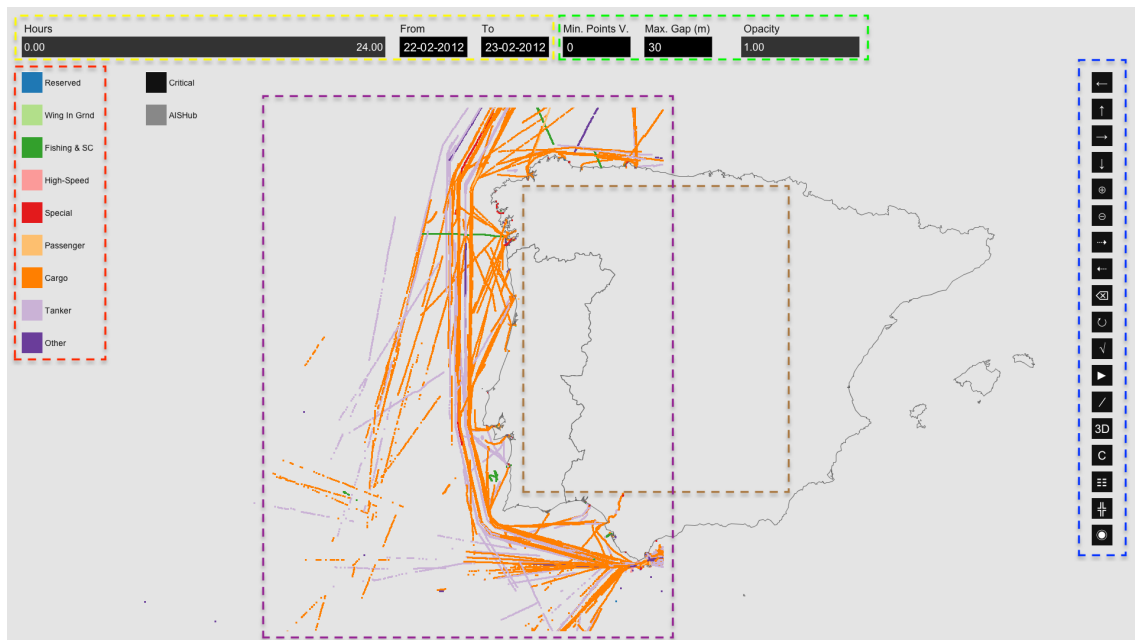


Figure 4.9: Initial interface.

As the print screen shows, the interface had several problems. The following ones were addressed:

¹² Available at <http://www.sojamo.de/libraries/controlP5>

¹³ Available at <http://colorbrewer2.org>

- The initial color palette created a dependency effect between the different vessels types. For example, the passenger vessels looked like a sub-type of the cargo vessels. The palette was updated to the one already presented on section 4.2 and this issue was solved. In this new palette an effort was also made to use colors distinguishable with low levels of opacity;
- All the controls were shifted to the top-left of the screen because it is the first area where the user looks;
- To gain more screen space for the visualization itself, the button bar that contains the main features of the platform is always visible but the remaining controls can be collapsed through a button placed on right-bottom corner of their section. These controls are mainly associated to the data filters and the user needs to work with them less often;
- The collapsible area contains the base data filters but, when an option that contains additional filters is activated, this area stops showing the base controls and instead shows the ones associated with the option until it is deactivated;
- The ControlP5 library suffered an extensive refactoring in order to improve its design and also to add missing features that were important to the platform, like adjustable labels according to the control value;
- The text input controls were abolished and replaced by more user-friendly ones like sliders and ranges;
- The cursors were updated for the controls and, instead of having a general pointer for everything, the clickable controls have a hand cursor and the sliders have a double arrow;
- The type of letter was replaced by the monospaced Office Code Pro¹⁴;
- The icons for the actions of the button bar were updated to more suggestive ones;
- On the map, the areas of land were distinguished by being colored with a dark grey.

The changes described above resulted on the final interface displayed on Figure 4.10. The different areas of the platform are marked with rectangles using the same colors as the ones from the initial interface, in order to allow an easier visual comparison between them. One can see that in the final interface not only the design is improved but the disposition of the controls is more user-friendly, the visualization itself gained more space for the visual marks, the land is clearly distinguishable from the sea and the controls in general are less error-prone.

¹⁴Available at <https://github.com/nathco/Office-Code-Pro>

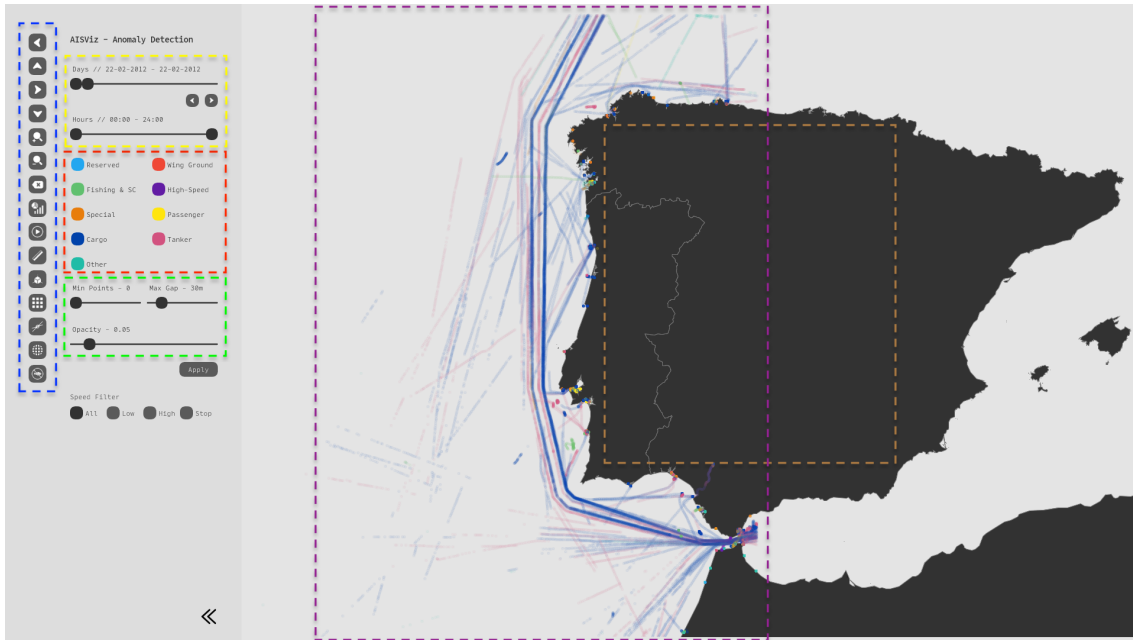


Figure 4.10: Final version of the interface.

4.9 Implementation

The implementation of the entire system was made with Java and Python using the architecture presented on Figure 4.11. Python was used for all the major data processing tasks, while Java was used for the platform itself and for some specific data related tasks, like processing the raw AIS messages, and also for some machine learning tasks, like the Density Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm.

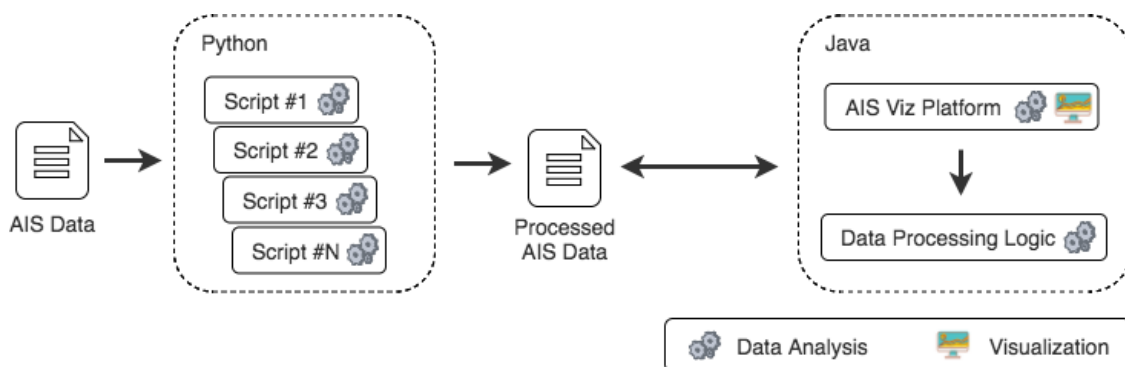


Figure 4.11: General architecture of the implementation.

Regarding the development in Java, it was mainly divided into two components: one responsible for the visual strategies and rendering, and another one containing all the data processing logic. The more important classes implemented for both components are de-

scribed on Figures 4.12 and 4.13, respectively. These images present a brief description for each class and groups them by context. Figure 4.14 presents the usage relation between the classes through arrows that show the direction. The combined effort of this implementation resulted on more than 16500 lines of code. As stated before, the used graphics API was the Processing 3¹⁵ with both the P3D and the JAVA2D render. The first one is based on OpenGL and the last one is a proprietary implementation of the Processing Foundation. Besides the implementation effort already presented, the following libraries were used for very specific features:

- An implementation of the Graham’s Scan algorithm (Graham, 1972) in Java¹⁶;
- An implementation of the ray casting algorithm (Haines, 1994) in Java¹⁷;
- An implementation of the Douglas–Peucker algorithm in Java¹⁸;
- PeasyCam, a Processing 3 library to control the 3D camera using the mouse¹⁹.

Also, an implementation of the DBSCAN algorithm²⁰ was used as a starting point for the necessary implementation, but suffered major changes like the adaption of the code to use custom classes and the addition of new density-reachable metrics with the necessary distances.

Regarding the developed Python scripts, 30 scripts were implemented with a combined effort of more than 3900 lines of code. These scripts can be grouped into the following categories:

- Data processing tasks - 11 scripts;
- Machine learning and related tasks - 8 scripts;
- Statistical tasks - 11 scripts.

¹⁵Available at <https://processing.org>

¹⁶Available at <https://github.com/bkiers/GrahamScan>

¹⁷Available at <https://github.com/sromku/polygon-contains-point>

¹⁸Available at <https://github.com/LukaszWiktor/series-reducer>

¹⁹Available at <https://github.com/jdf/peasycam>

²⁰Available at <https://github.com/chrfrantz/DBSCAN>

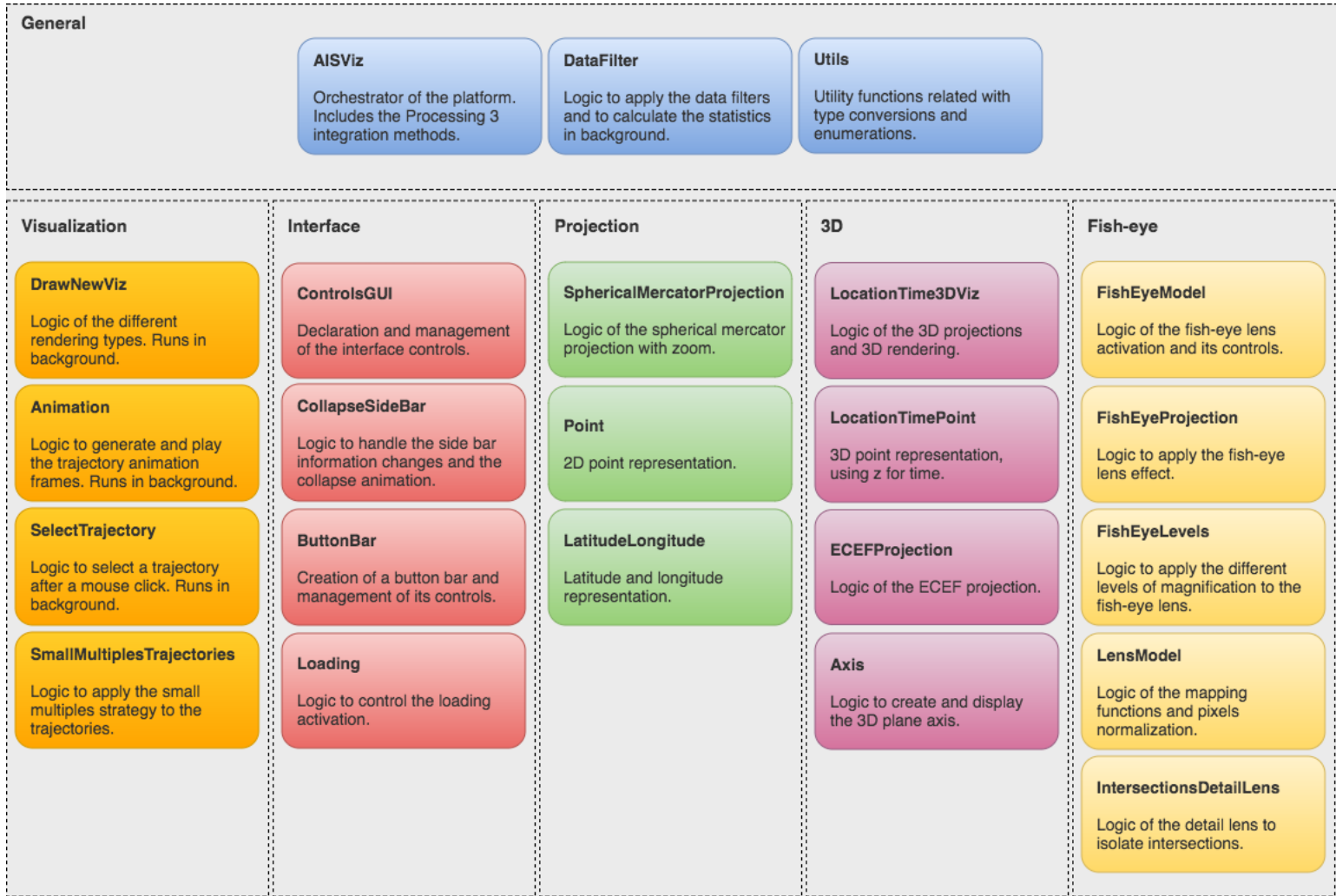


Figure 4.12: More important classes implemented on the AIS Viz Platform.

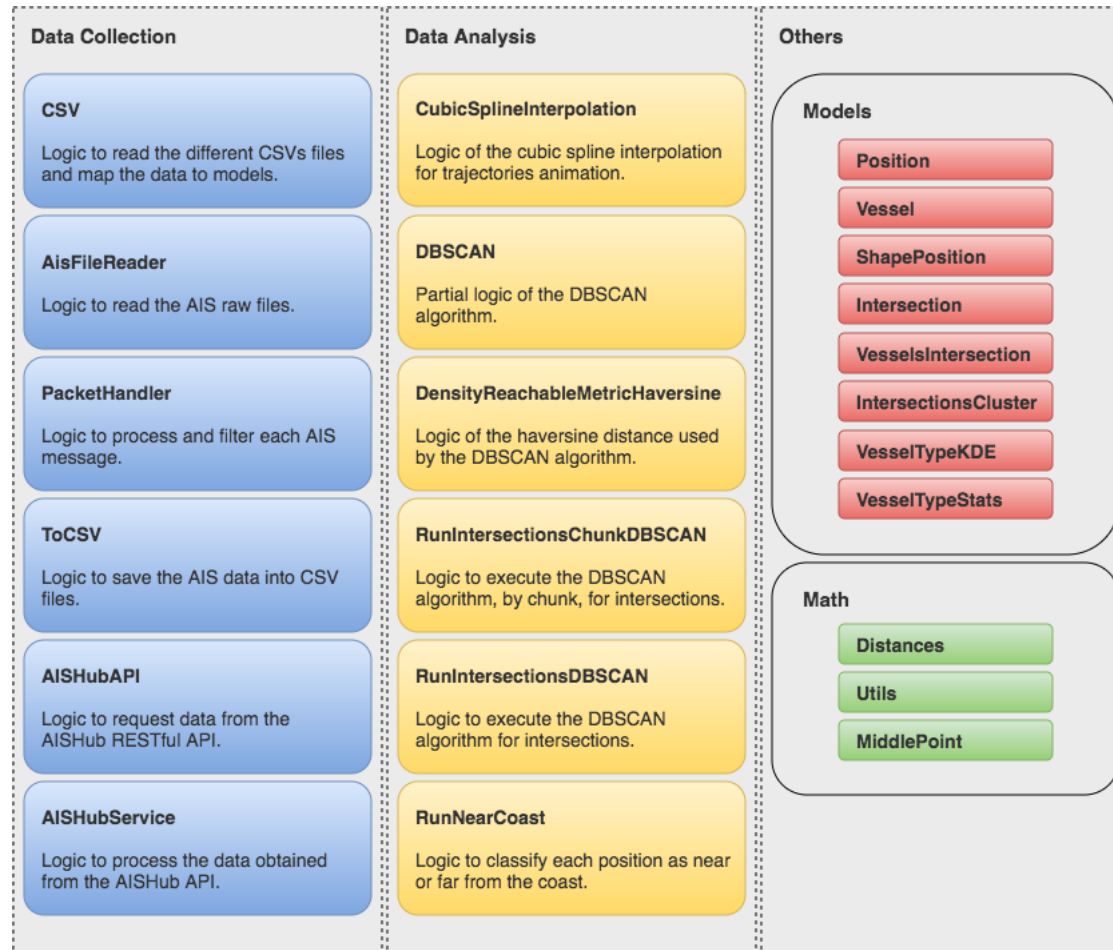


Figure 4.13: More important classes implemented on the data processing logic component.

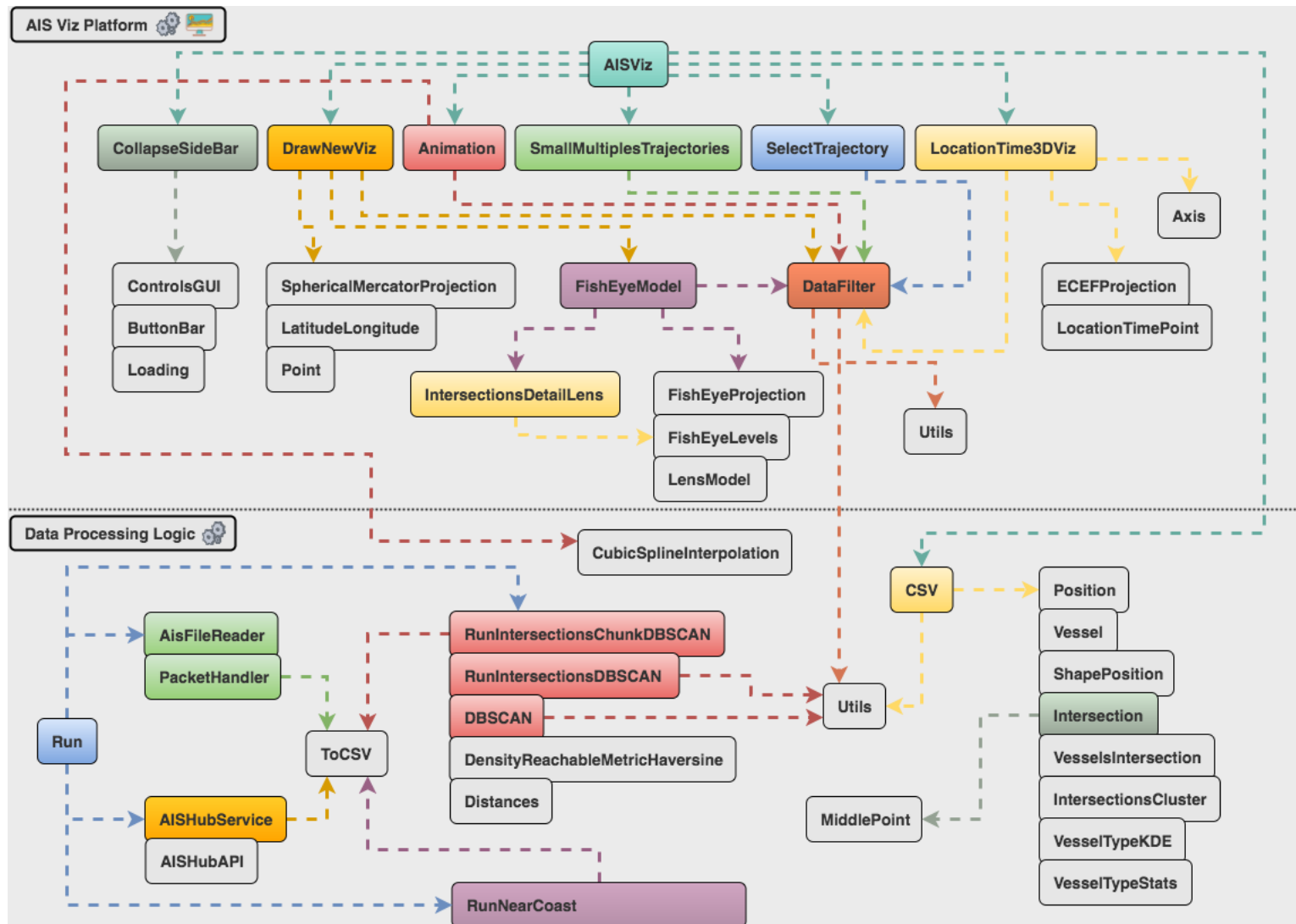


Figure 4.14: Usage relation between the more important classes of both components.

Chapter 5

Anomalous Behavior Analysis

The propose of the developed platform was to detect anomalous behaviors that were previously identified by domain experts. These behaviors were grouped into three categories:

- Intersections, where are included all the behaviors related with intersections between vessels;
- Speed outliers, which includes the behaviors related with vessels sailing at an abnormal speed;
- Forbidden fishing, which is a subcategory of the previous one that also includes the behaviors related with fishing vessels sailing in areas where fishing is forbidden.

Different visualization strategies, assisted by data processing tasks, statistics and other techniques, were developed for each of these categories. The following sections 5.1 and 5.2 describe these strategies. Finally, the section 5.3 presents the explorations made of the 3D plane to include the representation of the time evolution.

5.1 Intersections Based Behaviors

The proposed approach for intersections' analysis, presented on Figure 5.1, is composed by data processing tasks that extract the vessels intersections from raw Automatic Identification System (AIS) data, a visual search technique that uses a magnified fish-eye lens, a visual categorization of abnormality in high density areas and an analysis of repeated intersections over time. Each of these components of the approach are described in the following subsections.

5.1.1 Data Processing Tasks

Regarding the extraction of the intersections, the first necessary task is to remove the duplicated positions of each vessel. These duplicates can occur when vessels are stopped

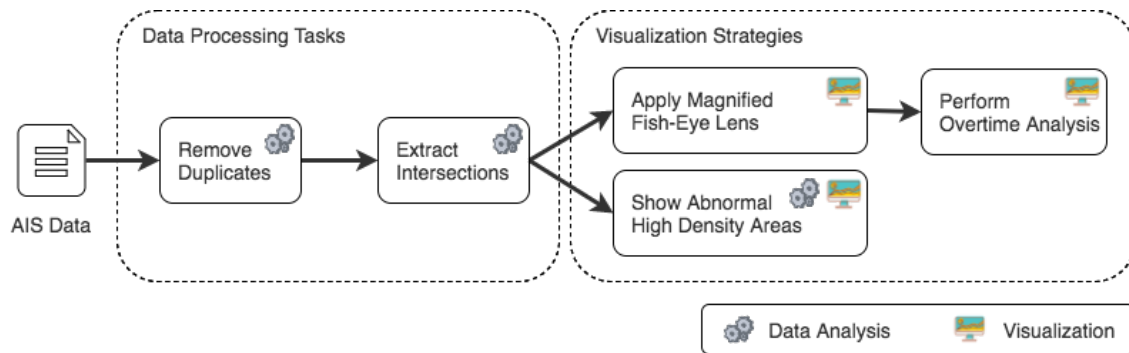


Figure 5.1: Proposed approach for intersections' analysis.

and keep reporting their positions or by malfunctions on the AIS communication system. Therefore, this is an important step because repeated AIS positions will lead to the detection of multiple intersections that in reality correspond to just a single one. For this task the Algorithm 1 was developed. The approach is to isolate the positions of each vessel and detect the ones that are within a minimum time and distance gaps, being these ones the so-called repeated. The time gap is calculated by the absolute difference between the time-stamps of both positions, which are in the UNIX Epoch time format. To calculate the distance between the same positions, considering that it is between two locations on Earth, the Haversine formula is used. This algorithm requires as parameters the minimum time and distance gap, and these values were obtained through experimentation being 15 seconds and 1 meter, respectively. The implementation was made in Python with a multithread environment to allow the processing of several days of data simultaneously.

The second and most important task is to detect and extract each intersection between different vessels. For this propose the Algorithm 2 was developed, and it also requires as parameters the minimum time and distance gaps. The approach considers that an intersection occurs when two positions of different vessels have a time and distance gaps below the minimum values passed as parameters. Both gaps are calculated using the same approach as in the Algorithm 1, that was presented above. The values for the parameters were also obtained through experimentation and they are 30 minutes and 1 kilometer, respectively. The implementation was also made in Python with a multithread environment, but several optimizations of the implementation were required. The initial approach was to implement the algorithm using the common iterative process with loops. However, the algorithm had major performance issues related with the amount of data that was being processed. To fix this issue the loops were "converted" to a vectorized approach, where the data was put into matrices, the time and distance gaps between the positions were calculated through common vector operations and the intersections were filtered from the results. This implementation has the exact same logic of the Algorithm 2 but revealed to run much faster. This algorithm does not take in account the direction of the intersection, considering intersections A to B and B to A as different, when they are actually the same. The algorithm could deal with this issue but that would have impact on its performance and, for that reason, the issue is fixed in a third task. A new algorithm

Algorithm 1 Remove the duplicated positions of every vessel in each day

```

1: function REMOVEDUPLICATEDPOSITIONS(data, minTimeGap, minDistanceGap)
2:   for each day of data do
3:     aisPositionsToKeep  $\leftarrow$  {}
4:     for each vessel of the current day do
5:       aisPositionsByVessel  $\leftarrow$  {}
6:       for each AIS position (nP) of the current vessel do
7:         foundEqualPosition  $\leftarrow$  false
8:         for each saved position (sP) in aisPositionsByVessel do
9:           if  $\text{abs}(nP.Timestamp - sP.Timestamp) < minTimeGap$  then
10:            distance  $\leftarrow$  haversine distance between nP and sP
11:            if  $distance < minDistanceGap$  then
12:              foundEqualPosition  $\leftarrow$  true
13:              break the current for loop
14:            end if
15:          end if
16:        end for
17:        if  $\neg foundEqualPosition$  then
18:          aisPositionsByVessel  $\leftarrow$  aisPositionsByVessel + nP
19:        end if
20:      end for
21:      aisPositionsToKeep  $\leftarrow$  aisPositionsToKeep + aisPositionsByVessel
22:    end for
23:    Save aisPositionsToKeep in a CSV file
24:  end for
25: end function

```

Algorithm 2 Extract intersections between different vessels in each day

```

1: function EXTRACTINTERSECTIONS(data, minTimeGap, minDistanceGap)
2:   for each day of data do
3:     dataToCompare  $\leftarrow$  positions of the current day + positions of the day be-
4:     fore within [midnight, midnight - minTimeGap] + positions of the day after within
5:     [midnight, midnight + minTimeGap]
6:     intersections  $\leftarrow$  {}
7:     for each AIS position (cP) of the current day do
8:       for each AIS position (oP) in dataToCompare do
9:         if  $cP.MMSI \neq oP.MMSI$  then
10:          timeGap  $\leftarrow$   $\text{abs}(cP.Timestamp - oP.Timestamp)$ 
11:          if  $timeGap < minTimeGap$  then
12:            distance  $\leftarrow$  haversine distance between cP and oP
13:            if  $distance < minDistanceGap$  then
14:              newIntersection  $\leftarrow$  {cP, oP, timeGap, distance}
15:              intersections  $\leftarrow$  intersections + newIntersection
16:            end if
17:          end if
18:        end if
19:      end for
20:    end for
21:    Save intersections in a CSV file
22:  end for
23: end function

```

iterates over the intersections, transforms them into sets and calculates a hash for each one. Considering that the set has no concept of order, the hash of the related intersections that have different directions will be the same. Finally the algorithm removes intersections with duplicated hashes, keeping just the first one.

With the application of these tasks in the presented order, the intersections for each day of raw AIS data will be extracted and saved in new CSV files.

5.1.2 Visual Search Through a Magnified Fish-Eye Lens

The extracted intersections are presented on the platform through black ellipses with a radius of 5 pixels and 75% of opacity. With all the AIS positions displayed in the same screen space it can become difficult to detect the intersections and even more difficult to analyze each one individually. Therefore a necessary step is to create the means that will allow and efficient search and focus of specific intersections. For this goal the usage of a fish-eye lens (Bettonvil, 2005) was considered. This type of lens applies a convex effect to the image, creating the illusion that it has the shape of a sphere, an effect commonly called barrel distortion. With this effect the center of the image becomes the focus, with the distortion from the sides being exaggerated with the increasing distance from the center. Consequently, the center of the image becomes more zoomed, with the cost of losing some resolution. When combining this type of lens with the movement of the mouse, it is possible to focus on specific AIS positions and intersections, and more easily search for points of interest. Different distortions with specific characteristics can be applied with a fish-eye lens through different mapping functions. The most popular ones are the equidistant, equisolid, orthographic and stereographic (Bettonvil, 2005).

Generalizing for any given image, the implementation of the fish-eye effect on the platform can be summarized in the following steps:

1. Each pixel position is normalized on a range between $[-r_x, r_x]$, being r_x the width radius of the fish-eye lens. The r_y for the height radius is calculated from r_x by keeping the original image aspect ratio. This transformation is necessary for a correct conversion of the values into polar coordinates, with the center of the image becoming $(x, y) = (0, 0)$.
2. Each normalized pixel position is then converted from cartesian coordinates to polar coordinates, first by calculating the distance with the formula on Equation 5.1, and then by calculating the angle of the direction using the formula on Equation 5.2. This last calculation takes in consideration the focal length f_l , which defines the angle of view and the zoom of the lens, and uses the \tan^{-1} function directly because the focal length is always greater than 0 ($f_l > 0$).

$$R_p = \sqrt{x_p^2 + y_p^2} \tag{5.1}$$

$$\theta = \tan^{-1} \left(\frac{R_p}{f_l} \right) \quad (5.2)$$

3. The mapping function that defines the distortion of each pixel position by calculating its new radial position is then applied. The formulas for the most popular mapping functions are displayed on Equations 5.3 to 5.6.

$$R_f = f_l * \theta, \text{ for the equidistant mapping} \quad (5.3)$$

$$R_f = 2f_l * \tan \left(\frac{\theta}{2} \right), \text{ for the stereographic mapping} \quad (5.4)$$

$$R_f = f_l * \sin(\theta), \text{ for the orthographic mapping} \quad (5.5)$$

$$R_f = 2f_l * \sin \left(\frac{\theta}{2} \right), \text{ for the equisolid mapping} \quad (5.6)$$

4. The new position of each pixel (x_f, y_f) is calculated from the original normalized position (x_p, y_p) by distorting it with the new distance from the center, using the formulas on Equations 5.7 and 5.8.

$$x_f = \begin{cases} x_p * \left(\frac{R_f}{R_p} \right) & R_f \geq 0 \\ 0 & R_f < 0 \end{cases} \quad (5.7)$$

$$y_f = \begin{cases} y_p * \left(\frac{R_f}{R_p} \right) & R_f \geq 0 \\ 0 & R_f < 0 \end{cases} \quad (5.8)$$

5. The new pixels positions are then denormalized to the original interval values comprehended between $[0, width]$ for x and $[0, height]$ for y ;
6. The resulting positions are real-valued and when they are used directly in the new image some gaps are created where no pixel values exist. This happens because the distortion applied to the original image created new positions that did not exist before. To fix this issue a pixel interpolation strategy needs to be applied. For this approach the 9-point interpolation scheme (Altera, 2008) was used because it offers a good computation performance, which is important considering that the fish-eye effect will be applied in real time, without significant losses of quality.

As described in the previous steps, two parameters are required for the fish-eye effect to be applied: the focal length and the width radius of the lens. Both parameters are correlated and their values were obtained through experimentation.

Regarding the usage of the fish-eye effect on the platform, before the implementation a mapping function must be chosen from the four that were described. An experiment was

conducted where the four functions were applied to the same image with two different focal length values: an intermediate value that only gives some zoom to the center of the image ($f_l = 4$) and an exaggerated value that increases a lot the zoom to the center of the image and distorts the boundaries more ($f_l = 2$). The resulting comparison is presented on Figure 5.2. Based on this experimentation the orthographic mapping was chosen, mainly because it is the one that better distorts the boundaries of the image through a more exaggerated curvature, which gives more focus and zoom to the center. Notice that on the examples with $f_l = 2$ ((c), (e), (g) and (i)), the zoom was much higher and the pixel interpolation method was unable to fill in all the missing gaps, which created the "black pixels" visible on the images.



(a) Normal rectilinear projection.

(b) Equidistant projection with $f_l = 4$.(c) Equidistant projection with $f_l = 2$.(d) Stereographic projection with $f_l = 4$.(e) Stereographic projection with $f_l = 2$.

Figure 5.2: Comparison between different mapping functions for the fish-eye effect using different focal length values and $r_x = 5$ for all the examples.

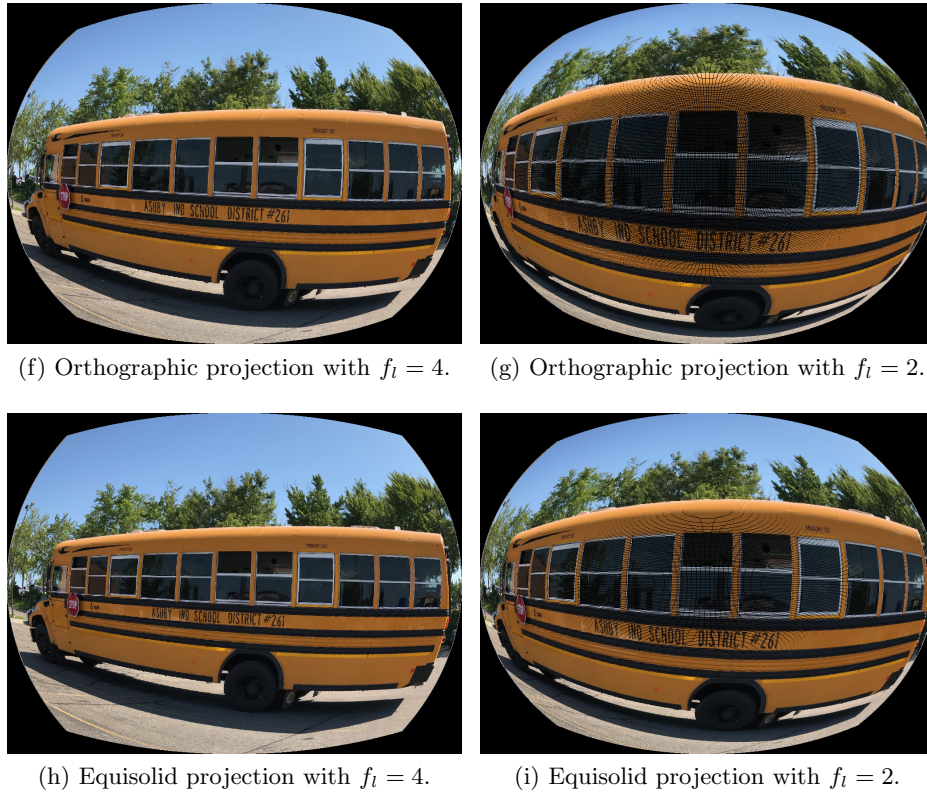


Figure 5.2: Comparison between different mapping functions for the fish-eye effect using different focal length values and $r_x = 5$ for all the examples. (*cont.*)

With the mapping function decided, the remaining parameters were tuned through experimentation and the chosen values were $f_l = 3$ and $r_x = 5$. The fish-eye effect is only applied to a specific area of interest, which is the area of the map that is going to be analyzed. This area is defined by the mouse position $(mouse_x, mouse_y)$ and by a radius $mouse_r$ that will define a circle around that position. To this specific circle the fish-eye effect is applied as described above. An example of this behavior is presented on Figure 5.3, with the original area displayed on the left and the same area with the fish-eye effect on the right.

The application of the fish-eye lens is a powerful way to focus and zoom on a specific area, but the level of magnification may not be enough for an efficient analysis. Increasing the zoom only by changing the focal length will lead to the "black pixel" problem presented on images (c), (e), (g) and (i) of Figure 5.2, and moreover this growth is limited. Therefore the proposed approach combines the fish-eye effect with levels of magnification. The key concept is to render the area of the map in analysis n times, where each time the level of zoom is increased by a scalar $zoom_g$ (similarly to the general zoom approach). The maximum number of levels can be predefined or it can be adjusted while the zoom increases, with this last strategy having a performance cost. Assuming that $zoom_c$ is the initial zoom of the area that corresponds to the level $n = 0$, the zoom of each n level is

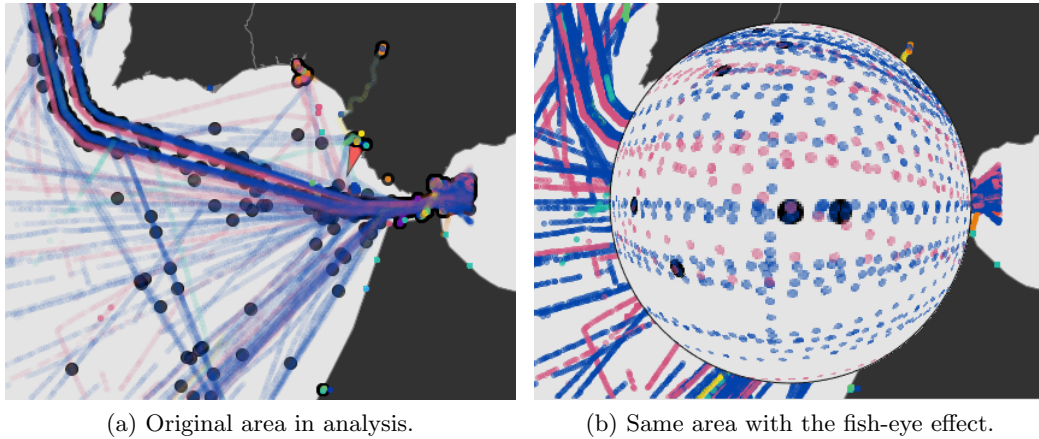


Figure 5.3: Example of the fish-eye effect applied and the intersections identified.

calculated using the formula on Equation 5.9.

$$zoom_l = zoom_c + (n * zoom_g) \quad (5.9)$$

An important aspect of this method is that the magnification effect needs to be applied to the center of the area in analysis, otherwise this area would start to change. To ensure this aspect the cartesian coordinates of each position in each level need to be shifted according to a point of reference from the base level ($n = 0$). This point of reference (x_{rb}, y_{rb}) is the AIS position that has the minimum euclidean distance between the fixed mouse position ($mouse_x, mouse_y$) and itself. Considering a point (x_p, y_p) on any given level and the point of reference (x_{rl}, y_{rl}) on the same level, the shift formulas are the ones on Equations 5.10 and 5.11. These formulas also have an adjustment between the point of reference in the base level and the mouse position because this point may not be exactly in the center of the area.

$$x_{new} = (x_p - x_{rl} + (x_{rb} - mouse_x) + mouse_r) \quad (5.10)$$

$$y_{new} = (y_p - y_{rl} + (y_{rb} - mouse_y) + mouse_r) \quad (5.11)$$

To navigate between the levels of magnification the area in analysis must be first fixed through a mouse click and, after that, two controls to change the zoom will appear. Each time the zoom level is updated, a new magnified image is applied to the fish-eye lens. This behavior is presented on Figure 5.4, displaying the desired effect on image (a) with the default magnification and the same effect on image (b) with the 3rd level of zoom.

When the area in analysis is fixed the intersections inside it are presented individually on a details lens located on the top-right corner of the screen. This details lens has the same level of zoom used on the fish-eye lens and allows the analysis of each intersection individually without the visual clutter. Figure 5.5 shows on image (a) all the intersections through the fish-eye lens and on image (b) one of the intersections isolated through the detail lens.

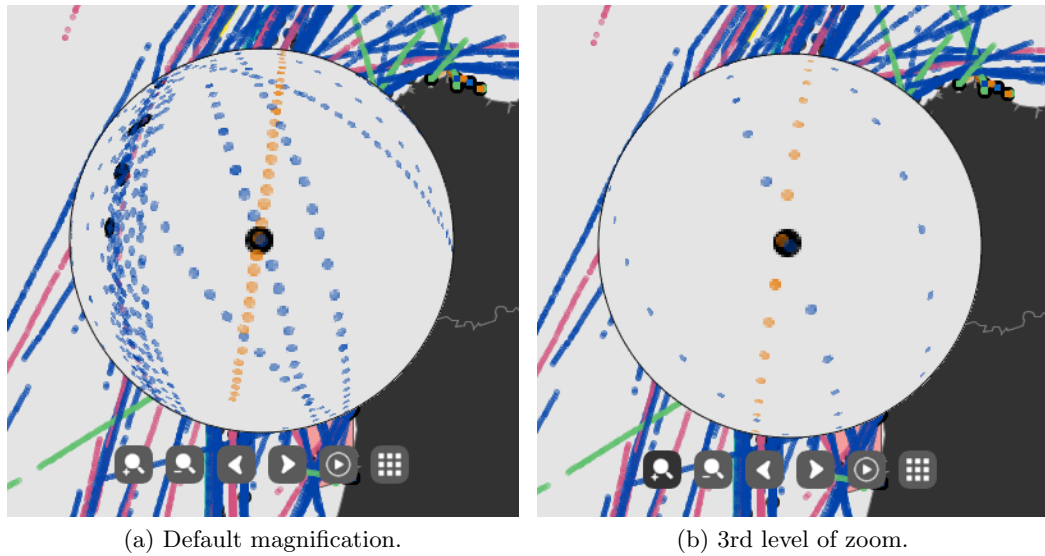


Figure 5.4: Example of the fish-eye lens with different levels of magnification.

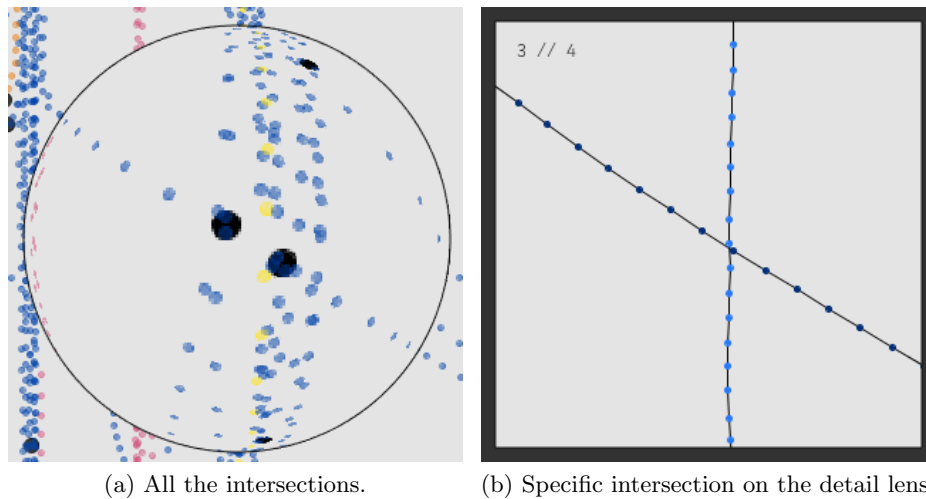


Figure 5.5: Example of the detail lens usage.

The intersections can be changed using two controls displayed next to the zoom ones. The analysis made to each one can be a static observation of the trajectories involved on the intersection (which are presented through the detail lens), an overtime analysis (described on the following subsection) or an animation of the trajectories (as described on section 4.5). This last one may offer a problem when the two vessels are from the same type: the collision of the colors can create difficulties on identifying the positions of each vessel. To fix this issue, a sub-palette of colors was created from the one presented on section 4.2. For each type of vessel two new colors were created based on the original one, a lighter and darker one. In the scenarios where two vessels of the same type are displayed on the animation, these two colors are used instead of the original one. This approach is also used for the detail lens of the intersections, with the addition of the positions also being connected with a black line, as the image (b) of Figure 5.5 shows. The sub-palette created is presented on Figure 5.6. Each original color is on the middle row while the lighter and

darker colors are displayed on the top and bottom rows, respectively.

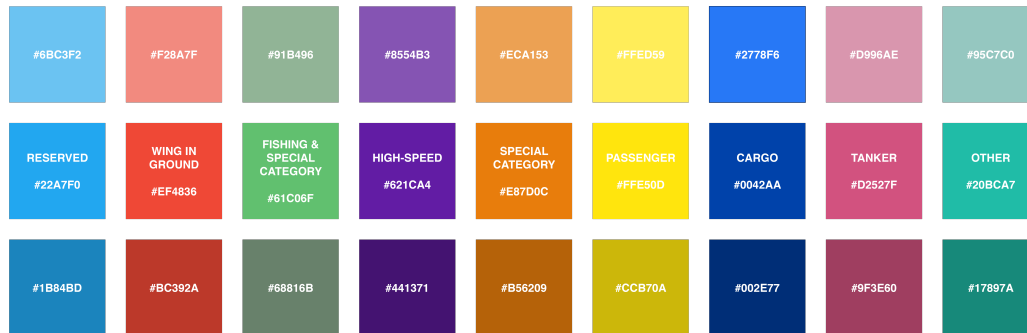


Figure 5.6: Color sub-palette for the representation of two vessels of the same type.

In order to complement the search mechanism, two additional filters were created specifically for this context. One allows the isolation of the intersections by two main areas using the following options:

- Near Coast, which only displays intersections close to the coast line (within 25 kilometers of it);
- High Sea, which only displays intersections far from the coast line (at least 25 kilometers from it).

For this filter to work each position was classified as being near or far from the coast. This classification process was made offline to avoid performance issues. A Python script compared each position with the boundaries of the countries and if the position was within 25 kilometers of any coast point it was considered near the coast. The other filter is a high frequency threshold of intersections between vessels. It defines a minimum number of intersections that two vessels had to make in the past for their intersections to be displayed. With this filter it is possible to isolate and detect suspicious behaviors related with vessels that intersect frequently. Similarly to the previous filter, the number of intersections between each pair of existing vessels was calculated offline to avoid performance issues with the filter. A Python script was used to create a hash map structure with all the combinations of two vessels and it executed a counting process by iterating over the intersections.

5.1.3 High Density Areas with an Abnormality Level

The visual search mechanism already described is important to identify and isolate intersections for individual analysis, but when a big quantity of intersections exist within the visible data it may be difficult to decide where to start the search. With this issue in mind, an approach to identify areas of particular interest was developed. The approach can be described in the following steps, also presented on Figure 5.7:

1. Identify the areas, for each day, where the quantity of intersections is higher;
2. Understand if those areas are constant over the days;
3. Define more common areas over the days as less probable of having abnormal behaviors and less common areas more probable;
4. Visualize the areas and identify the abnormality levels.

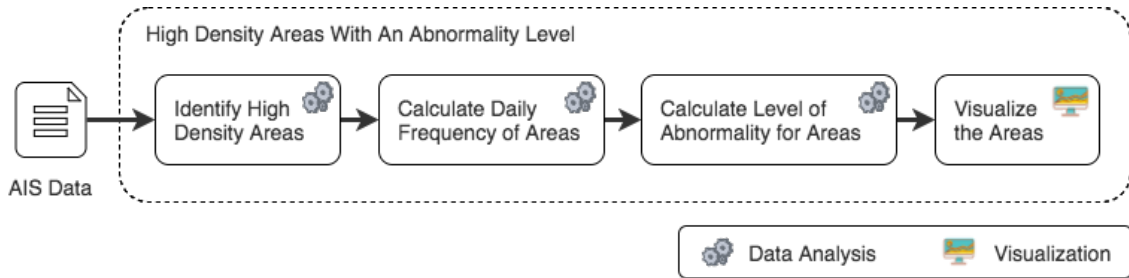


Figure 5.7: Proposed steps to identify areas of intersections with particular interest.

Regarding the first step, areas with an higher quantity of intersections can be seen as clusters with an higher density. Therefore, a density-based clustering strategy was used to extract these areas from the data. Density Based Spatial Clustering of Applications with Noise (DBSCAN) has been used extensively with AIS data, for different proposes, and had shown good results, which made it the most obvious choice for the algorithm to be used. However, this algorithm requires the minimum number of points by clusters ($MinPts$) and the distance between each point and its neighbors (ϵ), and this last one is not easy to estimate because there are zones of the map where the distance between the vessels positions is supposed to be lower (i.e. near the ports) and zones where it is supposed to be higher (i.e. high sea corridors). For this reason the Hierarchical Density Based Spatial Clustering of Applications with Noise (HDBSCAN) was also considered, because it uses an approach where the ϵ value is not required as a parameter and the clusters can have different densities. Both algorithms were experimented with different configurations of the parameters to allow a more effective choice of clusters. Regarding the values of $MinPts$, it was observed by visual analysis that, without the exception of ports, there were no considerable areas with more than 100 intersections. Therefore, this value was used as a maximum and the $MinPts$ was experimented with four values: 25, 50, 75 and 100. Regarding the values of ϵ for the DBSCAN, a maximum of 750 meters was defined and the parameter was experimented with three values: 250, 500 and 750 meters. Notice that the algorithms were applied individually for each chunk of AIS data because each one has the positions for a single day. To choose the better algorithm and configuration the silhouette coefficient was applied to the retrieved clusters, using an euclidean distance for the calculations. The coefficient results are presented on Tables A.1 to A.4 of Appendix A.1, each one containing the results by chunk for one of the four $MinPts$ values (25, 50, 75 and 100, respectively). The number of clusters extracted by each of the configurations are presented on Tables A.5 to A.8 of Appendix A.2, being the average values presented

on Table A.9 from the same Appendix. For the final decision the average and the standard deviation of all the chunks for the resulting coefficients was considered. Table 5.1 shows that the HDBSCAN with $MinPts = 75$ was the configuration that presented an higher silhouette coefficient average for all the chunks with a low standard deviation. Therefore, this was the chosen configuration for the clusters extraction.

Table 5.1: Silhouette coefficients averages for density-based clustering algorithms.

Algorithm	Average	Std. Deviation
HDBSCAN $MinPts = 25$	0.304	0.142
HDBSCAN $MinPts = 50$	0.458	0.092
HDBSCAN $MinPts = 75$	0.525	0.084
HDBSCAN $MinPts = 100$	0.507	0.081
DBSCAN $MinPts = 25$ and $\varepsilon = 250$	0.291	0.079
DBSCAN $MinPts = 25$ and $\varepsilon = 500$	0.431	0.124
DBSCAN $MinPts = 25$ and $\varepsilon = 750$	0.499	0.070
DBSCAN $MinPts = 50$ and $\varepsilon = 250$	0.230	0.101
DBSCAN $MinPts = 50$ and $\varepsilon = 500$	0.435	0.066
DBSCAN $MinPts = 50$ and $\varepsilon = 750$	0.431	0.116
DBSCAN $MinPts = 75$ and $\varepsilon = 250$	0.155	0.122
DBSCAN $MinPts = 75$ and $\varepsilon = 500$	0.407	0.072
DBSCAN $MinPts = 75$ and $\varepsilon = 750$	0.473	0.077
DBSCAN $MinPts = 100$ and $\varepsilon = 250$	0.093	0.095
DBSCAN $MinPts = 100$ and $\varepsilon = 500$	0.384	0.076
DBSCAN $MinPts = 100$ and $\varepsilon = 750$	0.442	0.064

Regarding the second step, the key idea was to associate a frequency to each cluster of each day. To calculate this frequency (F_c) the formula on Equation 5.12 was proposed.

$$F_c = \frac{\text{Number of days where the cluster (partially) exists}}{\text{Total number of days}} \quad (5.12)$$

Considering the necessary parameters for the formula above, the total number of days with AIS data is a known value, but the number of days where each cluster exists needs to be calculated. For this propose the Algorithm 3 was developed. This algorithm receives a cluster from a day, the clusters of the remaining days (the day from the passed cluster is not included) and an overlap threshold (between 0 and 1). The key idea is to evaluate if the cluster appears on other days by calculating the area of intersection between it and each of the remaining clusters. This area is then converted to a percentage by dividing it with the area of the cluster in analysis, and if this percentage is greater than the threshold (given as a parameter) it is considered that the cluster appears on the day of the other one. The overlap threshold is important in this analysis because it is highly unlikely that

two clusters have the exact same shape, which would be necessary for a degree of 100% of overlap. Moreover, the goal is to identify if an area that has an high density in one day also has in other days, and for that reason a total overlap is not required as the same area may be within a cluster with a bigger or smaller area on others days, depending of its cohesion. For these reasons, the value of this threshold was defined as 50%.

Algorithm 3 Calculate the number of days where a cluster exists

```

1: function CALCULATENUMBERDAYSCLUSTEREXISTS(cluster, clustersOtherDays,
   overlapThreshold)
2:   numberOfDays  $\leftarrow$  0
3:   baseClusterArea  $\leftarrow$  area of the cluster
4:   for each day of clustersOtherDays do
5:     for each otherCluster of the day do
6:       intersectionClusters  $\leftarrow$  cluster  $\cap$  otherCluster
7:       overlapArea  $\leftarrow$  area of the intersectionClusters
8:       overlapPercentage  $\leftarrow$  overlapArea/baseClusterArea
9:       if overlapPercentage  $\geq$  overlapThreshold then
10:        numberOfDays  $\leftarrow$  numberOfDays + 1
11:        break the current for loop
12:       end if
13:     end for
14:   end for
15: end function

```

Notice that the Algorithm 3 could use the overlap of each individual point from the clusters for the overlap calculations (which was actually the first approach), but the required time for the algorithm to compute was unfeasible. Therefore, the area from the clusters was considered for the overlap calculations, but calculate this area is not an immediate operation. The first approach was to consider the cluster as rectangle, calculate the maximum and minimum values of its points regarding both the x and y axis, and finally use the formula for the rectangle area. However, the results were very inflated (frequencies achieved values of 100% often) mainly because the error of the area was too high. For example, if a cluster contains the majority of its points near its centroid but a single one is far from it, the area would consider that point as the maximum or minimum (depending on the plane quadrant) and the resulting value would be inflated wrongly. An example illustrating this problem is presented on Figure 5.8.

The second approach was to consider a circle shape for the cluster, and consisted on calculating its centroid and finding the point that had the biggest Haversine distance from it. This distance was used as the radius of the circle, and the formula for the circle area was finally used. This approach revealed the same problems as the first one and was discarded, as the example on Figure 5.9 shows.

The third and final approach was to extract the convex hull¹ of the clusters points using the Graham's Scan algorithm (Graham, 1972). The shape of the convex hull adapts much

¹Given a set of points X , the convex hull is the minimum set of these points that create a convex polygon containing all entries of X .

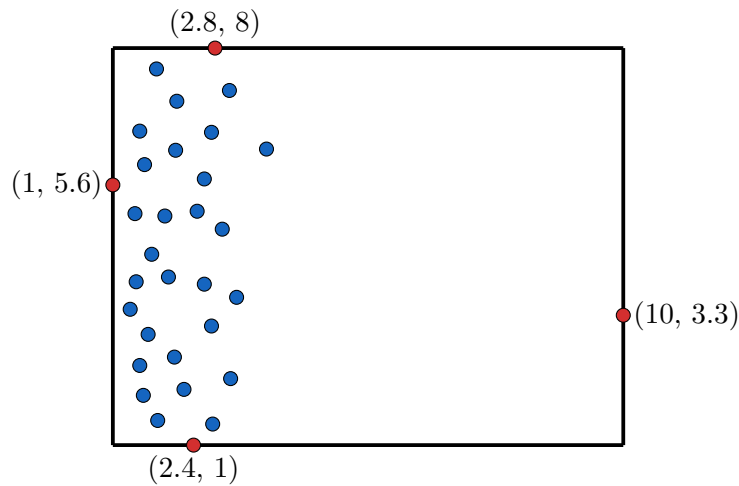


Figure 5.8: Example of the problem, raised by the usage of a rectangle to represent the cluster, when calculating its area.

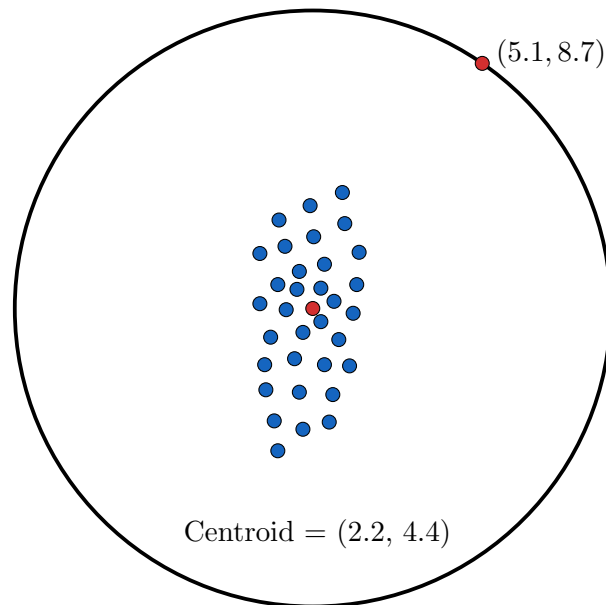


Figure 5.9: Example of the problem, raised by the usage of a circle to represent the cluster, when calculating its area.

better to the cluster because it considers all the boundaries and is able to represent them without restrictions (being an irregular polygon, it can assume any convex shape). An example of a convex hull is presented on Figure 5.10.

With the cluster represented as a convex polygon, and considering that the coordinates from the vertices are known, the formula presented on Equation 5.13 can be used to calculate its area.

$$A_{cp} = \frac{1}{2} \sum_{k=0}^{n-1} \left(x_k * y_{k+1} - y_k * x_{k+1} \right) \quad (5.13)$$

For example, the area of the polygon presented on Figure 5.10 is 30.78 (see Equation 5.14).

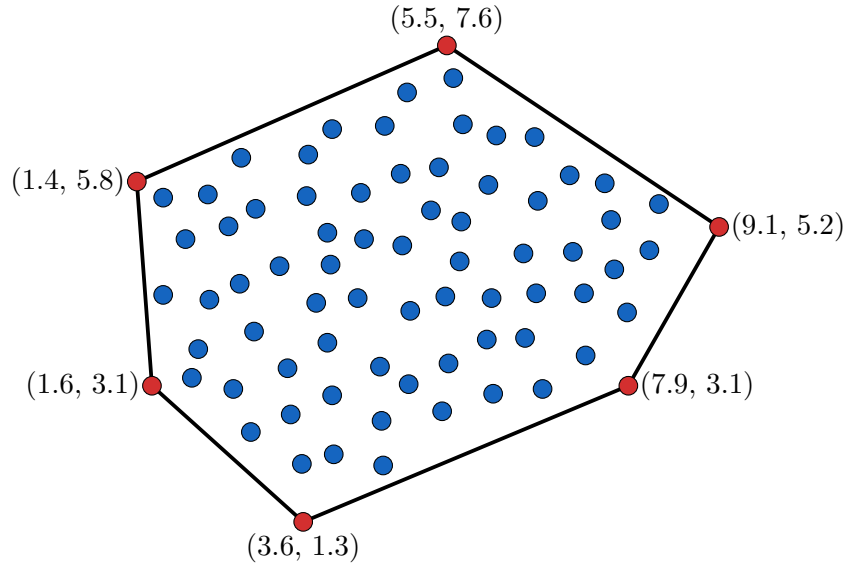


Figure 5.10: Example of a convex hull.

$$\begin{aligned}
 A_{cp} &= \frac{1}{2} * \left((5.5 * 5.8 + 1.4 * 3.1 + 1.6 * 1.3 + 3.6 * 3.1 + 7.9 * 5.2 + 9.1 * 7.6) - \right. \\
 &\quad \left. (7.6 * 1.4 + 5.8 * 1.6 + 3.1 * 3.6 + 1.3 * 7.9 + 3.1 * 9.1 + 5.2 * 5.5) \right) \\
 \Leftrightarrow A_{cp} &= \frac{1}{2} * (159.72 - 98.16) \\
 \Leftrightarrow A_{cp} &= \frac{61.56}{2} \\
 \Leftrightarrow A_{cp} &= 30.78
 \end{aligned} \tag{5.14}$$

This approach had a cost on the performance (mainly because of the extraction of the convex hulls), but the results were more accurate as the clusters were better represented by the convex polygons.

With the frequency of each cluster calculated, the third step was addressed by associating less common clusters as more likely to include abnormal behaviors. This concept has been used in other studies and relies on the fact that an area where a higher density of intersections is often found should be considered less probable of being abnormal when compared to one where this higher density happens as an exception. Therefore, the level of abnormality of each cluster was calculated based on the frequency using the formula on Equation 5.15. This level is still normalized between 0 and 1, and can also be seen as a percentage.

$$AL_c = 1 - F_c \tag{5.15}$$

Finally, addressing the fourth step, to visualize these clusters the convex hulls are drawn on the platform and the visual variable color was used for the representation of the abnor-

mality levels. These levels were discretized into four intervals, namely $[0, 0.25[$, $[0.25, 0.50[$, $[0.50, 0.75[$ and $[0.75, 1]$. A gradient of the color red was created with four levels, each one for a specific interval, and being the 1st level the lower and the 4th level the higher. The strongest red is for the last interval and the lightest red is for the first interval. The gradient is presented on Figure 5.11. The convex hull drawn for each cluster is filled with the color that matches its interval of abnormality.



Figure 5.11: Color palette for the representation of the abnormality levels.

Figure 5.12 shows an example with two of these areas drawn on the platform. The one from the left has a lower level of abnormality when compared to the one on the right.

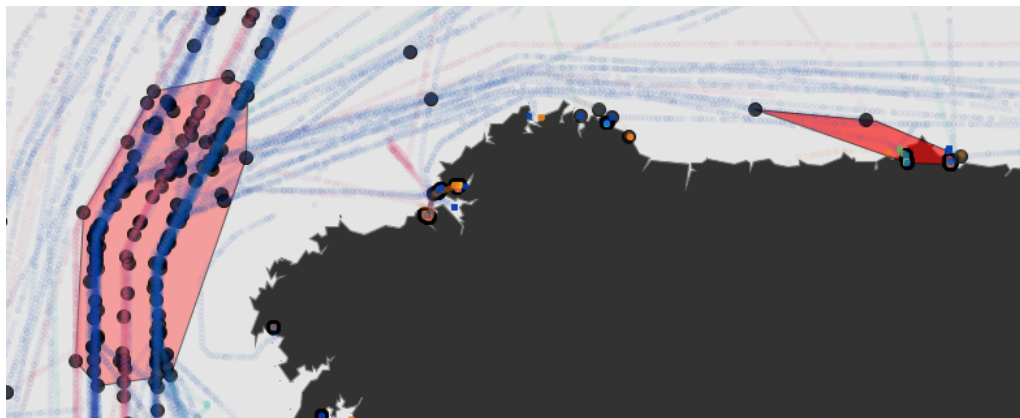


Figure 5.12: An example with two abnormal areas drawn on the platform.

5.1.4 Overtime Analysis with Small Multiples

When analyzing an intersection with the propose of detecting abnormal behavior, an important aspect is the frequency that it occurs. For a domain expert, the frequency or the existence of recurrent patterns may be useful for confirming or discarding suspicious behaviors. In order to compare the trajectories in different periods a small multiples strategy was used because it allows the comparison of multiples views by varying a specific attribute of the data. The vessels involved in a specific intersection are selected and their trajectories are displayed through small multiples with different granularities. The first scale uses a monthly granularity and displays a grid of 3 by 4 cells where each one displays the trajectories of the vessels in a month (the year in analysis is the same of the

intersection). Months where no data exists for these vessels are filled in with grey. Figure 5.13 shows an example of this scale for a set of selected trajectories within February and March of 2012. Then, by clicking on a specific month, a drill-down on the granularity

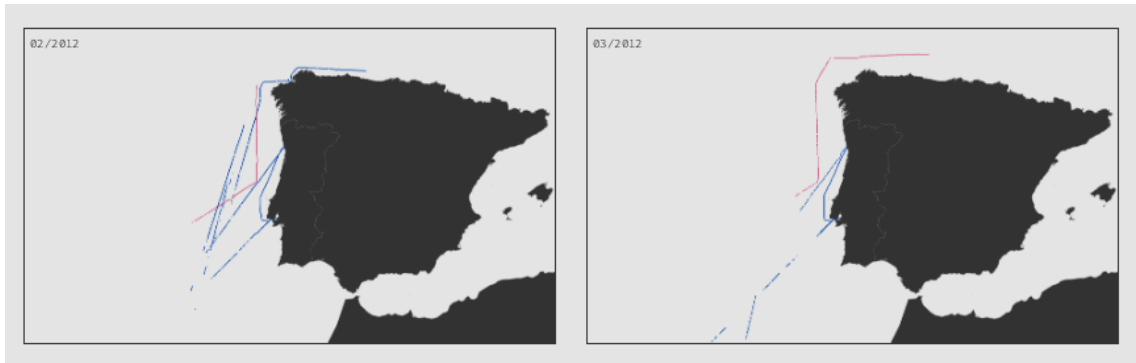


Figure 5.13: An example of the small multiples approach with the monthly granularity.

will occur and a second scale will be displayed where all the days of that month are presented with the respective trajectories in them, or filled with grey if there are no reported positions. This scale displays a grid with 7 columns, each one corresponding to a day of the week (starting on Sunday), which allows an easier pattern correlation (for instance, intersections that happen on the same day of the week or only on weekends). The number of lines depends of the selected month. Figure 5.14 shows an example for this granularity for a set of selected trajectories between 21 and 23 of February of 2012.

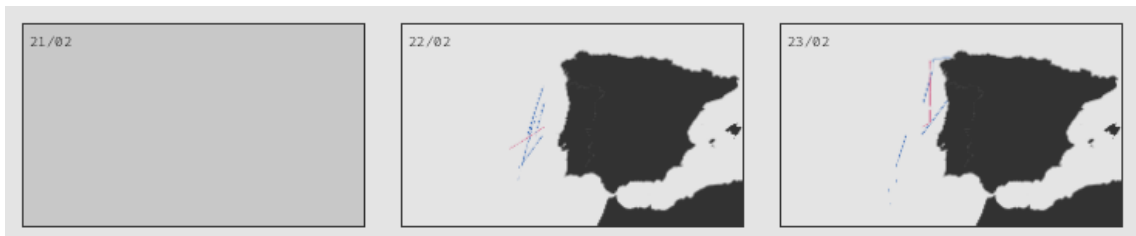


Figure 5.14: An example of the small multiples approach with the daily granularity.

Finally, if a specific day with data is clicked then the animation of the involved trajectories starts automatically playing. This small multiples strategy is available for individual intersections through a button next to the fish-eye lens, but can also be applied to a set of selected trajectories (similarly to the animation).

5.2 Speed Outlier Behaviors

The different types of vessels sail on the sea with an average speed that, in normal scenarios, should not have big variations. This means that vessels sailing too slow or too fast can be

classified as having a suspicious (and eventually abnormal) behavior. Besides, several of the identified illegal activities are commonly associated with a reduction of the involved vessels speed (i.e., fishing in illegal areas). Therefore, an approach to detected vessels with speed values that can be considered outliers was developed.

An initial study of the statistical distribution followed by the speed variable of each type of vessel was conducted. For this study only valid speed values (below or equal to 100 knots) and values of moving vessels (greater or equal to 0.5 knots) were considered. The goal was to understand if these variables followed a normal distribution. If this condition was verified the detection of outliers could be achieved by detecting which values had a probability density situated on the extremes ends of the Gaussian curve. However, as Figure 5.15 shows, the variable does not follow a normal distribution for any of the types of vessels. The ones that are closer to follow this distribution are the cargo and tanker vessels (images (h) and (i)), but an initial peak near the 0 value on the x axis makes it not normal. Therefore, this approach was discarded.

Based on the previous results, a non-parametric approach was followed using a Gaussian Kernel Density Estimation (KDE), which is a strategy to estimate the Probability Density Function (PDF) of a variable using its known data and without assuming a predefined distribution (Parzen, 1962). The estimated PDF for each vessel type is also presented on Figure 5.15. The results show that the functions fit the data distributions, and this approach was the adopted one. Therefore, the strategy defined to classify a speed value of a vessel as too slow, normal or too fast was the following:

1. Through a Python script, estimate the PDF of the speed variable for each vessel type using a Gaussian KDE;
2. Discretize each PDF into 0.5 intervals regarding the x axis (the speed itself), and save the associated y values for each of these intervals;
3. Define a PDF threshold to each vessel type where values below it would be considered as outliers;
4. On the platform, a filter was created to display only the positions of the vessels that have a low or high speed, or that are stopped (this last one does not use the described strategy, it considers any position with a speed below 0.5 knots as the vessel being stopped). For the low and high speed filter the steps are:
 - (a) Round the speed value to its closest 0.5 interval (i.e., a value of 10.3 would be rounded to 10.5 and a value of 9.1 to 9);
 - (b) Obtain the PDF result for the rounded speed value from the previously discretized list of the corresponding vessel type;
 - (c) Evaluate if the PDF value is below the threshold for that vessel type. If it is below, classify the position as low or high by verifying if its speed is less or greater than the average value for that vessel type, respectively.

Within the described strategy a particularly difficult task is to find the ideal threshold for the PDF of each vessel type. The ideal threshold would have to be sufficiently high to include all PDF values that are small but without including the peaks of the function where the normal values reside. Considering that the functions are never monotonic and include several times the two described situations, one can see the smaller PDF values as local minima and the peaks as local maxima. Therefore, the formula on Equation 5.16 was defined to calculate the threshold of each PDF.

$$T_{pdf} = \overline{Local\ Minima \cup Local\ Maxima} \tag{5.16}$$

The calculated thresholds for each PDF function are also presented on Figure 5.15.

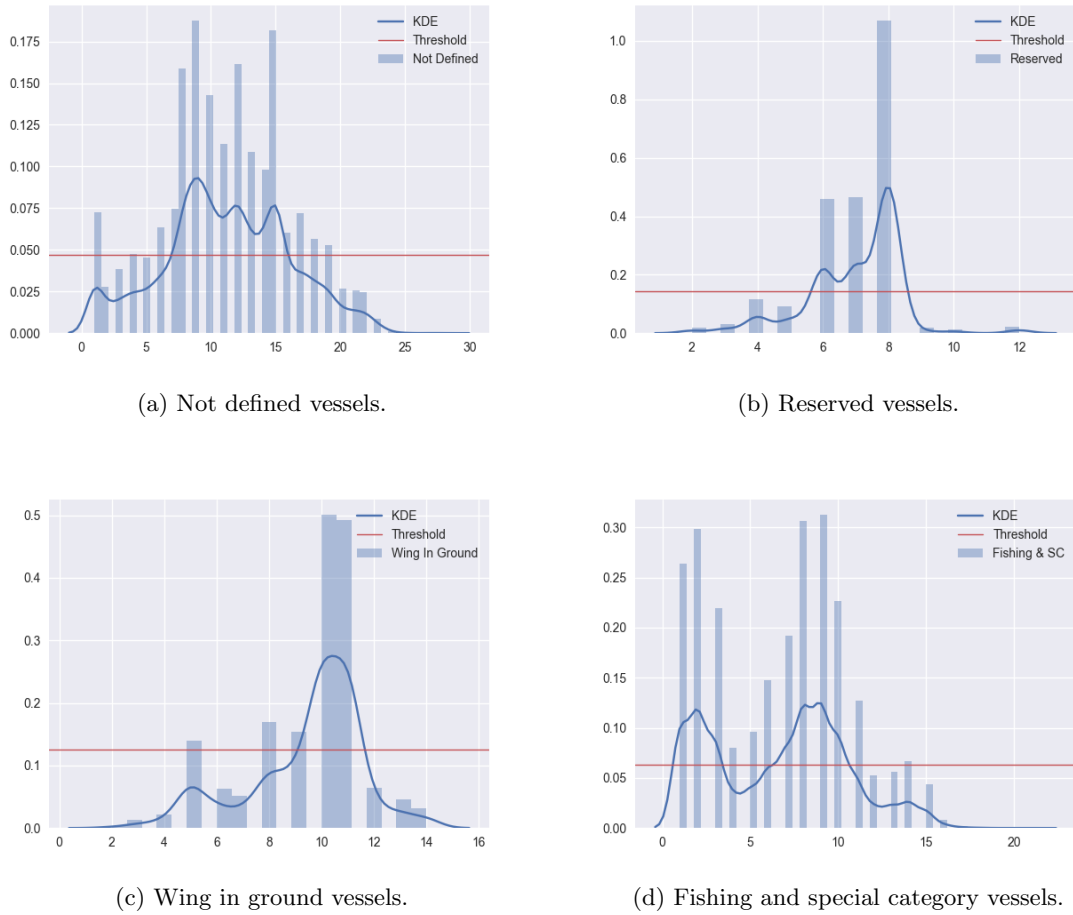
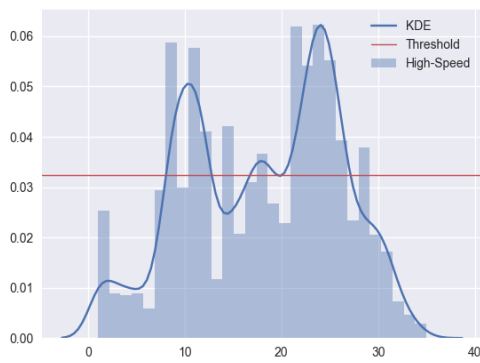
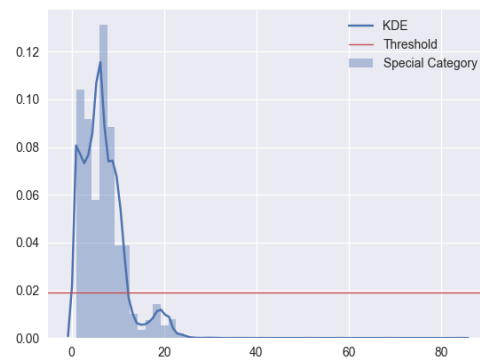


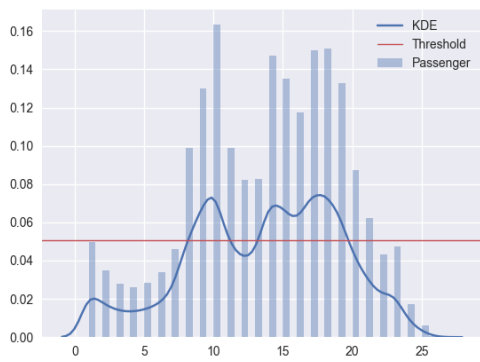
Figure 5.15: Distribution of the speed variable from each type of vessel.



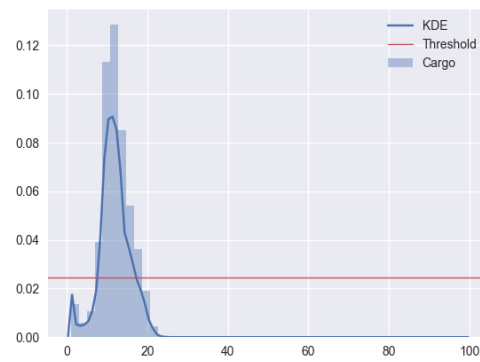
(e) High-speed vessels.



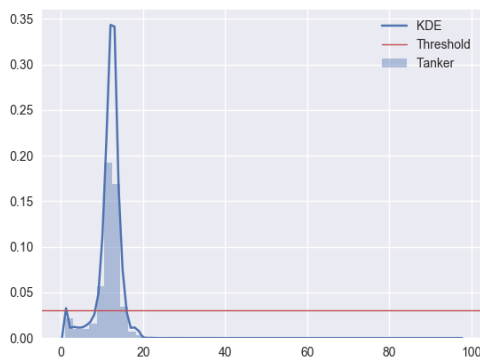
(f) Special category vessels.



(g) Passenger vessels.



(h) Cargo vessels.



(i) Tanker vessels.



(j) Other types of vessels.

Figure 5.15: Distribution of the speed variable from each type of vessel. (cont.)

With the general problem of detecting speed outliers solved, there was a particular scenario that was not considered in the described approach. For the special case of the fishing vessels, to detect the particular abnormal behavior of illegal fishing not only the speed of the vessel must be considered but also its location. As the image (d) from Figure 5.15, is actually normal to see fishing vessels sailing in a low speed when they are fishing. So, to detect this abnormal behavior, a further constraint was necessary to take in consideration:

the area where the vessel is sailing in a low speed. Therefore, a new filter was added to the platform for this propose. The illegal areas are defined through concave or convex polygons that are loaded to the system, each one being identified by its vertices. When the filter is active, only the positions of the fishing vessels that are inside these polygons are displayed. To verify if a point is inside of one of the polygons an implementation of the ray casting algorithm (Haines, 1994) was used. Figure 5.16 shows an example of fishing vessels sailing inside an illegal fishing area (this area is fictional).

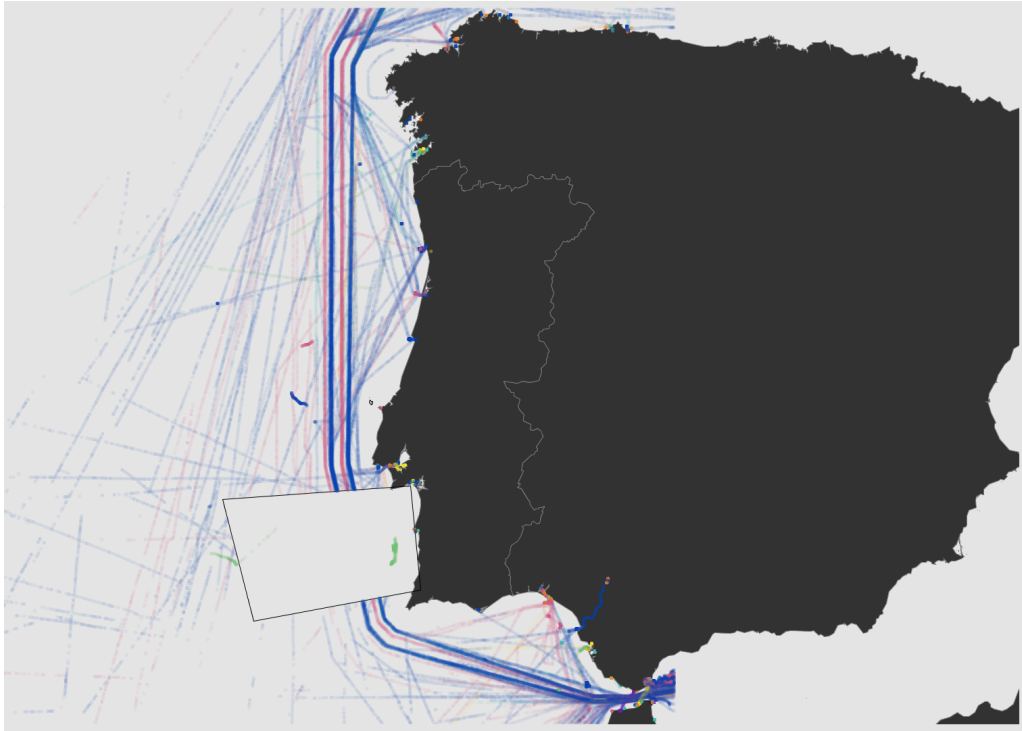


Figure 5.16: An example of fishing vessels sailing inside an illegal fishing area.

Regarding the generalization of these approaches, the strategy to detect speed outliers can be applied to any dataset of AIS data, from any maritime area, just by recalculating the PDF for each vessel type and the respective thresholds. The strategy to detect vessels fishing in illegal areas can also be generalized, with the restriction that the areas must be define through polygons. This means that, for example, if the area is defined by a non-polygon (i.e., a circle) then its representation must be the polygon which has that non-polygon inscribed.

5.3 3D Explorations for Time Evolution

The two-dimensional visualization of the AIS data has natural restrictions when considering the amount of information that can be displayed simultaneously. Only when the motion variable is introduced more information can be displayed on a 2D plane. In geographical systems, the two-dimensional point (x, y) is commonly use to represent the position of an

object and, as described on section 4.2, the developed platform follows the same approach. However, in a three-dimensional plane each point is represented by three values (x, y, z) , which gives the possibility of displaying more information simultaneously. Therefore, some attempts of exploring the 3D plane were developed, with particular emphasis on using the new available variable to introduce the time information of the positions. The key ideas were to understand the evolution of specific vessels over the time and also to understand patterns related with this variable.

Four 3D projections were developed individually with the following proposes:

- Use (x, y, z) to display the AIS positions with Earth's curvature through an ECEF projection (Leick et al., 2015);
- Use (x, y) for the normal 2D projection and the z axis to represent the date of the positions;
- Similar to the previous one, but using the z axis to represent the timestamp of the positions. The intersections between trajectories are simultaneously displayed;
- Use (x, y) for representing the timestamp of the positions through a 24h clock format and the z axis to represent each different vessel.

Notice that to use these 3D projections the trajectories from the vessels in analysis need to be previously selected. Besides, all the data from these vessels (without date and time restrictions) is considered and displayed. Finally, the 3D axes are always displayed on the origin with the x axis painted red, the y axis green and the z axis blue.

Regarding the first approach, it was a direct usage of the ECEF projection, which can be seen as an equivalent of the Spherical Mercator projection for the 3D plane, that is able to represent the Earth's curvature. Although no time information is included in this projection, it was used for testing the 3D environment components that are presented later in this section. Besides the longitude and latitude of each position, this projection also requires as a parameter the radius of the sphere that is used for mapping the Earth's curvature. In order to keep the proportions as real as possible, the Earth radius (approximately 6371 kilometers) was used. Figure 5.17 shows an example of this approach, applied to 4 different vessels, in two different perspectives. All the trajectories of these vessels are displayed simultaneously and it seems like they all overlap. However, the trajectories of each vessel occurred in different days, and this projection is agnostic to the time variable. The images also show that the vessels have similar routes over the time and that the cyan vessel has several gaps in its reported positions.

The second approach uses the Spherical Mercator projection for displaying the positions through the x and y variables, as if the plane was 2D. However, the z axis is used to display the date of each position by applying an offset of 30 units to each date. This means that, for example, a position of the first day of data will have $z = 0 * 30 = 0$ and a position of the third day of data will have $z = 2 * 30 = 60$. This offset between dates allows the

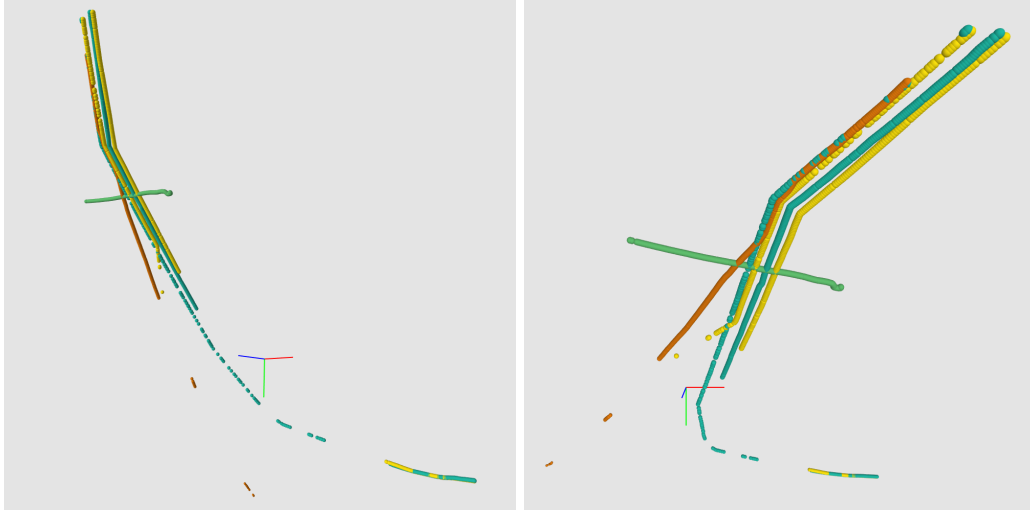


Figure 5.17: ECEF projection of 4 different vessels. Each image corresponds to the same data with different perspectives.

perception of which vessels sail more and less often. To calculate the z value in a generic way only the date part of the UNIX epoch timestamp must be considered (the time part is cleared with zeros), and, assuming $Date_{first}$ as the date of the first position, the formula from Equation 5.17 is used. The formula assumes the positions were previously sorted by date.

$$z = \left(\frac{Date_{current} - Date_{first}}{86400} \right) * 30 \quad (5.17)$$

Figure 5.18 shows an example of this approach, applied to 4 different vessels, in two different perspectives. The images show that only from specific perspectives the difference between the dates of the trajectories are noticeable. However, the time variable is already considered by this projection in a discretized way and, for the perspectives where the different dates are visible, the trajectories no longer overlap.

The third approach is very similar to the second one but the z axis is used to display the complete timestamp of each position. A range of 20 units is used to represent each hour and the necessary offsets are calculated to achieve the correct z value for each timestamp. This approach does not allow an explicit distinction between the dates of positions but creates a continuity effect that allows a more intuitive analysis of the evolution of the vessels over the time. To calculate the z value in a generic way the UNIX epoch timestamp is divided in two parts, one that contains only the date and other that contains only the time, and, assuming $Date_{first}$ as the date of the first position, the formula from Equation 5.18 is used.

$$z = \left(\left(\frac{Date_{current} - Date_{first}}{86400} \right) * 480 + \left(\frac{Time_{current}}{3600} \right) * 20 \right) \quad (5.18)$$

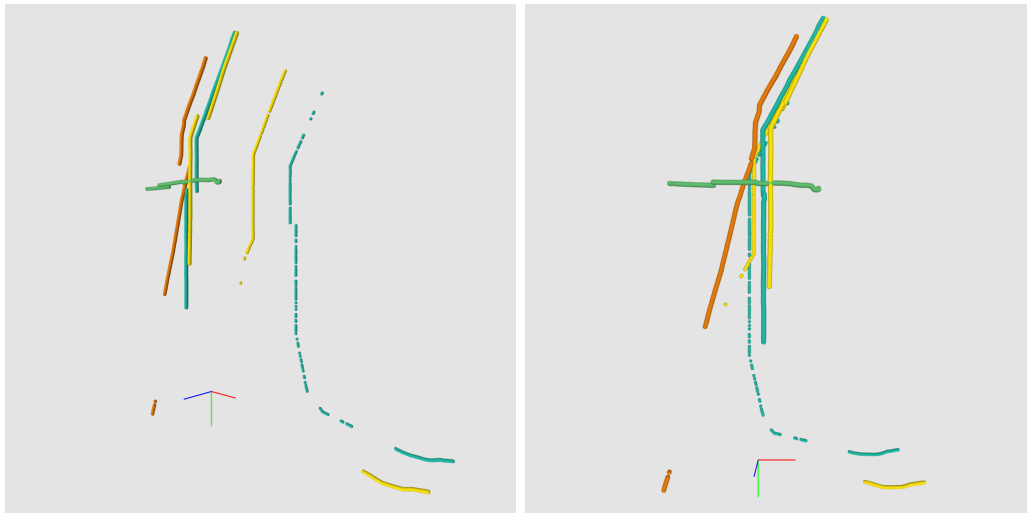


Figure 5.18: Projection using z axis to represent the date, applied to 4 different vessels. Each image corresponds to the same data with different perspectives.

In this approach another important aspect is that the intersections between positions are displayed through black spheres. These intersections only consider the spatial component of the positions and are calculated in real time by comparing the positions between each other. An intersection occurs when a point has a linear distance smaller or equal to 2, in all the axes, when compared to another point from a different vessel. Figure 5.19 shows an example of this approach, applied to 4 different vessels, in two different perspectives. The images show that this projection displays the time variable in a continuous way, and that is the reason why it can also be used to detect intersections. However, from specific perspectives some false intersections may appear, and that is why the real ones are represented through a black sphere. In this case, only two of the vessels trajectories have an intersection and it is visible on both perspectives.

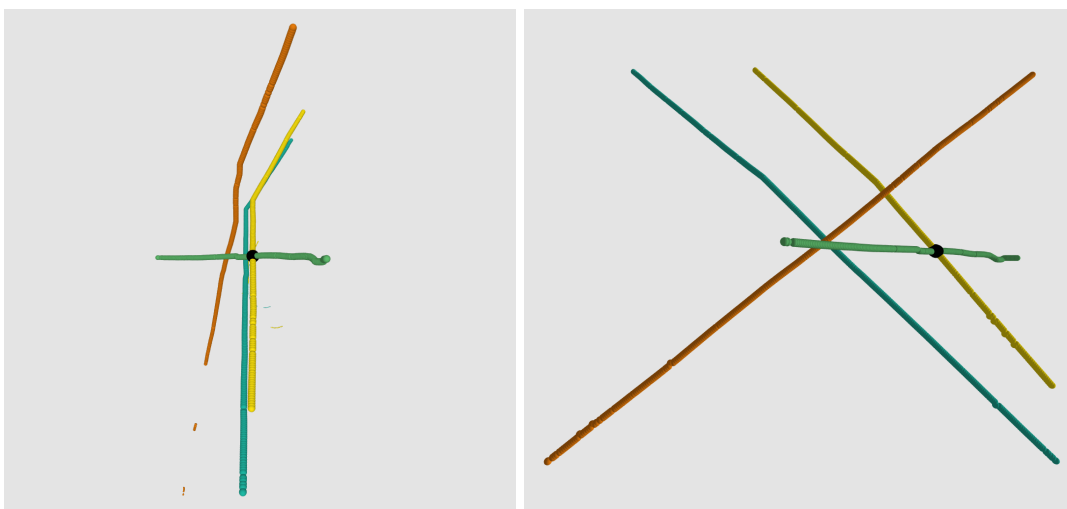


Figure 5.19: Projection using z axis to represent the time, applied to 4 different vessels. Each image corresponds to the same data with different perspectives.

The fourth and last approach is very different from the previous ones and is totally focused on a time analysis. The key idea is to represent the positions in a 24 hour clock two understand the periods of the days when the vessels sail. The dates of the positions are represented by the radius of the clock, meaning that when the date increases the radius also increases, with an offset of 50 units by day. With the x and y variables used for the described propose, the z axis is used to represent the different vessels. Each vessel is associated with an integer index between 0 and n (number of vessels), and for each one an offset of 30 units is applied, which creates a clear separation between them. To calculate the x and y values the radius and the angle of the clock position are first calculated using the formulas on Equations 5.19 and 5.20, respectively.

$$radius = \left(\frac{Date_{current} - Date_{first}}{86400} \right) * 50 \quad (5.19)$$

$$angle = toRadians \left(\left(\frac{Time_{current}}{3600} \right) * \left(\frac{360}{24} \right) \right) \quad (5.20)$$

The values of x and y are then calculated using the formulas to convert polar coordinates to cartesian coordinates, as Equations 5.21 and 5.22 show. The z value is calculated by applying the offset to the vessel based on its index, as Equation 5.23 shows.

$$x = radius * \cos(angle) \quad (5.21)$$

$$y = radius * \sin(angle) \quad (5.22)$$

$$z = Vessel_{index} * 30 \quad (5.23)$$

Figure 5.20 shows an example of this approach, applied to 4 different vessels, in two different perspectives. The images show that this projection clearly identifies the different trajectories by situating them simultaneously on the specific date and time when they occurred. However, the identification of the different vessels is only possible from some perspectives and the geographical location of the positions is not represented.

Regarding all the approaches, the following environment configurations were also implemented:

- A set of mouse based operations to control the camera like performing translations, rotations, pan and zoom, which allows the exploitation of the 3D visualizations from different perspectives;
- The visualizations are centered on the origin to allow an easier contextualization with the axes. This is achieved by calculating the middle value for each axis from

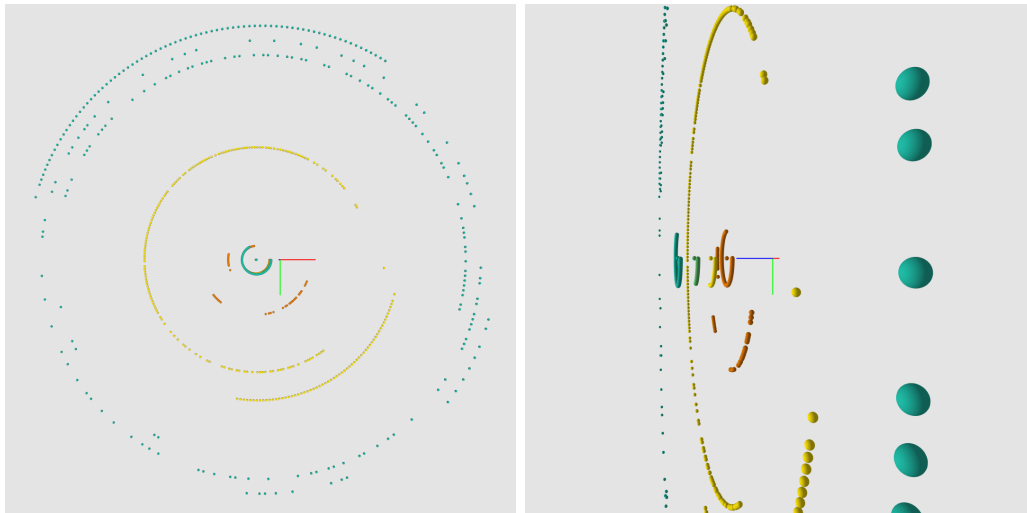


Figure 5.20: Projection representing the positions through a 24 hour clock shape. Each image corresponds to the same data with different perspectives.

all the points and offsetting each one by those individual values.

All the 3D strategies presented similar problems that require further analysis:

- Its very difficult to contextualize the points in their real positions on Earth. Although this aspect was not the target of the approaches, this contextualization is always important from the analysis perspective;
- The navigation on the 3D plane is difficult and, sometimes, unnatural. Besides, the usage of rotations to view the points from different perspectives aggravates the problem described on the previous point;
- Certain information is only visible from specific perspectives. For example, on the 4th projection the different vessels can only be seen through a specific rotation of the y axis.

Although 3D approaches can display more information through more variables, the problems described above allow the conclusion that the navigation and contextualization of such visualizations is much more difficult and can make them unfeasible.

Chapter 6

Case Studies

In order to validate the developed visualization approaches for the detection of anomalous behaviors, three case studies were conducted. These cases illustrate different scenarios where the detection would not be possible without a visual approach. The studies cover intersections scenarios, one with the speed outlier filter.

6.1 Hidden Intersection

There are specific areas of the sea where the vessels are supposed to sail, the so-called maritime corridors. The density of the traffic in these areas is much higher when comparing to others. Figure 6.1 displays several Automatic Identification System (AIS) trajectories from vessels that sailed through the main corridor of the Portuguese maritime area on February 22 of 2012. Apparently, the marked area contains only cargo vessels (the dark

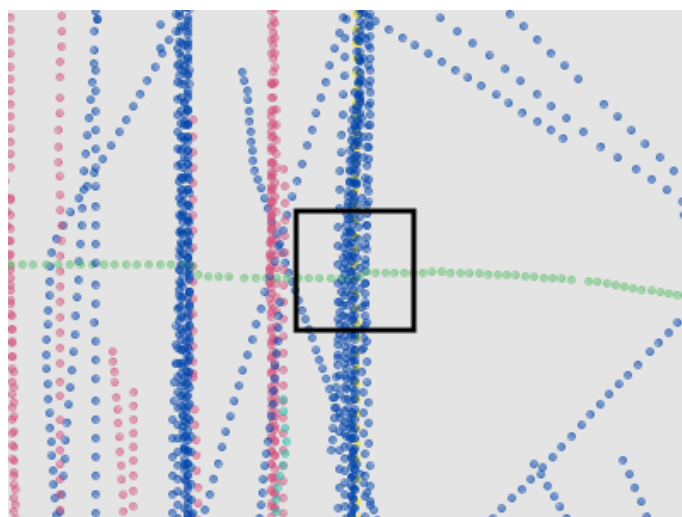


Figure 6.1: Visualization of several AIS trajectories.

blue ones) and a fishing vessel (the green one). When analyzing the fishing vessel, it appears that it may eventually intersect with several cargos. However, when the intersections

are activated and the fish-eye lens is applied on the black square area, with at least one level of zoom, an unexpected intersection is revealed. Figure 6.2 shows this intersection on the fish-eye lens (image (a)) and on the detail lens (image (b)).

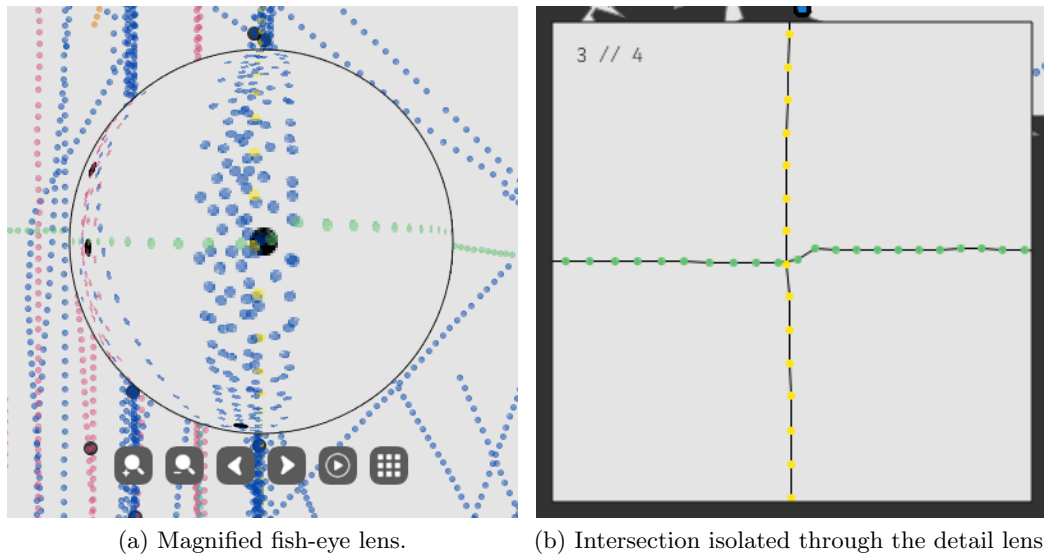


Figure 6.2: Identification and individual analysis of the intersection.

The intersection between the fishing and the passengers vessel (the yellow one) was hidden in the visual clutter created by the trajectories of the remaining vessels. Without the usage of the magnified fish-eye lens it would be very difficult to detect and analyze this intersection. After the isolation of the involved trajectories the animation was used for a more detailed analysis, as Figure 6.3 shows.

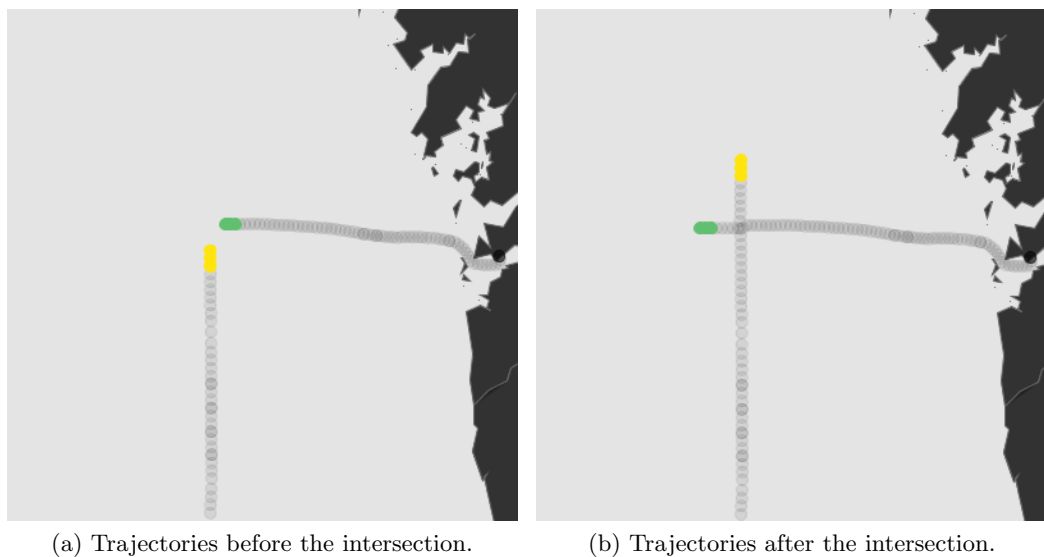


Figure 6.3: Animation of the trajectories from the intersection.

The displayed frames on Figure 6.3 show the current positions of the vessels with 100% opacity and the past positions with only 10%, creating the trace effect. Using the interpolation method already explained, the animation shows that the fishing vessel moves

slower, particularly in the beginning of the trajectory when it is leaving the coast.

When applying the small multiples strategy one can conclude that the vessels did not intersect again. Figure 6.4 shows that both vessels intersect only once on February.

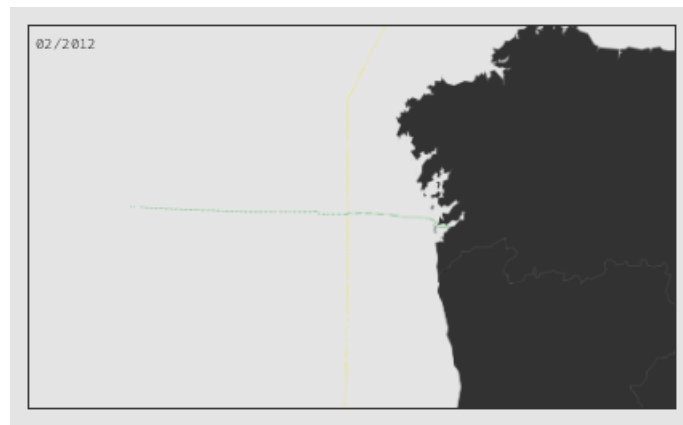


Figure 6.4: The small multiples approach for this intersection with the monthly granularity.

6.2 Fishing Vessels Intersections

Fishing vessels can be particularly hard to analyze considering that their trajectories are very irregular when comparing, for example, with cargos or tankers. Figure 6.5 displays AIS trajectories from March 10 of 2012. Notice that the vessel types filter was used to remove the cargos and tankers from the screen.



Figure 6.5: Visualization of AIS trajectories.

The area marked by the black square on the Figure 6.5 appears to contain a trajectory from a fishing vessel. However, when the intersections feature is enabled it shows that there are more than one trajectory from fishing vessels in that area and, more importantly, they intersect in several points. Moreover, the area is considered to have a high density of intersections and the red color indicates that the level of abnormality is the 3rd (out of 4). Figure 6.6 shows these intersections and the high density area.

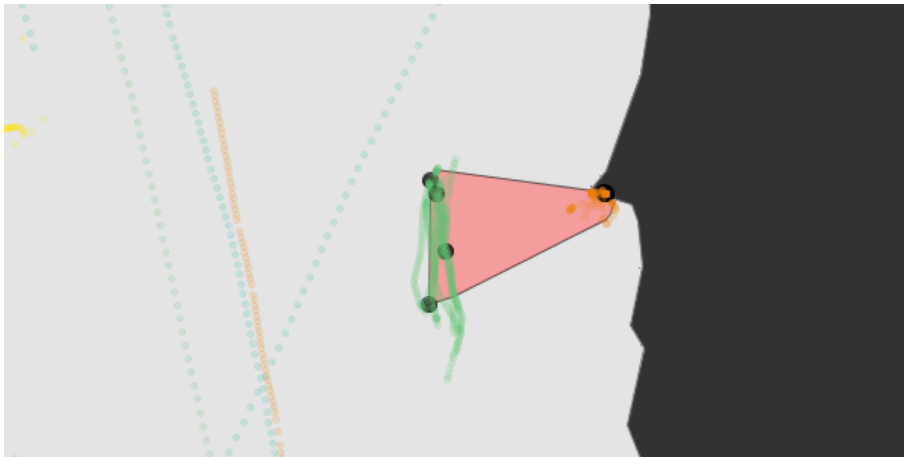
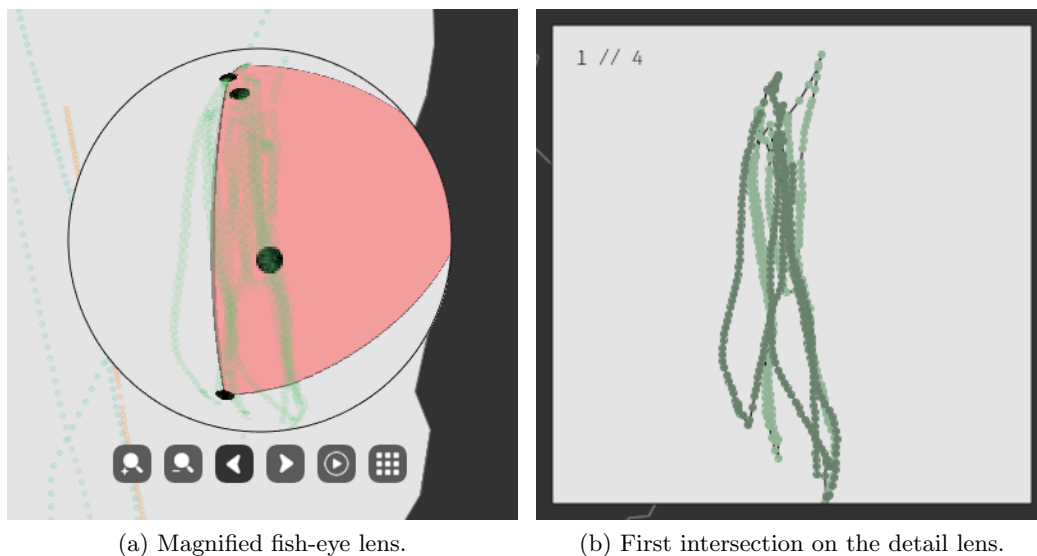


Figure 6.6: Intersections and high density area of the fishing trajectories.

When the fish-eye lens is applied on the area the intersections are isolated through the detail lens. Figure 6.7 shows all intersections on the fish-eye lens (image(a)) and only the first intersection on the detail lens (image (b)). Notice that, being the two vessels from the same type, they are represented by a lighter and darker green colors.



(a) Magnified fish-eye lens.

(b) First intersection on the detail lens.

Figure 6.7: Detection and isolation of the first intersection.

The animation of trajectories was applied to the intersection and, as Figure 6.8 shows, one of the vessels is always following the other. This pattern could be an important aspect to confirm or discard the behavior as suspicious. The speed of the vessels is more or less constant with the exception of some turning points where it decreases.

The application of the small multiples strategy also reveals that these vessels sailed on the same area every one of the 20 days of the available data, as Figure 6.9 shows.

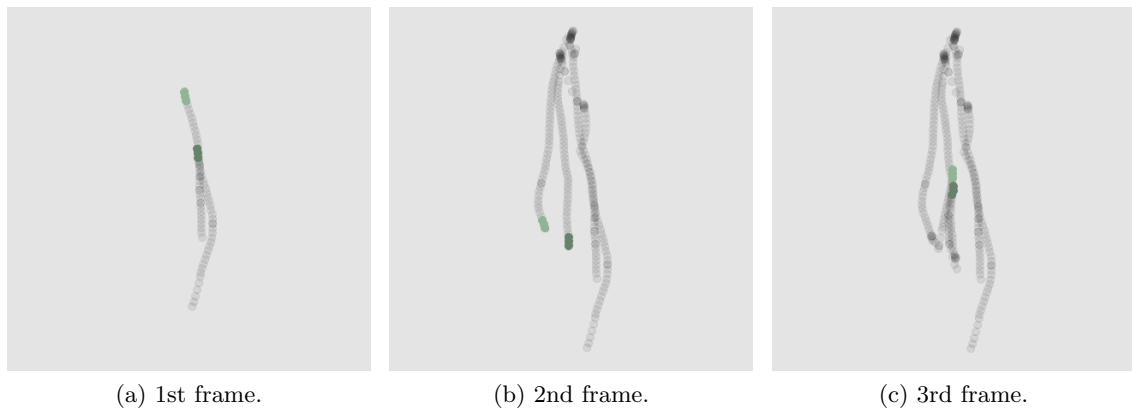


Figure 6.8: Three frames of the animation from the isolated intersection.

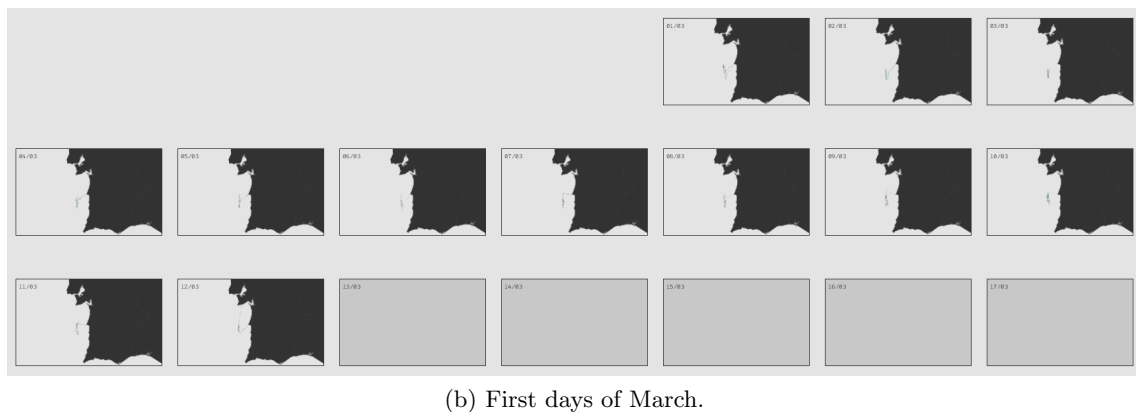
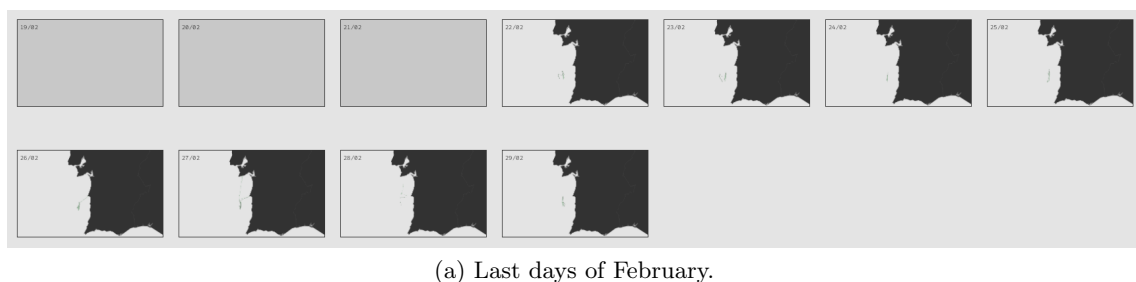


Figure 6.9: The small multiples approach, with the daily granularity, for the intersection trajectories.

6.3 Low Speed Intersections

The low speed filter is able to isolate the vessels that are sailing below the average speed of their type. When this filter is applied to the data from March 11 of 2012, a few intersections are detected near the north of Portugal, as Figure 6.10 shows. These intersections are also within an area of high density with the 3rd level of abnormality.

The first intersection between tankers is isolated by applying the fish-eye lens to it, as Figure 6.11 shows, and the trajectory followed by one of the tankers (the darker one) is suspicious because it was on left side and reduced the speed just to cross the trajectory of the other vessel, intersecting it.

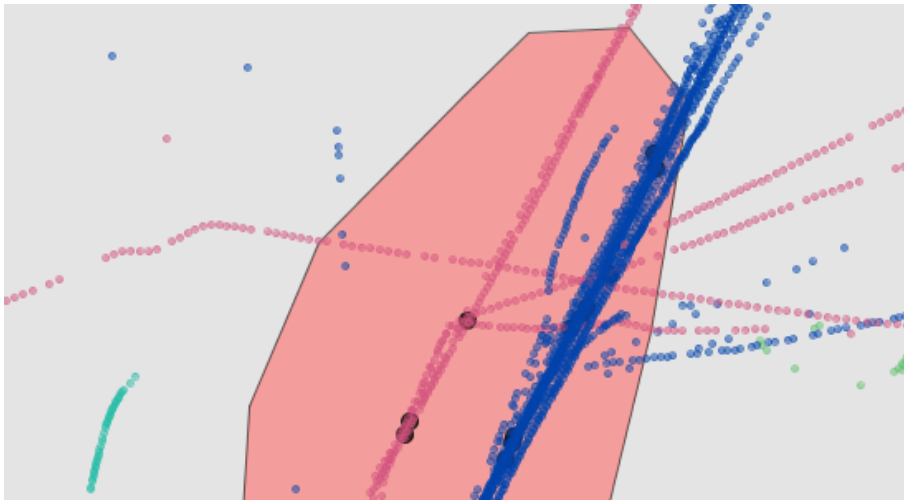
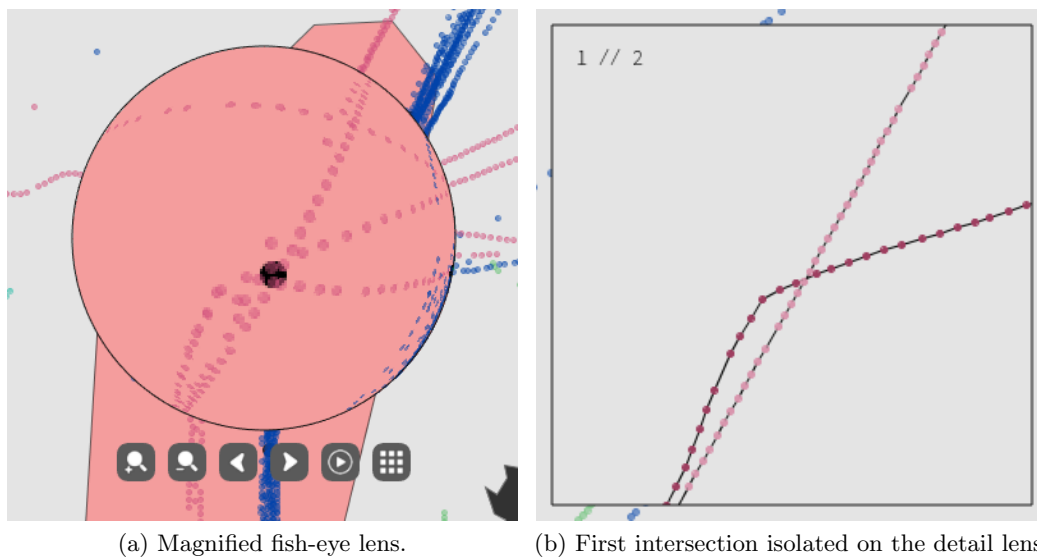


Figure 6.10: Visualization of low speed AIS trajectories.

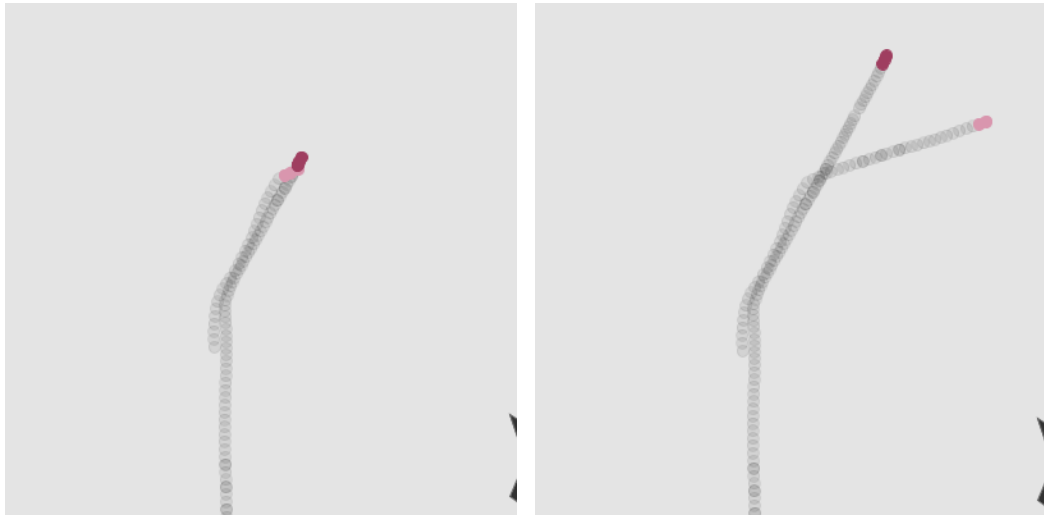


(a) Magnified fish-eye lens.

(b) First intersection isolated on the detail lens.

Figure 6.11: Identification and isolation of the first intersection with low speed.

The usage of the animation also revealed that the suspicious vessel started reducing the speed very late when compared to the other one, suggesting it was made on purpose, as Figure 6.12 shows. Image (a) shows the small curve made by the suspicious vessel and the moment where both intersect, and image (b) shows that after the intersection both follow different directions.



(a) Trajectories before the intersection.

(b) Trajectories after the intersection.

Figure 6.12: Animation of the trajectories from the intersection.

This page is intentionally left blank.

Chapter 7

Conclusions and Future Work

The AIS was originally implemented for assisting in maritime safety but its data is nowadays used for numerous proposes related with maritime traffic analysis. One of this proposes is the identification of specific abnormal behaviors with the goal of helping law enforcement authorities in detecting and responding to those actions. This thesis exploits the usage of data visualization techniques, assisted by data analysis, to achieve the exposed propose. Several visualization approaches were defined to address different categories of anomalous behaviors, being all implemented on a platform and validated through case studies. The 1st semester was mainly focused on studying the necessary background knowledge and related work for the thesis, and the 2nd semester was focused on the definition, implementation and validation of the visual strategies and data analysis tasks that are used for the different types of behaviors.

Considering the goals proposed for the thesis, one can consider that they were fully achieved. The developed platform implements visualization approaches to address all the types of anomalous behaviors that were initially identified by the domain experts. Moreover, the developed work originated two research articles that were submitted to a conference and a journal, and are both waiting review.

Regarding the work it-self, this thesis shows that the usage of visualization approaches on the AIS context can provide a good decision support system to help operators detect anomalous behaviors. The implemented strategies have presented good results for each of the anomalous categories addressed, as the case studies show.

In terms of future work, an obvious step is to test the developed strategies with real domain experts and evaluate them. Extending the categories of anomalous behaviors that are supported by the platform and improving its interaction and usability would also be important directions.

This page is intentionally left blank.

References

- Altera (2008). A flexible architecture for fisheye correction in automotive rear-view cameras. Technical report, Altera Corporation.
- Bertin, J. (1983). *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press.
- Bettonvil, F. (2005). Fisheye lenses. *WGN, Journal of the International Meteor Organization*, 33:9–14.
- Birant, D. and Kut, A. (2007). St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221.
- Bomberger, N. A., Rhodes, B. J., Seibert, M., and Waxman, A. M. (2006). Associative learning of vessel motion patterns for maritime situation awareness. In *Information Fusion, 2006 9th International Conference on*, pages 1–8.
- Bošnjak, R., Šimunović, L., and Kavran, Z. (2012). Automatic identification system in maritime traffic and error analysis. *Transactions on maritime science*, 1(02):77–84.
- Busler, J., Wehn, H., and Woodhouse, L. (2015). Tracking vessels to illegal pollutant discharges using multisource vessel information. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(7):927.
- Campello, R. J., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172.
- Cazzanti, L. and Pallotta, G. (2015). Mining maritime vessel traffic: Promises, challenges, techniques. In *OCEANS 2015-Genova*, pages 1–6.
- Chen, C., Wu, Q., Zhou, Y., and Mao, Z. (2016). Information visualization of ais data. In *2016 International Conference on Logistics, Informatics and Service Sciences (LISS)*, pages 1–8.
- Dobrkovic, A., Iacob, M.-E., and Van Hillegersberg, J. (2016). Maritime pattern extraction from ais data using a genetic algorithm. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 642–651.

-
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery & Data Mining*, pages 226–231.
- Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37.
- Fiorini, M., Capata, A., and Bloisi, D. D. (2016). Ais data visualization for maritime spatial planning (msp). *International Journal of e-Navigation and Maritime Economy*, 5:45–60.
- Gao, X. and Shiotani, S. (2013). An effective presentation of navigation information for prevention of maritime disaster using ais and 3d-gis. In *Oceans-San Diego, 2013*, pages 1–6.
- Gonzalez, J., Battistello, G., Schmiegelt, P., and Biermann, J. (2014). Semi-automatic extraction of ship lanes and movement corridors from ais data. In *2014 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1847–1850.
- Graham, R. L. (1972). An efficient algorithm for determining the convex hull of a finite planar set. *Information processing letters*, 1(4):132–133.
- Hadzagic, M., St-Hilaire, M.-O., Webb, S., and Shahbazian, E. (2013). Maritime traffic data mining using r. In *2013 16th International Conference on Information Fusion (FUSION)*, pages 2041–2048.
- Haines, E. (1994). Point in polygon strategies. *Graphics gems IV*, 994:24–26.
- Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *proceedings of the 17th international conference on data engineering*, pages 215–224.
- Handayani, D. O. D., Sediono, W., and Shah, A. (2013). Anomaly detection in vessel tracking using support vector machines (svms). In *2013 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, pages 213–217.
- Jacobs, F. (2016). Portugal is 97 Available at <http://bigthink.com/strange-maps/652-nil-jellyfish-nation-portugal-is-97-water>. Accessed on 19-12-2017.
- Jiacai, P., Qingshan, J., Jinxing, H., and Zheping, S. (2012). An ais data visualization model for assessing maritime traffic situation and its applications. *Procedia Engineering*, 29:365–369.
- Kriegel, H.-P., Kröger, P., Sander, J., and Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240.
- Laxhammar, R. (2008). Anomaly detection for sea surveillance. In *2008 11th International Conference on Information Fusion*, pages 1–8.

- Lee, J.-G., Han, J., and Li, X. (2008a). Trajectory outlier detection: A partition-and-detect framework. In *2008 IEEE 24th International Conference on Data Engineering (ICDE)*, pages 140–149.
- Lee, J.-G., Han, J., Li, X., and Gonzalez, H. (2008b). Traclass: trajectory classification using hierarchical region-based and trajectory-based clustering. *Proceedings of the VLDB Endowment*, 1(1):1081–1094.
- Lee, J.-G., Han, J., and Whang, K.-Y. (2007). Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604.
- Lei, P.-R., Tsai, T.-H., and Peng, W.-C. (2016). Discovering maritime traffic route from ais network. In *Network Operations and Management Symposium (APNOMS), 2016 18th Asia-Pacific*, pages 1–6.
- Leick, A., Rapoport, L., and Tatarnikov, D. (2015). *GPS satellite surveying*. John Wiley & Sons.
- Li, Y., Liu, R. W., Liu, J., Huang, Y., Hu, B., and Wang, K. (2016). Trajectory compression-guided visualization of spatio-temporal ais vessel density. In *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–5.
- Li, Z., Ding, B., Han, J., Kays, R., and Nye, P. (2010). Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1099–1108.
- Liu, B. (2015). Maritime traffic anomaly detection from ais satellite data in near port regions. Master’s thesis, Dalhousie University.
- Liu, C. and Chen, X. (2013). Inference of single vessel behaviour with incomplete satellite-based ais data. *The Journal of Navigation*, 66(6):813–823.
- Liu, T.-K., Sheu, H.-Y., and Chen, Y.-T. (2015). Utilization of vessel automatic identification system (ais) to estimate the emission of air pollutant from merchant vessels in a port area. In *OCEANS 2015-Genova*, pages 1–5.
- Lutins, E. (2017). DbSCAN: What is it? when to use it? how to use it. Available at <https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818>. Accessed on 08-11-2017.
- Maling, D. H. (2013). *Coordinate systems and map projections*. Pergamon Press.
- Mascaro, S., Nicholso, A. E., and Korb, K. B. (2014). Anomaly detection in vessel tracks using bayesian networks. *International Journal of Approximate Reasoning*, 55(1):84–98.

-
- Mazimpaka, J. D. and Timpf, S. (2016). Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science*, 2016(13):61–99.
- Mazzarella, F., Arguedas, V. F., and Vespe, M. (2015). Knowledge-based vessel position prediction using historical ais data. In *Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2015*, pages 1–6.
- Mazzarella, F., Vespe, M., Damalas, D., and Osio, G. (2014). Discovering vessel activities at sea using ais data: Mapping of fishing footprints. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–7.
- Nanni, M. and Pedreschi, D. (2006). Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27(3):267–289.
- Osekowska, E., Axelsson, S., and Carlsson, B. (2015). Potential fields in modeling transport over water. In *Transport of Water versus Transport over Water*, pages 259–280.
- Pallotta, G., Vespe, M., and Bryan, K. (2013a). Traffic knowledge discovery from ais data. In *2013 16th International Conference on Information Fusion (FUSION)*, pages 1996–2003.
- Pallotta, G., Vespe, M., and Bryan, K. (2013b). Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction. *Entropy*, 15(6):2218–2245.
- Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.
- Rhodes, B. J., Bomberger, N. A., Seibert, M., and Waxman, A. M. (2005). Maritime situation monitoring and awareness using learning mechanisms. In *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 646–652.
- Riveiro, M. and Falkman, G. (2009). Interactive visualization of normal behavioral models and expert rules for maritime anomaly detection. In *2009 Sixth International Conference on Computer Graphics, Imaging and Visualization (CGIV'09)*, pages 459–466.
- Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.
- Sang, L.-Z., Yan, X.-P., Mao, Z., and Ma, F. (2012). Restoring method of vessel track based on ais information. In *2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science (DCABES)*, pages 336–340.
- Sang, L.-z., Yan, X.-p., Wall, A., Wang, J., and Mao, Z. (2016). Cpa calculation method based on ais position prediction. *The Journal of Navigation*, 69(6):1409–1426.
- Seriai, A., Benomar, O., Cerat, B., and Sahraoui, H. (2014). Validation of software visualization tools: A systematic mapping study. In *2014 Second IEEE Working Conference on Software Visualization (VISSOFT)*, pages 60–69.

- Software, C. (2017). *Oversee*. Available at <https://www.criticalsoftware.com/pt/products/p/oversee>. Accessed on 22-12-2017.
- Soleimani, B. H., De Souza, E. N., Hilliard, C., and Matwin, S. (2015). Anomaly detection in maritime data based on geometrical analysis of trajectories. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1100–1105.
- Sun, F., Deng, Y., Deng, F., Zhu, Q., and Chu, H. (2015). Unsupervised maritime traffic pattern extraction from spatio-temporal data. In *2015 11th International Conference on Natural Computation (ICNC)*, pages 1218–1223.
- Tetreault, B. J. (2005). Use of the automatic identification system (ais) for maritime domain awareness (mda). In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1590–1594.
- Tu, E., Zhang, G., Rachmawati, L., Rajabally, E., and Huang, G. B. (2017). Exploiting ais data for intelligent maritime navigation: A comprehensive survey from data to methodology. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–24.
- Tufte, E. R. (1986). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA.
- Ward, M., Grinstein, G., and Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. A. K. Peters, Ltd., Natick, MA, USA.
- Willems, N., Van De Wetering, H., and Van Wijk, J. J. (2009). Visualization of vessel movements. In *Computer Graphics Forum*, volume 28, pages 959–966.
- Wu, X., Wu, L., Xu, Y., An, Z., and Diao, B. (2015). Vessel trajectory partitioning based on hierarchical fusion of position data. In *Information Fusion (Fusion), 2015 18th International Conference on*, pages 1230–1237.
- Xu, W., Zhong, D., Wu, S., and Ni, H. (2015). A track fusion method of a vessel. In *Sixth International Conference on Electronics and Information Engineering*, pages 1–5.
- Yan, W., Wen, R., Zhang, A. N., and Yang, D. (2016). Vessel movement analysis and pattern discovery using density-based clustering approach. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3798–3806.
- Yang, C.-S., Kim, T.-H., Hong, D., and Ahn, H.-W. (2013). Design of integrated ship monitoring system using sar, radar, and ais. In *Proc. of SPIE Vol*, volume 8724, pages 872411–1.
- Zhang, D., Li, J., Wu, Q., Liu, X., Chu, X., and He, W. (2017). Enhance the ais data availability by screening and interpolation. In *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, pages 981–986.
- Zhang, D., Li, N., Zhou, Z.-H., Chen, C., Sun, L., and Li, S. (2011). ibat: detecting anomalous taxi trajectories from gps traces. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 99–108.

Zhang, S.-k., Liu, Z.-j., Cai, Y., Wu, Z.-l., and Shi, G.-y. (2016). Ais trajectories simplification and threshold determination. *The Journal of Navigation*, 69(4):729–744.

Appendices

Appendix A

Density-Based Clustering Results

A.1 Silhouette Coefficient Results

Table A.1: Silhouette coefficients for density-based clustering algorithms with $MinPts = 25$.

Chunk	HDBSCAN	DBSCAN		
		$\epsilon = 250$	$\epsilon = 500$	$\epsilon = 750$
1	0.472	0.376	0.511	0.557
2	0.280	0.201	0.554	0.365
3	0.329	0.183	0.475	0.351
4	0.267	0.241	0.285	0.523
5	0.087	0.175	0.494	0.481
6	0.430	0.359	0.508	0.550
7	0.522	0.199	0.368	0.508
8	0.460	0.355	0.452	0.513
9	0.050	0.425	0.578	0.582
10	0.419	0.212	0.532	0.506
11	0.351	0.256	0.459	0.503
12	0.294	0.305	0.432	0.519
13	0.404	0.407	0.549	0.509
14	0.197	0.256	0.368	0.551
15	0.177	0.304	0.559	0.469
16	0.208	0.366	0.538	0.463
17	0.241	0.247	0.303	0.656
18	0.335	0.266	0.194	0.439
19	0.076	0.289	0.174	0.426
20	0.485	0.396	0.295	0.514

Table A.2: Silhouette coefficients for density-based clustering algorithms with $MinPts = 50$.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
1	0.500	0.253	0.457	0.515
2	0.586	0.296	0.495	0.535
3	0.591	0.033	0.440	0.512
4	0.325	0.154	0.520	0.281
5	0.334	0.109	0.385	0.502
6	0.378	0.320	0.441	0.386
7	0.563	0.021	0.298	0.403
8	0.477	0.135	0.430	0.491
9	0.505	0.339	0.515	0.592
10	0.477	0.310	0.439	0.533
11	0.460	0.304	0.461	0.148
12	0.473	0.287	0.359	0.480
13	0.464	0.347	0.435	0.529
14	0.261	0.251	0.481	0.347
15	0.478	0.211	0.498	0.383
16	0.514	0.299	0.523	0.598
17	0.420	0.285	0.410	0.309
18	0.403	0.139	0.347	0.393
19	0.372	0.181	0.317	0.370
20	0.583	0.326	0.443	0.315

Table A.3: Silhouette coefficients for density-based clustering algorithms with $MinPts = 75$.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
1	0.610	0.123	0.399	0.502
2	0.605	0.182	0.421	0.503
3	0.604	-0.032	0.403	0.502
4	0.348	0.116	0.501	0.530
5	0.553	0.023	0.443	0.452
6	0.401	0.218	0.458	0.504
7	0.607	-0.171	0.191	0.263

Continues in next page.

Continued from previous page.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
8	0.531	-0.006	0.407	0.452
9	0.572	0.277	0.474	0.512
10	0.507	0.234	0.444	0.501
11	0.550	0.267	0.446	0.406
12	0.506	0.189	0.399	0.463
13	0.515	0.254	0.435	0.507
14	0.490	0.197	0.384	0.488
15	0.501	0.243	0.410	0.511
16	0.633	0.315	0.478	0.584
17	0.556	0.188	0.351	0.295
18	0.480	0.119	0.323	0.529
19	0.339	0.117	0.308	0.445
20	0.590	0.247	0.470	0.516

Table A.4: Silhouette coefficients for density-based clustering algorithms with $MinPts = 100$.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
1	0.613	-0.023	0.339	0.456
2	0.500	0.102	0.377	0.472
3	0.563	0.059	0.353	0.453
4	0.348	0.055	0.473	0.513
5	0.320	-0.047	0.389	0.436
6	0.397	0.159	0.425	0.502
7	0.518	-0.156	0.152	0.243
8	0.448	-0.009	0.358	0.438
9	0.549	0.172	0.465	0.517
10	0.550	0.163	0.430	0.432
11	0.465	0.199	0.435	0.440
12	0.469	0.119	0.395	0.371
13	0.531	0.193	0.445	0.477
14	0.477	0.098	0.300	0.458
15	0.558	0.169	0.382	0.479
16	0.612	0.203	0.465	0.438

Continues in next page.

Continued from previous page.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
17	0.565	0.124	0.390	0.462
18	0.589	0.049	0.365	0.440
19	0.502	0.061	0.287	0.327
20	0.568	0.166	0.449	0.484

A.2 Number of Extracted Clusters

Table A.5: Number of clusters extracted from density-based clustering algorithms with $MinPts = 25$.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
1	52	29	29	27
2	70	35	33	33
3	58	30	27	24
4	79	33	31	29
5	112	34	35	29
6	93	50	43	38
7	29	17	23	19
8	40	20	26	24
9	133	39	38	33
10	77	38	40	36
11	88	38	36	31
12	97	34	38	34
13	97	39	37	35
14	107	45	39	35
15	97	39	37	35
16	88	39	39	34
17	94	47	36	33
18	74	32	27	26
19	78	29	30	29
20	71	40	42	37

Table A.6: Number of clusters extracted from density-based clustering algorithms with $MinPts = 50$.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
1	20	17	19	16
2	27	21	21	19
3	29	21	20	17
4	38	23	19	19
5	39	22	23	21
6	46	33	26	25
7	18	7	14	12
8	18	12	14	15
9	37	27	25	23
10	43	23	27	22
11	34	19	21	20
12	35	22	18	18
13	42	30	26	24
14	46	28	23	22
15	40	27	25	22
16	39	31	31	28
17	39	30	33	24
18	39	24	22	20
19	31	21	23	19
20	34	27	24	23

Table A.7: Number of clusters extracted from density-based clustering algorithms with $MinPts = 75$.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
1	17	16	17	16
2	17	16	17	15
3	18	14	17	16
4	26	20	17	17
5	26	22	19	21
6	36	21	22	19
7	15	4	6	7

Continues in next page.

Continued from previous page.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
8	14	7	9	9
9	23	20	20	20
10	27	23	21	23
11	23	18	17	16
12	26	20	19	16
13	31	21	21	20
14	29	19	22	19
15	32	21	24	21
16	28	24	24	21
17	25	19	22	22
18	23	16	20	17
19	29	16	16	17
20	28	22	23	22

Table A.8: Number of clusters extracted from density-based clustering algorithms with $MinPts = 100$.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
1	16	11	15	14
2	19	15	15	13
3	16	12	16	14
4	23	15	16	14
5	24	19	16	14
6	30	15	21	18
7	9	3	5	7
8	12	6	10	9
9	20	19	18	17
10	23	20	20	22
11	20	19	15	14
12	22	18	16	16
13	27	17	18	19
14	23	15	20	16
15	20	16	18	16
16	23	17	20	19

Continues in next page.

Continued from previous page.

Chunk	HDBSCAN	DBSCAN		
		$\varepsilon = 250$	$\varepsilon = 500$	$\varepsilon = 750$
17	20	17	18	18
18	18	13	18	17
19	21	14	15	16
20	22	19	19	18

Table A.9: Average number of clusters extracted from density-based clustering algorithms.

Algorithm	Average	Std. Deviation
HDBSCAN 25	81.700	24.671
HDBSCAN 50	34.700	8.511
HDBSCAN 75	24.650	5.941
HDBSCAN 100	20.400	4.751
DBSCAN 25 250	35.350	8.061
DBSCAN 25 500	34.300	5.723
DBSCAN 25 750	31.050	5.042
DBSCAN 50 250	23.250	6.398
DBSCAN 50 500	22.700	4.824
DBSCAN 50 750	20.450	3.762
DBSCAN 75 250	17.950	5.042
DBSCAN 75 500	18.650	4.580
DBSCAN 75 750	17.700	4.118
DBSCAN 100 250	15.000	4.389
DBSCAN 100 500	16.450	3.692
DBSCAN 100 750	15.550	3.426