Numéro d'ordre NNT: 2022ECDL0014 Année: 2022

THESE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
OPÉRÉE AU SEIN DE L'ÉCOLE CENTRALE DE LYON

ÉCOLE DOCTORALE InfoMaths

Spécialité: Informatique

Soutenue le 25/11/2022 par

Liqun Liu

Visualization of Spatial and Temporal Road Traffic Data

Devant le jury composé de:

Sidonie Christophe (DR)	Univ. Gustave Eiffel, ENSG, IGN	Rapportrice
Christophe Hurter (PR)	Ecole Nationale de l'Aviation Civile	Rapporteur
Gilles Gesquière (PR)	Université Lumière Lyon 2	Président
Arnaud Prouzeau (CR)	Inria Bordeaux	Examinateur
Stéphane Derrode (PR)	École Centrale de Lyon	Co-Directeur de thèse
Romain Vuillemot (MdC)	École Centrale de Lyon	Co-Directeur de thèse

Acknowledgements

I eventually finished my dissertation and my PhD period of study. First, I would like to say a big thank you for all the help from my two supervisors during the past four years. Prof. Stéphane Derrode brought me here and allowed me to continue my research as a PhD student at Ecole Centrale de Lyon (ECL). Dr. Romain Vuillemot taught me how to do research in the visualization community. I appreciate all suggestions you have given me on every research topic and the work you have done in every published paper.

Besides, I would like to show my greatest appreciation to my defence jury. First, the report written by Prof. Sidonie Christophe and Prof. Christophe Hurter inspired me in many aspects to improve the quality of my dissertation, as well as to provide the idea for my future research. Furthermore, I would like to thank Prof. Gilles Gesquiere and Dr. Arnaud Prouzeau for their gentle and insightful questions during my defence.

I am also thankful to the SICAL team. We have the weekly meeting on Friday. During this meeting, I got many ideas and valuable feedback from team members and others' presentations. Also, all my research works have been presented in our team meeting and I received much valuable feedback that is the solid foundation for my publications.

In addition, I would like to thank my colleagues who came into my life during the past four years. We might have different native languages and different cultural backgrounds. However, you are super friendly and always give your heartful help to me. Especially for my defence, you gave me many valuable suggestions on my presentation at rehearsal.

During these four years, the friends whom I met in Lyon made my life more colorful. Going to Carrefour and walking around ECL campus were our regular weekly activities. The BBQ and hotpot times were always full of joy, which is an important part of my memory.

Playing basketball is my favorite activity. That can relieve my pressure. I would like to thank my small basketball groups. I will not forget our happy time (every Saturday afternoon) and the people who played basketball with me.

I thank my parents for their support. We always show the most gentle side to outsiders and leave our lousy temper to those closest to us. I appreciate my parents for their patience and for giving me the greatest consolation when I was in a bad mood.

Apart from that, I cannot forget to thank my partner for her support. She encouraged me when I had problems with my research. She gave me suggestions for the issues I faced both in my life and in my work. She is a 'fighter', my teammate in life, and my lover.

Finally, thanks to the people who helped me during this challenging but worthwhile four-year experience. It is one of the most critical steps in my life. I cherish the four-year experiences that will be saved in my deep heart for my whole life.

Contents

List of Figures

List of Tables

Abstract

This manuscript deals with the design of novel methods for the visual analysis of road traffic data. As recent technological advances allow traffic analysis with a finer and more varied spatial and temporal granularity, at several scales, both local and global, from the level of a street to an agglomeration. Thus experts and operators of road checkpoints can explore these large volumes of data in a non-technical way, identify interesting patterns and make better decisions informed by the data (*e.g.,*to reduce traffic congestion).

The first part of the manuscript addresses the problem of univariate analysis (found in *e.g.,* traffic density and traffic flow data) by proposing an interactive categorization method named FuzzyCut. This method is based on the fuzzy logic theory by proposing an interactive version of the category membership function. We introduce the interactions and the design of this technique, as well as its implementation on different types of quantitative data (*e.g.,* traffic densities and taxi speeds). This technique has also undergone a user evaluation and its code and evaluation data are available online.

The second part focuses on the analysis of the spatial component of road traffic which is inherent in this data type. We propose the adaptation of an existing technique called Origin-Destinations maps, that preserves both explicit relationships (spatial trajectories) and implicit relationships (abstract attributes of those trajectories) of datasets, using spatial nesting: a first level of the map encodes the origin (starting point of objects), and a second nested level encodes the destination (ending point of objects) in cells nested on the map. We generalize this technique beyond origins and destinations relationships (2-attribute datasets) to explore multi-dimensional datasets (N-attribute datasets). We present an abstraction framework, Gridify, and its implementation as an interactive open-source tool with several levels of nested maps to explore the relations of geo-coded entities (location or object) with multi-dimensional attributes.

The third part focuses on the problem of temporal analysis which is also an important component of traffic flows. We propose GroupSet, a technique to explore temporal changes using a set-based approach. Such exploration reveals elements' patterns and similarities, such as increases or decreases in traffic flow values during a day. We demonstrate the technique's applicability to traffic flow and report on usability feedback of an interactive prototype implementing the technique.

The last section discusses the three techniques introduced in the manuscript (FuzzyCut, Gridify and GroupSet). First, how to deploy them in road traffic control centers, within a unified prototype. Secondly, their application beyond road traffic data, as generic tools for the analysis of univariate, spatial and time-varying data.

Keywords: Data visualization, Road traffic data.

Résumé

Ce manuscrit porte sur la conception de nouvelles méthodes d'analyse visuelle de données de trafic routier. En effet, les avancées technologiques récentes permettent une analyse du trafic avec une granularité spatiale et temporelle plus fine et variée, au moyen de capteurs ou de boitiers GPS. La visualisation permet de mieux explorer ces données, à plusieurs échelles, aussi bien globales que locales, au niveau d'une agglomération jusqu'à un segment de rue. Ainsi, experts et opérateurs de postes de contrôle routier peuvent explorer de manière non-technique ces grands volumes de données, identifier des motifs intéressants et prendre une décision informée par les données.

La première partie de la thèse aborde le problème de l'analyse univariée (que l'on trouve dans les données de densité de trafic ou de flux de voiture) en proposant une méthode interactive de catégorisation nommée FuzzyCut. Cette méthode est basée sur la théorie de la logique floue en proposant une version interactive de la fonction d'appartenance à une catégorie. Nous introduisons les interactions et le design de cette technique, ainsi que sa mise en œuvre sur différents types de données quantitatives univariées. Cette technique a aussi fait l'objet d'une évaluation utilisateur et son code et les données de l'évaluation sont disponibles en ligne.

La seconde partie porte sur l'analyse de la composante spatiale du trafic routier qui est inhérante à ce type de données. Nous proposons l'adaptation d'une technique existante de visualisation de segments de trajectoires sous formes d'origines et destinations. Cette méthode permet de visualiser à la fois des relations globales et locales des données, en utilisant l'imbrication spatiale, où un premier niveau de la carte encode l'origine (point de départ des objets), et un second niveau imbriqué encode la destination (points d'arrivée des objets). Nous généralisons cette technique au-delà des relations d'origine et de destination (qui est un jeu de données à 2 attributs) pour explorer les ensembles de données multidimensionnels (ex. données à N attributs). Nous présentons un cadre d'abstraction, Gridify, et son implémentation en outil open-source interactif avec plusieurs niveaux de cartes imbriquées pour explorer les relations d'entités géocodées (lieu ou objet) avec des attributs multidimensionnels.

La troisième partie se concentre sur le problème d'analyse temporelle qui est aussi une composante importante des flux de trafic. Nous proposons GroupSet, une technique pour explorer les changements au fil d'une journée ou d'une année en utilisant une approche basée sur la théorie des ensembles. Une telle exploration révèle les similitudes de comportement temporel des données, telles que les augmentations ou les diminutions du flux de trafic au cours d'une journée. Cette technique a plusieurs applications, au-delà du trafic routier, pour l'analyse des séries temporelles. Nous faisons un retour d'expérience d'utilisabilité d'un prototype interactif mettant en œuvre la technique sous forme d'application web.

La dernière section discute la combinaison de trois techniques introduites dans le manuscrit (FuzzyCut, Gridify et GroupSet) au sein d'un même environnement d'analyse. En particulier afin de les déployer dans des centres de contrôle du trafic routier, pour réaliser des tâches de suivi et d'analyse de flux de voitures. Nous discutons également leur application au-delà des données de trafic routier, comme outils génériques d'analyse de données univariées, de données géo-codées et de séries temporelles.

Mots-clés: Visualisation de données, Données de trafic routier.

CHAPTER 1
Introduction

Contents

Any use of ”we” in this chapter refers to Liqun Liu and Romain Vuillemot.

1.1 Context and Motivation

Road traffic[1] has been in constant and rapid growth over the past decades, which helped economic development at the local and global scale [32]. But since then, such traffic has reached levels of density that generate traffic congestion, where vehicles are slowly moving or even stuck on the road (*e.g.,* Figure 1.1). Such road traffic congestion problem has become one of the most severe urban-related issues worldwide. It generates many negative externalities that impact social and economics life of cities. According to a report released by INRIX [33] (a private company providing traffic-relevant data, such as real-time and historical traffic flows), in 2019, traffic congestion has cost 17.1€ billion in France because of the loss of work time or increased wear and tear of vehicles. Besides, the extra travel time caused by traffic congestion exacerbates another problem: air pollution, which results from the fossil fuel burned by car engines. According to a study by Khreis [34], in 2015, traffic-related air pollution was a high proportion of city pollution that were 24% in Toronto (Canada), 66% in Beijing (China), 67% in Paris (France).

In the meantime, there is now a wealth of data sources to better understand road traffic behaviors, such as traffic flow, taxi trajectories, traffic events, and webcam (video) data (more data sources will be provided and detailed in Section 2.1.5). For instance, most taxis in cities now require to install GPS to collect their status data every minute or seconds (*e.g.,* taxis collect status data 3 or 4 times every minute). Also, the increase of CCTV (Closed-Circuit Television Camera) for road security purposes or public webcams for general traffic information, provide new ways to automatically quantify and characterize what occurs on roads. The volume, temporality, and multi-dimensionality of such datasets require a better presentation to experts who do not have the technical skills to query those data in their raw format and enable automated analysis.

[1] It represents travel and transportation in public ways (roads), including vehicles, trains, pedestrians, or other conveyances.

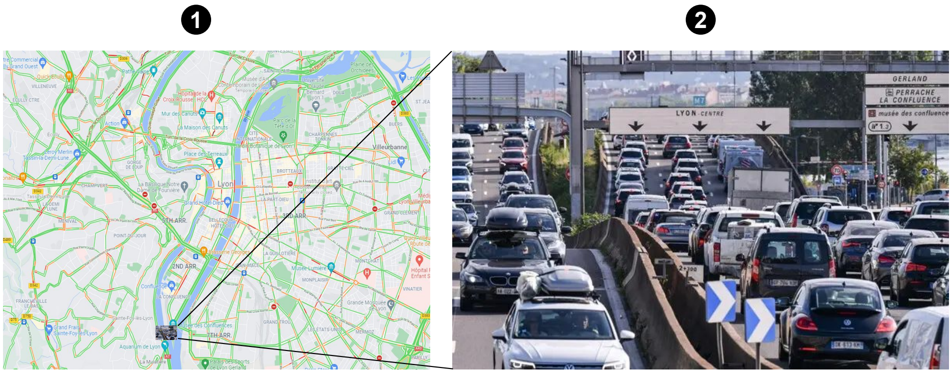


Figure 1.1: Example of traffic congestion in Lyon, France. ① is a traffic map (Google Map) [1] that shows the traffic situations (*e.g.,* traffic congestion) in Lyon. ② is a photograph taken from the selected road segment (Pont ferroviaire de la Mulatière) showing the severe road traffic congestion [2].

The field of Information Visualization (InfoVis) aims at providing such visual tools to help humans interactively explore data. InfoVis is "*the use of computer-supported, interactive visual representations of numerical and non-numerical abstract datasets in order to amplify human cognition*" [35]. The formats of InfoVis generally involve multiple visual representations from basic ones (*e.g.,* bar graph, histogram, line charts) to more advanced ones (*e.g.,* networks, graphs). The application of InfoVis to road traffic visualization is already very rich, mainly using maps to visualize the abstract data containing coordinates (*e.g.,* traffic flows and taxi trajectories as we will review in Section 2.1.5). InfoVis helps humans understand, explore and analyze a large of useful information in intuitive and interactive ways enabling users to compare different values, show the bigger picture, track trends in the data, and understand different relationships between variables, among others.

InfoVis techniques serve not only as communication mediums, but also as Exploratory Data Analysis (EDA) mediums—the approach of analyzing datasets by summarizing the data characteristics with visualization methods [36]. During this stage, few assumptions in data distribution and quality can be made, so automatic methods (*e.g.,* clustering, classification) may not be applicable directly. Such issue is frequent when collecting real-world data, such as road traffic data, which needs methods to not only analyze traffic patterns but assess the quality and distribution before any analytical process. Once such exploratory steps have been conducted, users can pick a model to facilitate or automate analysis in a more informed way [37].

Such need for exploratory traffic road data visualizations was raised while collaborating with experts[2] who provided us with road traffic datasets, lists of routine tasks to achieve and access to traffic control centers in Lyon and Paris to observe their work in real settings. We focused mainly on road traffic congestion data exploration and communication by developing novel interactive techniques aimed at those experts (in opposition to a general audience). This manuscript addresses four visual analysis questions related to road traffic data (summarized in Figure 1.2):

[2] This work was conducted in conjunction with the MI2 (Mobilité Intégrée Île-de-France) project https://projet.liris.cnrs.fr/mi2/ which aimed at improving multimodal mobility in French large urban areas.

[3] https://www.google.com/maps/

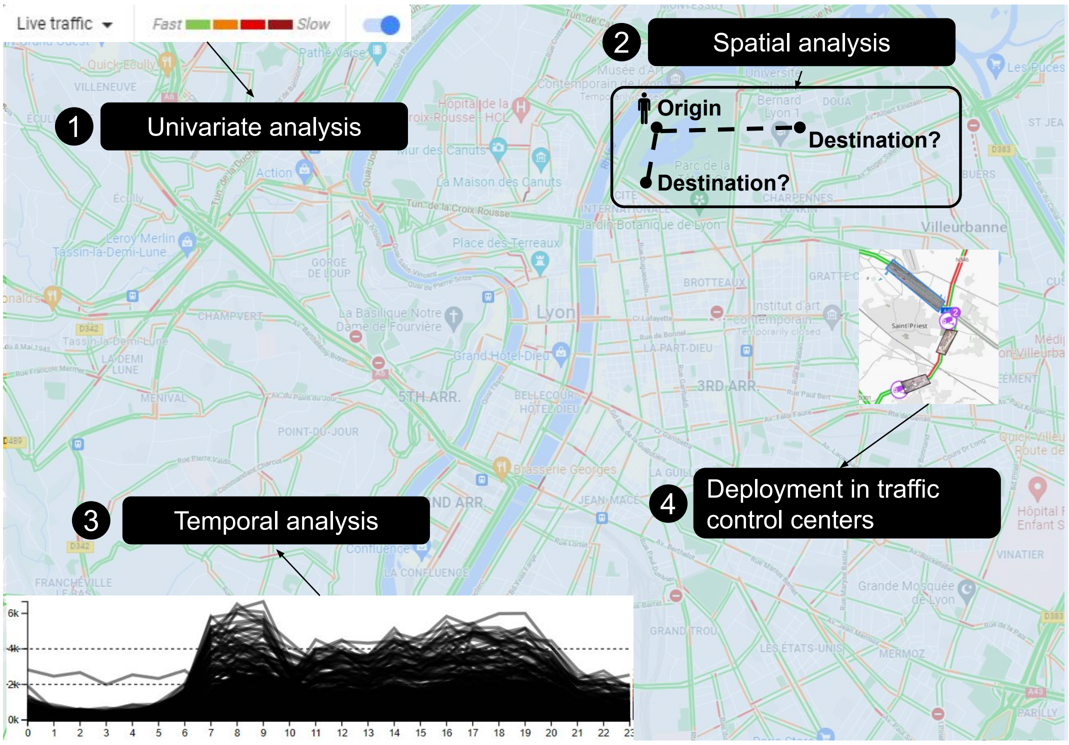


Figure 1.2: Above is an illustration of the four challenges we address in this manuscript: *"how to categorize traffic flows?"*① which we address as a univariate analysis problem. *"how do taxis travel in a city?"* ② which we address as a spatial analysis problem. *" how traffic flows change over time?"* ③ which we address as a temporal analysis *"how to analyze the temporal and spatial traffic information simultaneously?"*④ It refers to the deployment of different visualization techniques in one dashboard of traffic control centers. Map credits: Google Map [3].

- ***"How to categorize traffic flows?"*** Traffic flow data can be regarded as univariate data, *i.e.* observations over a single attribute. Dealing with univariate data is apparently a simple form of data analysis, but it raises challenges such as *categorization* (*i.e.* separating a continuous scale into intervals). This is one of the main topics in single variable analysis [38], and is frequent for traffic flow understanding and communication (*e.g.,* to design color scales).

- ***"How do taxis travel in a city?"*** The analysis of this question enables traffic experts to better know humans' behaviors to explore the real reason for traffic congestion (*e.g.,* it might be because humans have a particular commuting behavior). This is a spatial analysis that seeks to explain, mathematically and geometrically, patterns of human behavior and their spatial representation [39].

- ***"How do traffic flows change over time?"*** Analyzing how traffic flows change over time enables traffic experts better know if roads have traffic congestion and when they occur. This information helps them better guide the vehicles to reduce road traffic pressure. Traffic flow data is the typically temporal data that changes along with a sequence of

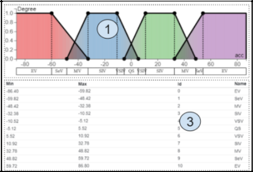
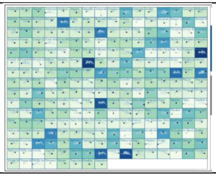
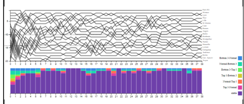
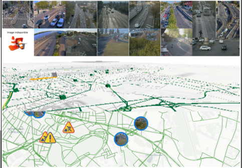
Tool Design space	FuzzyCut	Gridify	GroupSet	ControlCenter
Users	• Citizens • Urban planners • Transport planners • Traffic control centers	• Citizens • Urban planners • Transport planners • Traffic control centers	• Citizens • Urban planners • Transport planners • Traffic control centers	• Citizens • Urban planners • Transport planners • Traffic control centers
Tasks	• Monitor traffic • Pattern discovery and clustering • Situation-aware exploration and prediction	• Monitor traffic • Pattern discovery and clustering • Situation-aware exploration and prediction	• Monitor traffic • Pattern discovery and clustering • Situation-aware exploration and prediction	• Monitor traffic • Pattern discovery and clustering • Situation-aware exploration and prediction
Data sources	• Intusive sensor • Non-intusive sensor • Off-roadway sensor • Simulated data source	• Intusive sensor • Non-intusive sensor • Off-roadway sensor • Simulated data source	• Intusive sensor • Non-intusive sensor • Off-roadway sensor • Simulated data source	• Intusive sensor • Non-intusive sensor • Off-roadway sensor • Simulated data source
Data types	Quantitative data	Spatial data	Temporal data	Heterogeneous spatio-temporal Data
Visualization types	Univariate visualization	Spatial visualization	Temporal visualization	Multiple coordinated views
Visualization previews				
Applications beyond traffic data	• Fuzzy inference systems • Quantities categorization	• Soccer players trajectory • World trade	• Machine learning • Ranking data	

Figure 1.3: Summary of the contributions in this manuscript. We designed three novel visualization techniques considering each user, task, and data type. We explored the application beyond traffic data with three of them (FuzzyCut, Gridify, and GroupSet). We also discuss deploying multiple visualization techniques in traffic control centers (Figure 1.4) with ControlCenter.

timestamps. Thus, our approach focuses on exploring the changing patterns of traffic flows, addressing it as the temporal analysis problem.

- *"How to analyze temporal and spatial traffic information simultaneously?"* We introduce both temporal and spatial analysis problems separately so far. However, analysis of road traffic datasets usually requires a simultaneous analysis using a single visualization environment. In particular, in traffic control centers, traffic operators monitor situations through heterogeneous road traffic data using wall-display dashboards. Thus, we focus on how to deploy multiple visualization techniques in traffic control centers.

1.2 Contributions and Publications

The main contribution of the manuscript is developing novel road traffic interactive visualizations to assist traffic experts in discovering and communicating patterns hidden in road traffic data. Figure 1.3 provides more details on the specific users, tasks, data sources, and data types that each contribution addresses. The manuscript introduces three novel visualization techniques (FuzzyCut, Gridify, and GroupSet), implemented as interactive prototypes available

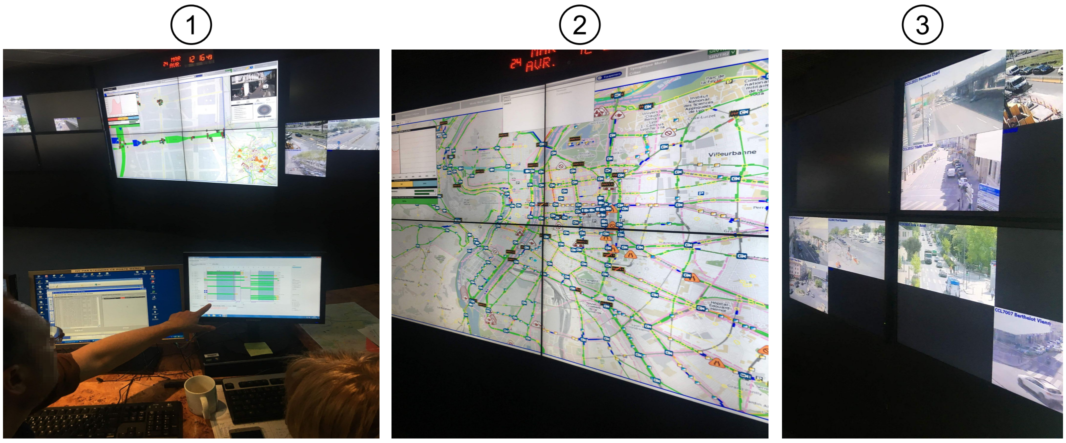


Figure 1.4: Our visual design challenges come from traffic control centers (pictures are from the control center in Lyon). A control center consists of multiple wall-display screens (①) where a map (②) offers an overview of road segments and webcams (③) monitor specific road intersections.

as web applications. We also discuss how to deploy multiple visualization techniques using ControlCenter in a dashboard in traffic control centers and how each visualization can be applied beyond traffic data. We introduce now detail our contributions and publications as follows (that match the four visual questions we introduced in the previous section):

Contribution "FuzzyCut" (Chapter 3): We introduce a novel visualization technique to categorize univariate data, such as traffic flow. It relies on fuzzy logic theory, particularly on the membership function [40] that maps values to categories with a confidence degree. We investigate how an *interactive* version of the membership function can be used to categorize quantitative data. We report on implementing the interactive function for several case studies with quantitative data (*e.g.,* traffic densities and taxi speeds). After that, we report on a formal user evaluation to investigate how users categorize quantities using the technique. We have published a paper from this work as follows:

- Liqun Liu and Romain Vuillemot. "Categorizing Quantities using an Interactive Fuzzy Membership Function," In *The* 12*th International Conference on Information Visualisation Theory and Applications*, P. 8, On-line, Feb 2021. (Link)

Contribution "Gridify" (Chapter 4): We propose a novel visualization technique to explore spatial data relations, such as taxi trajectories datasets. It relies on an existing technique called Origin-Destinations maps that uses spatial nesting, where a first level of the map encodes the origins (starting point of geographic objects) and a second nested level encodes the destinations (ending point of geographic objects). We generalize this technique beyond origins and destinations relationships (2-attribute datasets) to explore multi-dimensional datasets (N-attribute datasets). We present the underlying abstraction framework and its implementation as an interactive prototype to explore geo-coded entities (location or object) with multi-dimensional attributes. We have submitted a paper from this work as follows:

- Liqun Liu, Romain Vuillemot, Philippe Rivière, Jeremy Boy and Aurélien Tabard. "Generalizing OD-Maps to Explore Multi-Dimensional Geo-Coded Datasets," In *The Cartographic Journal*, P. 26, 2022. (Under review, Link)

Contribution "GroupSet" (Chapter 5): We introduce a novel visualization technique to analyze the changing patterns in time-varying data, such as traffic flow data changing over time. It relies on sets theory and existing set-based visualizations that explore sets intersections. The technique can help users reveal temporal patterns and similarities, such as increases or decreases in traffic flow values during a day. We demonstrate the technique's applicability to traffic flow and report on usability feedback of an interactive prototype implementing the technique. We have published a paper from this work as follows:

- Liqun Liu and Romain Vuillemot. "GROUPSET: A Set-Based Technique to Explore Time-Varying Data," In *EuroVis 2022 - Short Papers*, the Eurographics Association, Roma, Italy, P. 5, June 2022. (Link)

In the final part of the manuscript, **Chapter 6**, we discuss the applications of three contributions (FuzzyCut, Gridify, and GroupSet) beyond traffic-relevant data and how to deploy them in traffic control centers (Figure 1.4) using ControlCenter, using wall-display screens, close to a real workplace setting. We conclude with the open research challenges that remain to address in the future.

CHAPTER 2

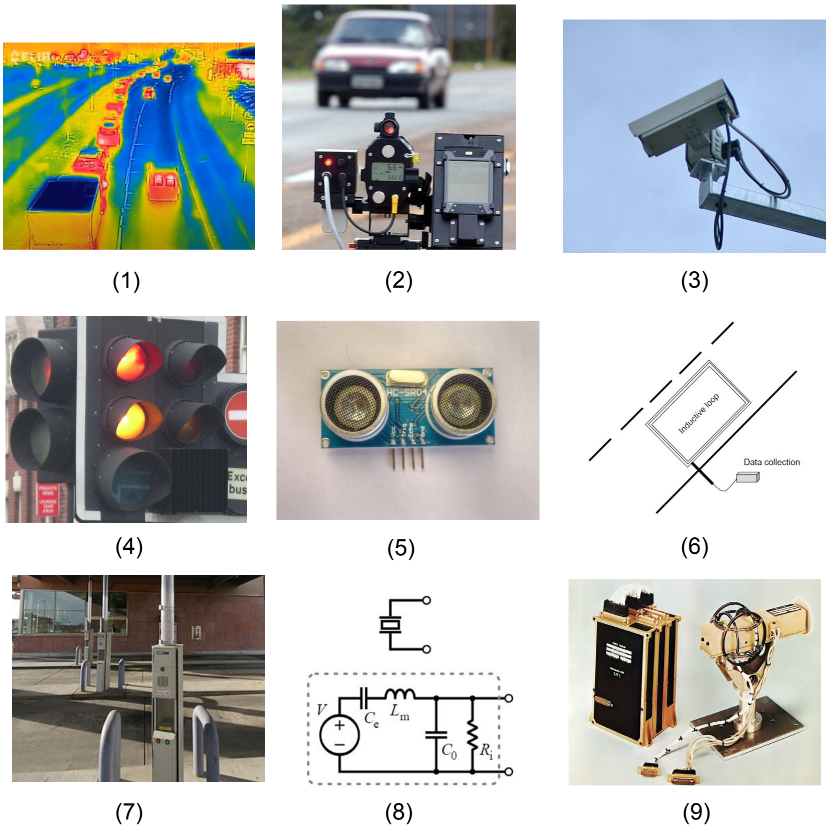
Background and Related Works

Contents

2.1 Traffic Flow Data Sources

Traffic flow is the study of the movement of individual drivers and vehicles between two points and the interactions they make with one another [41]. It aims to develop optimal traffic networks with efficient movement and minimal traffic congestion problems. Traffic flow data can be collected in many ways, and this section reviews the main ones:

- Non-intrusive sensors that reside on roadsides or above pavements;



(1) (2) (3) (4) (5) (6) (7) (8) (9)

Figure 2.1: Examples of non-intrusive and intrusive sensors. (1) Infrared, from a YouTube video [3]. (2) Roadside radar, from Wikipedia [4]. (3) Roadside camera, from Wikipedia [5]. (4) Traffic signal light, from Wikipedia [6]. (5) Ultrasonic sensor, from Wikipedia [7]. (6) Induction loop. (7) Pneumatic road tube, from Wikipedia [8]. (8) PieZoelectric sensor, from Wikipedia [9]. (9) Magnetic sensor, from Wikipedia [10].

- Intrusive sensors that reside inside roads, such as grooves, tunnels under road surfaces, or holes;
- Off-roadway sensors that reside on moving objects to collect moving information of objects, such as taxi speed and cruising distance;

Simulations—mathematical modeling method to reproduce traffic and transportation systems using computer software—can also be used to produce traffic flow data [42].

2.1.1 Non-intrusive Sensors

This section introduces non-intrusive sensors. It includes infrared, roadside radar, roadside camera, traffic signal light, and ultrasonic, as shown in Figure 2.1 (1-5). The advantage of non-intrusive sensors is not to disrupt traffic; however most non-intrusive sensors can be easily disrupted by bad weather. For example, rainy days have an enormous impact on video detection.

Infrared can detect vehicles passing through a specific road segment once in the view of the sensor. The infrared sensor leverages the infrared theory, which emits the light in the infrared spectrum and measures how much light is reflected from the objects. Different temperatures caused by engines or lights beam would reflect different light volumes to infrared sensors. So the infrared sensor can detect the presence, speed, and type of vehicles. Moreover, the advantage of the infrared sensor is that it is easy to install and does not hinder traffic. Nevertheless, it does not perform well in some cases, such as bad weather. It also has a high initial cost, and its accuracy would be decreased if too many vehicles were on road segments. Compared with the induction loop detector, it has lower accuracy, only 95.5% for highways and 92% [43] for road intersections.

Roadside radar resides at a fixed location on the road to detect the speed of vehicles in a concise period of time (a few seconds). Like the infrared sensor, the radar sensor emits microwaves to the environment and then measures the reflected microwaves and their time. It can determine vehicles' presence and motion. Also, it determines the type of vehicles based on their outline using unique algorithms. The advantages of the radar sensor are that it does not disturb traffic during installation, measures the speed of vehicles with high accuracy, and is stable even in bad weather. However, the detection accuracy may be affected by the occlusion of vehicles. In addition, it is not sensitive to low-speed objects, so the accuracy decreases sharply when the traffic volume is low.

Roadside camera detects vehicles entering road intersections based on pixel changes [44]. Initially, the camera was only used to detect the number of vehicles passing through specific road segments. But with the development of computer vision research, the camera is now also used to detect vehicle information, such as ID and speeds. The advantages of the camera are quick installation and traffic not being affected during installation. However, video-based surveillance also has disadvantages (*e.g.,* privacy issues). Also, it has lower efficiency when it rains or snows, affecting the pixels that the video can detect.

Traffic signal light is a piece of widely used equipment for regulating traffic around the world. Proper traffic signal phasing can significantly reduce road pressure, especially during morning and evening traffic peaks. The traffic control centers record the traffic light's status, including how often the traffic light is green, red, or yellow. This data is beneficial for the city administration while planning the road networks.

Ultrasonic sensor emits high-frequency sound waves to the environment and receives the sounds when they bounce back. The ultrasonic sensor can calculate the speed of vehicles based on the duration of the sound wave between the time it is emitted and the time it bounces back. The ultrasonic sensor is quick to install and does not interfere with traffic during installation. It is also cheaper than many other sensors. However, it has the disadvantage of being easily affected by obscured vehicles and weather (*e.g.,* temperature changes and wind noise).

2.1.2 Intrusive Sensors

This section introduces the intrusive sensors. The advantages of intrusive sensors are their high accuracy. However, these sensors are easily affected by poor road conditions. Also, the road repairs greatly disturb the intrusive sensors such as induction loop detectors, magnetic sensors, pneumatic tubes, and piezoelectric sensors, as shown in Figure 2.1 (6-9).

Induction loop detector generates an electromagnetic field with two induction loops installed under each road section. It cannot only determine how many vehicles are passing a section of road but also determine the speed of vehicles at intervals of a few minutes. Its advantages are lower cost and higher accuracy than other roadside sensors. The accuracy of this sensor is 99.3% for highways and 97.9% [43] for road intersections. However, the induction loop detector cannot detect the type of vehicle, and its performance is also greatly affected by bad weather, especially temperature fluctuations.

Pneumatic road tube collects data through the changes in air pressure. It is an axle sensor installed on the road. The air pressure changes when the wheels of the vehicle pass over the hoses (a tube made of rubber and contains air inside to test the air pressure) since it disturbs the air pressure. The disturbance is transmitted to the data center to count the vehicles on a particular road. This sensor is typically used for short-term counts, which is cheap, very easy to install, and consumes little power. However, heavy vehicles easily damage them and they usually have a short useful life.

Piezoelectric sensor is an axle sensor cut into a groove on the road. The piezoelectric sensor collects data by converting mechanical information into electrical information. The sensor is compressed when a vehicle passes over it, which causes the deformation of sensors that generate a voltage signal. The advantages of the sensor are low power consumption and high accuracy. However, it is easily destroyed by damage to the road.

Magnetic sensor [45] determines the disturbance of the earth's magnetism when metallic vehicles are near the sensors. It consists of two devices, the sensor node (SN) and the access point (AP). The sensor node is stuck on the road surface to detect the vehicles, and the access point is installed on the roadside to collect the data from AP. Then the access point forwards the data to the traffic control center. The advantages of magnetic sensors are that they are inexpensive compared to inductive loop detectors, cameras, and radars. They also have a long life and consume little power.

2.1.3 Off-roadway Sensors

This section introduces sensors attached to objects (*e.g.,* vehicles and bicycles), as shown in Figure 2.2. They usually record activities of objects over a long period, such as the trajectories generated by vehicles. The data collected from off-roadway sensors reflect how objects move, which helps analyze travel behaviors in a city. We introduce them as follows:

GPS (Global Positioning System) is a satellite-based navigation system developed by the United States Space Force in 1978 [46]. GPS receivers are capable of computing a four-dimensional space-time position of four satellites. In this case, each satellite calculates a position and time. And then, they transmit the correctly recorded data to the GPS receiver. Every vehicle must have a GPS installed, providing vehicle navigation service and recording much space-time information about vehicles. However, it still has disadvantages because it is greatly affected by obstacles such as tunnels, mountains, and trees.

Mobile phone is similar to GPS navigation systems. The difference is that the cell phone replaces the GPS receiver, and the telephone antenna station replaces satellites [47]. The position data of the cell phone is entirely accurate in the cities, but it is not very reliable in the suburbs and some areas far from the cities. Another disadvantage of the cell phone is the

Mobile phone GPS Automotive camera Automotive radar Smart card

Figure 2.2: Off-roadway sensors reside on the moving objects. As shown above, pictures are taken or generated ourselves. They can record the trajectories and status of objects.

concern about privacy since more people are concerned that the information will be recorded without authorization.

Automotive camera is similar to the roadside camera but installed in vehicles, aiming to record the self-driving car's environment. With the development of the self-driving car, the camera has been used by many car manufacturers because the cameras are cheaper than most other detection devices [48].

Automotive radar is similar to roadside radar, which also emits radio waves and receives them when they bounce back. The difference is that automotive radar is to detect the surroundings of one's vehicle, including objects' speed, distance, and direction, which contains two types: short-range radar and long-range radar. The short-range radar [49] is for close-range applications, such as blind-spot detection and parking aids. Long-range radar is to measure the distance and speed of objects.

Smart card is a physical, electronic authorization device used to control access to a resource. The transport domain uses it for the transit fare and park fee payment. It records the travelers' information, including their routes, time, and fees. The recorded data from the smart card can extract trajectory information to help analyze the humans' patterns and spatio-temporal relations.

2.1.4 Traffic Simulation

This section introduces the simulated data source. It enables better planning, design, and operation of the traffic systems. Traffic flow simulation is essential in traffic research because it can develop complex models to estimate and predict traffic status. It also can be used in other research, such as understanding travel patterns and producing intuitive visualization. Many years ago, Hoogendoorn *et al.* [50] have summarized traffic flow modeling methods based on the level-of-detail classification: microscopic (*e.g.,* Car-following models), mesoscopic (*e.g.,* Cellular automata model), and macroscopic (*e.g.,* Monte Carlo method). Inspired by these classifications of methods, this section introduces several typical traffic flow simulation methods and three popular simulation software.

2.1.4.1 Approaches

The traffic simulation approach is the mathematical theory of reproducing traffic data. It includes probability and statistics, differential equations, and numerical methods. This section introduces the three most common approaches.

Car-following model is the typical microscopic simulation method, which regards the vehicle as the unit to construct the model considering the position and velocity of every vehicle. As a result, the car-following model can predict the vehicle's behavior by analyzing the relations between the single vehicle's properties and the stream of traffic flows [51]. Besides, Liao *et al.* [52] develop a car-following model considering the drivers' habits to estimate the safety rate.

Cellular automata model is a discrete model of computation based on automata theory, which enables dynamical evaluation and description of the traffic states. It can be used in traffic simulation since space is a discrete regular cell with a finite number of possible states. The states depend on the model of traffic phenomena, and they update synchronously during the discrete timestamps [53]. The advantage of the discrete space of the cellular automata model is that it allows for faster computation than the continuous model.

Monte Carlo method repeatedly generates random samplings to obtain the numerical results [54]. In the context of traffic simulation, it can generate traffic data based on probability distribution. For example, Jeon *et al.* combine the Monte Carlo method [55] with the time-series forecasting method to predict traffic flows.

2.1.4.2 Software

Various simulation softwares can generate simulated traffic data. Traffic experts use softwares to understand traffic patterns and predict traffic flows. Compared to the real data, the data generated from simulation softwares are easily implemented in visualization techniques since they do not have the data quality problem.

CarSim is a software for simulating the state of vehicles in a given environment (*e.g.,* driving environment). This software was developed by an American company, Mechanical Simulation Corporation. The original technology came from the Transportation Research Institute at the University of Michigan. It simulates the distance to the vehicle ahead, the friction on the road, and the state of the traffic lights. It can also simulate the corresponding driving behavior, including braking, shifting, and clutching [56]. By using it, traffic experts can improve driving control, test the condition of vehicles, and estimate the mathematical model.

VISSIM is a microscopic traffic flow simulation software developed in 1992 by PTV Planung Transport Verkehr AG in Karlsruhe, Germany [57]. Today it becomes the world's most popular microscopic traffic flow simulation software. This software simulates the objects moving on the road individually thanks to microscopic simulation. This indicates that the data collected by this software is quite detailed, so we know the exact speed of the vehicles in each second.

Visum is a complex macroscopic traffic simulation software also developed by PTV Planung Transport Verkehr AG in Karlsruhe. Compared to VISSIM, Visum simulates traffic flows in a macroscopic way. The most significant difference is that Visum estimates the movement of objects on roads with the average traffic flow and density. It benefits transportation planning, travel demand modeling, and network data management. Visum integrates all relevant modes (*e.g.,* vehicles, bicycles, pedestrians, buses, and trains) into a unified network.

2.1.5 Data Availability and Examples of Traffic Datasets

There is a growing availability of open datasets in traffic domains coming from sensors introduced in the previous section. However, this section only focuses on several typical road traffic datasets used in this manuscript. Depending on the characters of these datasets, we will design visualization techniques to expose valuable traffic information to help traffic experts know traffic from multiple perspectives. These datasets are as follows:

- **CRITER datasets (Traffic density and event datasets) in Lyon, France**. We utilized CRITER datasets from the government website of Lyon, which is a freely accessible API [1] that can extract relevant traffic situation data in Lyon. We extracted the traffic density datasets named ”*Etat du trafic de la Métropole de Lyon - disponibilités temps réel*” and event datasets named ”*Alertes trafic du réseau des Transports en Commun Lyonnais*”. The number of road segments is more than 1000, and the data collection frequency of the traffic density information is once per minute. This API can also collect the event log data, including *warnings* and *roadwork*. This manuscript extracted the traffic density and event data in 2018. The amount of data size is more than 10 million. We use the traffic density and event datasets in **Chapter 4** (to explore the spatial relations of traffic density distribution in Lyon), **Chapter 5** (to discover how traffic density changes over time), and **Chapter 6** (to discuss how to deploy the multiple visualization techniques in traffic control centers to augment its monitoring ability).

- **Taxi taking passengers trajectory datasets in Wuhan, China**. Taxis are everywhere in cities, and to a certain extent, taxi trajectories reflect human mobility [58]. This manuscript used the taxi trajectory data from Wuhan, China, which includes more than 7271 taxis and the total number of records is more than 220 million during two months (September and October) in 2013. The trajectory data was collected by the GPS equipment installed in the taxis, and it records the taxis’ travel data 3 to 4 times per minute. We use the taxi trajectory datasets in **Chapter 3** (to categorize the taxi speeds) and **Chapter 4** (to explore the spatial relations of taxi trajectories).

- **Webcam datasets (Video datasets)**. Webcam is a video camera that feeds or streams an image or video in real-time and through a computer network [59]. It can monitor the traffic situation (*e.g.,* traffic accidents) and explore the vehicle’s behaviors by designing a visual system. In this manuscript, we use the webcam data collected with open access, presenting real-time traffic video in Lyon, France [2]. These videos were captured by the cameras installed on the roadside. We extracted 15 cameras [3] and their characteristics, such as the coordinates, the road’s name, and the road’s *ID*. In this manuscript, we use the webcam datasets in **Chapter 6** as the views in the dashboards of traffic control centers.

- **Simulated traffic density datasets**. We generated a simple random traffic density datasets [4] using a bimodal normal distribution to simulate the morning and evening peaks. In this

[1] https://data.grandlyon.com/jeux-de-donnees?q=trafic

[2] https://worldcam.eu/webcams/europe/france/51-lyon-panoramic-view

[3] https://observablehq.com/d/07cbd1f56725f0cf

[4] https://observablehq.com/@pierreleripoll/simulate-trafic-data

manuscript, we did not utilize the mathematical theory mentioned in the previous section since we were only interested in generating realistic data rather than accurate ones. Simulated traffic density datasets play an important role in visualization design because the real traffic density values include some extreme values and some items have null values, affecting the test of the novel visualization techniques. In this manuscript, we use the simulated traffic density datasets in **Chapter 6** to test the availability of the traffic map.

- **Transit datasets**. We used a dataset including 45,520 trips of Paris every hour on a given day. The trips start from three distinct locations (origins), and destinations are all reachable areas surrounding the origin for a given period (e.g., five minutes) by different commuting ways (walk or public transport). We also used various dimensional attributes, such as CO_2 emissions. This dataset comes from the Navitia API [5] and we use the transit dataset in **Chapter 4** to discover the accessibility of transport in urban.

- **Road information (Road map)**. We utilized the road coordinate information data from both Lyon and Wuhan. They offer road segment information, such as coordinate and road *ID*. The road *ID* can be regarded as the connection between maps and other data sources (*e.g.,* traffic density data and taxi trajectory data), therefore, one can know where the objects are on a map (*e.g.,* where a webcam is located). We use the road map in **Chapter 6** to locate the objects in a geographical space.

2.2 Data Characterization

This section introduces the characterization of the typical road traffic data and derived data. The data characterization is *"a summarization of the general characteristics or features of a target class of data"* [60]. It reflects the data abstraction in different features, such as the coordinates of the vehicle trajectories.

2.2.1 Data Types

Generally, data types are strongly influential to visualization types. Shneiderman categorized the data into seven taxonomy: 1-dimensional, 2-dimensional, 3-dimensional, temporal, multi-dimensional, tree and network data [61]. However, we do not address all the data types in this manuscript. We focus on 1-, 2-dimensional, and temporal data. They are as follows:

- **1-dimensional data (1D data)** refers to the univariate data that only contains a single characteristic or attribute, such as the taxis' speed and traffic density values. Analysts generally use 1-dimensional data classification when they want to divide the data into segments, such as dividing data by year or quarter. In this manuscript, we address the 1-dimensional data in Chapter 3 by creating a categorization technique for 1-dimensional quantitative values (*e.g.,*to split traffic speed data into *fast* or *slow* categories).

[5] https://www.navitia.io/

- **2-dimensional data (2D data)** exists in a two-dimensional coordinate space represented by X (longitude) and Y (latitude) coordinates. In the road traffic domain, a typical 2-dimensional data is the movement data. It can describe how objects move and their parameters, such as speed, direction, and acceleration, as shown in Figure 2.3, the taxi taking passenger trajectories from Wuhan, China. We define movement data P containing multiple attributes where a trajectory $p \in P$ is the ordered point containing temporal and spatial information as $p =< p_1, ..., p_l >$. Each trajectory point $p_k : 1 <= k <= l$ can be defined as $p_k = [s^n, t, a_1, ..., a_m]$ where s refers to the spatial coordinate of the point p_k (If the spatial information contains only latitude and longitude, $n = 2$. If it also contains altitude information, $n = 3$). $t \in T$ refers to the time window of the point p_k, and $a_i \in A$ refers to the attribute, where $a_i : 1 <= i <= m$. This definition not only shows the trajectories' spatio-temporal relationships but also represents each point's attributes that can be used to explore implicit patterns (*e.g.,* moving direction and speed). We address the 2-dimensional data in Chapter 4 by creating a nested visualization technique, which explores spatial patterns and information in cities.
- **Temporal data** refers to data that contains variables changing over time. The traffic data collected by point-based sensors (including both intrusive sensors and non-intrusive sensors that sit on the road or roadside, introduced in Section 2.1.1 and 2.1.2) can be viewed as temporal data for each recording point. Generally, sensors collect position-based data with a fixed frequency (*e.g.,* every 1-4 minutes) and the timestamps are multi-dimensional, such as *Day*, *Time* of the day, *Hour*, *Minute*, or *Second*. Traffic managers utilize the temporal data to monitor what happened on specific road segments over time and the monitored results help urban planners optimize the routes. We address the temporal data in Chapter 5 with an interactive visualization technique to analyze how the temporal data change over time (*e.g.,* traffic density data).

These three data types provide temporal or spatial information. Nevertheless, combining all the spatial and temporal information to synthetically analyze road traffic situations can bring more valuable information to traffic experts. Thus, we also address the heterogeneous data in Chapter 6 by deploying the multiple visualization techniques in traffic control centers using different traffic data sources, such as the webcam, and road events data.

2.2.2 Derived Data

We have introduced position-based sensors (non-intrusive and intrusive sensors) in Section 2.1. The position-based sensors collect raw road section data that describe the basic information in a particular road segment. However, analysts usually need more attributes contained in the data to explore more valuable and interesting patterns. This section introduces how to derive the new attributes based on the single-vehicle information of road section raw data. Road section raw data reveal 1) the number of vehicles passing through the specific road segments and 2) the time when vehicles pass them. Although the road section raw data only tests the single-vehicle data, it can still estimate the relevant traffic parameters by aggregating the microscopic single-vehicle data with the average values in particular time intervals Δt if the recorded number of vehicles is ΔN.

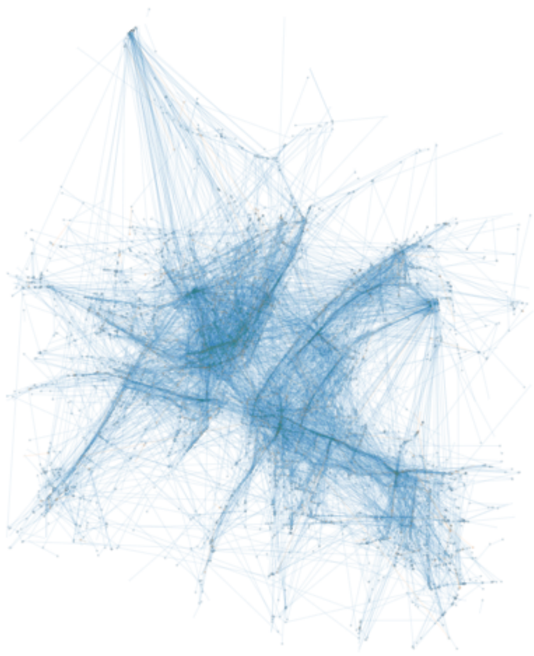


Figure 2.3: A sample of movement data (the taxi taking passenger trajectories in Wuhan, China) is a typical 2D data that we focus it in Chapter 4. Each blue segment encodes *OD data* of taxis: *O* refers to the origin when a taxi picks up the passengers; *D* refers to the destination when a taxi drops off the passengers.

For single-vehicle detection, t_α^0 refers to the front of vehicle α passing through the detection position, and the t_α^1 refers to the end of vehicle α passing through the detection position. Thus, we define **traffic flow** as the number of vehicles ΔN passing through a fixed position with the time interval Δt, shown as Equation 2.1:

$$Q(x,t) = \frac{\Delta N}{\Delta t} \tag{2.1}$$

where the x refers to sensor's position. Thus **microscopic flow** (it presents the single vehicle-driver unit's properties, *e.g.,* the position and the velocity, used to estimate the traffic density) shows as Equation 2.2:

$$q_\alpha = \frac{1}{\Delta t_\alpha} \tag{2.2}$$

Occupancy is the percentage of aggregation interval that the detection zone is occupied by a vehicle, shown as Equation 2.3:

$$O(x,t) = \frac{1}{\Delta t} \sum_{\alpha=\alpha_0}^{\alpha_0+\Delta N-1} (t_\alpha^1 - t_\alpha^0) \tag{2.3}$$

Arithmetic mean speed refers to the average vehicle speed of ΔN vehicles passing through the detection point during the aggregation time interval, shown as Equation 2.4:

$$V(x,t) = \langle v_\alpha \rangle = \frac{1}{\Delta N} \sum_{\alpha=\alpha_0}^{\alpha+\Delta N-1} v_\alpha \tag{2.4}$$

Speed variance represents the spread of speed values during the aggregation interval, which is given by the standard deviation σ^2, shown as Equation 2.5:

$$Var(v) = \sigma_v^2(x,t) = \langle v^2 \rangle - \langle v_\alpha^2 \rangle \tag{2.5}$$

The previous quantities are measured directly by the single-vehicle parameter. However, we have to estimate quantities with assumptions. **Traffic density** can be estimated if we know the microscopic quantities flow Q and the arithmetic means speed V. In this case, the traffic density ρ is the average value during the fixed time interval. Thus, we can calculate it as Equation 2.6:

$$\rho(x,t) = \frac{Q(x,t)}{V(x,t)} \tag{2.6}$$

Space mean speed $\langle V(t) \rangle$ represents the arithmetic mean speed of all vehicles in a given time t on a specific road segment that is:

$$\langle V(t) \rangle = \frac{1}{n(t)} \sum_{\alpha=1}^{n(t)} v_\alpha(t) \tag{2.7}$$

New attributes above describe the traffic from more perspectives, such as the average speed, occupancy, or traffic density. We utilize the attributes in Chapter 4 that we design a nest-based visualization technique to analyze the relations among these derived attributes, such as the relation between traffic densities and the weather.

2.3 Users and Tasks

This section introduces the users who can benefit from our visualization techniques and their tasks. Regarding users, we summarize four types: citizens, transport planners, urban planners, and traffic control centers. We group tasks into three categories that are monitoring the traffic, pattern discovery and clustering, and situation-aware exploration and prediction.

2.3.1 Users

This section introduces the relevant users in the road traffic domain based on our reviewed literature and existing commercial tools (*e.g.,* google maps). We classify users into two main categories that are experts and non-experts defined as follows:

- **Experts** work in the traffic domain to improve the traffic service with prior knowledge of human mobility and traffic density distribution, including **traffic operators in traffic control centers**, **transport planners**, and **city planners**.

- **Non-experts** are the people who travel on the road (*e.g.,* drivers and pedestrians) but do not have prior knowledge of traffic. Generally, what they need from the visualization techniques are navigation and road planning, to avoid losing themselves in cities and find optimal paths. In this manuscript, we refer to non-experts as **citizens**.

Transport planners play a crucial role in optimizing transit lines to reduce traffic congestion. Public transport usually is not in the cost-benefit equilibrium (balance between the road capacity and the number of vehicles on the road) since some lines of bus or metro are jam-packed while others are taking fewer travelers. Chu *et al.* [62] help transport planners better know the taxi traveling pattern by developing a semantic analytical system to explore the taxi traveling patterns from the massive taxi trajectory data. In addition, Oliveira *et al.* [63] design Bike-sharing systems (BSSs), helping traffic planners understand the commuters through biking travel. Also, Miranda *et al.* [64] contribute Urban Pulse, a visual system for capturing spatio-temporal activities through public transport, enabling traffic planners to observe how citizens commute during the different periods. By doing so, they can know places where people have many activities during a particular time but some places with fewer activities. In the manuscript, we introduce two visualization techniques dedicated to transport planners to observe the spatial relations (Chapter 4) and temporal relations (Chapter 5) of traffic data.

Citizens (*e.g.,* drivers and pedestrians) generally want to find a suitable route using maps. The maps support citizen with route planning to avoid losing in the complex route networks. Researchers have designed systems to assist users in route recommendations. Wang *et al.* [65] design the TaxiRec, a route recommendation system that assists drivers in improving the ability to find passengers using a trajectory clustering method. Similarly, Lu *et al.* [66] design an interactive visual system with a filtering trajectory feature to analyze the taxi trajectory selection behaviors, which is helpful for route recommendation. The manuscript does not design visualization techniques for citizens since they need a more simple visual representation. We will expand the visualization techniques to assist citizens in future work.

Urban planners aim to build and optimize the infrastructure of cities, such as selecting the places to build gas stations. Good planning of city infrastructure can improve the experience of citizens and raise the efficiency in some services. For instance, a gas station having a suitable place can provide drivers a good experience since they may not drive a long time and long distance to find a gas station. In order to find suitable positions for the infrastructure of cities, knowing the urban space is a critical skill for urban planners. For the purpose of helping urban planners understand the urban spaces, researchers have designed interactive visualization techniques based on the big massive road traffic data. Shen *et al.* [67] design StreetVizor, a visual exploration system to assist users in urban planning and environmental auditing by facilitating machine learning to detect the street view patterns. Ferreira *et al.* [68] propose a visual system named Urbane to simulate the urban parameter, which enables urban planners to interact with it by adding or removing buildings to see what happens if they do so. Apart from understanding urban spaces, urban planners need to optimize the infrastructure, such as selecting billboard places. In this issue, Liu *et al.* [69] design SmartAdP, a visual technique for selecting the billboard locations based on the large-scale taxi trajectories. SmartAdP can help planners find a suitable solution with the selection of billboard locations and assist them in intuitively comparing the solutions. The manuscript helps urban planners explore how traffic

Tasks	Sub-tasks	Related works	Datasets
Traffic monitor	Monitor systems	[72] [73] [74] [75] [76] [77] [20] [23]	Taxi trajectory, traffic event, traffic flow
	Monitor Approaches	[78] [79] [80] [81] [82] [83] [84]	Video, taxi trajectory
Pattern discovery	Clustering and aggregation	[85] [86] [87] [88] [89] [90]	Taxi trajectories, vehicle trajectories
	Topology visualization	[91] [92]	Taxi trajectories
	Density map	[93] [94] [18] [76] [95]	Taxi trajectories
	Space-time cube	[13] [96]	Vehicle trajectories
Situation-aware exploration	Traveling purpose	[97] [98] [99] [11]	Taxi trajectories, events
	Correlations with events	[100] [62] [69]	Taxi trajectories, events

Table 2.1: Summary of tasks in road traffic data visualization. It introduces three tasks and the corresponding related works. The visualization techniques designed in the manuscript are based on the three tasks.

densities change over time in Chapter 5, which is an essential factor while optimizing the city infrastructure positions.

Traffic control centers serve as the controlling center of the urban area, including the main street, road intersections, and traffic lights. Such centers are indoor physical facilities with restricted access as they play a key role in managing traffic but also roadworks and road message boards (*i.e.* to announce congestion or closed road segments). Each operator in the room has a workstation composed of several regular screens on which they can access useful information, including a detailed version of the traffic map and a specific CCTV camera (which they can also control). The traffic control center is essential for the administration of the information center, which generally gives traffic operators an overview of the roads using the wall-displayed dashboard. One primary mission is to monitor traffic situations and road incidents, such as traffic accidents. Traffic monitoring can improve road traffic by maximizing road capacity, minimizing the impact of incidents, and assisting in emergency services. We discuss how to improve the traffic control centers and deploy multiple visualization techniques in dashboards of control centers in Chapter 6.

2.3.2 Tasks

Tasks are human activities that analyze the dataset regarding specific and similar questions. Our approach is aligned with the task analysis process, which has been generally documented in the visualization community [70]. As we work on a specific application domain, we rely upon specific tasks in road traffic data composed of monitoring the traffic, pattern discovery and clustering, and situation-aware exploration and prediction [71]. We list them, along with related works in Table 2.1.

2.3.2.1 Traffic Monitoring

Traffic monitoring helps traffic control centers improve their network operations and make better decisions, thus improving the service for travelers and commuters (we introduce how to improve traffic control centers and how to deploy multiple visualization techniques in control centers in Chapter 6, thus improving the centers' ability to monitor road traffic). This section introduces the tasks related to traffic situation monitoring, such as monitoring travel time and speed. And then, this section introduces monitoring systems combining algorithms to monitor travel and driving behaviors. We introduce two aspects as follows:

1. Traffic situation monitoring

Most monitoring systems focus on traffic flows because they can directly reflect the traffic *situations* (*e.g.,* traffic congestion or free flow). However, developing a system for interactively monitoring traffic flows is difficult since it requires dealing with real-time and large historical data. Cao *et al.* [23] define three challenges in real-time monitoring of traffic situations, which are **adaptivity**, **interpretability** and **interactivity**.

The anomaly detection with large data is not easy in the visual system. The *adaptivity* challenge is that the calculating time must be considered while designing the detecting algorithm to keep both the accuracy and calculating speed. In this research field, Wang *et al.* [72] use a simple method to estimate the traffic jam by calculating the speed of vehicles. Hilton *et al.* [73] introduce a heat map visual system to reduce the calculation time while detecting the traffic safety situations.

Avoiding the clutter and overlap visualization while dealing with a large amount of data enables improvements in the *interpretability*. The actual traffic situation might not be observed, and experts have difficulty finding interesting patterns. Hence, designing an effective visualization system is very helpful. In order to better monitor the traffic situations and address the visual problems caused by a large amount of data, Andrienko *et al.* [75] investigate the aggregation methods in movement data. Furthermore, to improve the visualization's interpretability, Scheepens *et al.* [76] introduce a particle system to help traffic experts explore the interesting trajectories, combining the density map with moving particles to display the additional trajectory information.

The *interactivity* challenge is the designing of interactive visual systems with online detection. Researchers have been addressing this problem by developing a specific database to avoid time-costuming data structures [77]. Wang *et al.* [20] create a road-based query model constructed on *TripHash*. Based on this data structure, they design a dynamic spatial-temporal query of trajectories system to help traffic experts evaluate and monitor the traffic situations. Furthermore, researchers address the problem by designing suitable algorithms to make the anomaly detection cost less time and improve its accuracy. Cao *et al.* [23] design an algorithm for detecting the anomalies with streaming data and an interactive traffic monitoring system emerging human guidance.

2. Travel and driving behaviors monitoring

A most useful equipment for monitoring road traffic is the video feed. It not only benefits traffic experts for monitoring traffic flows but also for monitoring more complex and implicit information, such as the travel and driving behaviors. To do this, current surveillance system or traffic monitoring system introduced the computer vision approach. Computer vision surveillance system is an essential part of intelligent transportation systems (ITS). Tian *et al.* [78] summarize the related surveillance systems for monitoring and managing traffic flows. It shows that the computer vision surveillance systems first extract the attributes of vehicles and then understand the behaviors. Hence, the extraction of vehicles' attributes is the foundation for understanding the travel and driving behaviors, which contains three aspects: vehicle detection, vehicle tracking, and vehicle recognition.

Accurate and robust vehicle detection is vehicle tracking and recognition's first step. In computer vision, vehicles contain different pixel characteristics to detect vehicles. In general, detection algorithms contain two methods: appearance-based and motion-based. The appearance-based method detects the vehicle's shape, color, and texture. Conversely, the motion-based method determines the vehicle based on the moving characteristic, such as speed and acceleration. The most interesting feature of detecting vehicles using the motion-based method is to extract the dynamic vehicles from the static background. The detection algorithm calculates the difference between the front object pixels and the background pixels [79]. Several basic methods used in vehicle detection contain wavelets [80], Kalman filter [81], and Gaussian pixel distribution [82]. However, two general problems raised are vehicle shadow and vehicle occlusion. The vehicle shadow affects the detecting results since it changes the colors and texture of actual vehicles. The vehicle occlusion caused by the high traffic densities is another common problem in detecting vehicles.

Besides computer vision, researchers also use the taxi trajectory data to monitor travel behaviors. Taxis are the leading transport equipment in cities, and most taxis work 24 hours daily. Hence, taxi trajectory offers vital information to estimate the traffic matrix or detect trajectory anomalies. Wang *et al.* [83] use GPS data as the data source to detect anomalous travel behaviors. They propose an algorithm to calculate the similarity of trajectories using edit distance and cluster the trajectories into groups with the hierarchical method. Kuang *et al.* [84] combine wavelet transform and PCA (principal component analysis) to detect the high deviated traffic flows and propagate the sub-region where the anomaly behaviors happen.

The computer vision approach introduced in the surveillance system brings more efficient traffic movement and reduces human supervision. We have such video datasets as introduced in Section 2.1.5 — the webcam datasets. We utilize the datasets in Chapter 6 to discuss how to deploy them and other visualization techniques in traffic control centers. However, it also raises privacy concerns since the webcam records vehicle trajectories. Although there are methods for addressing the problem, *e.g.,* video blurring [101], they may not always be available.

2.3.2.2 Pattern Discovery and Clustering

The section introduces task **pattern discovery and clustering**, which allows users to know how humans or other objects move in the network systems. Travel patterns reflect where people prefer to travel and when they usually do activities. Travel patterns contain three components: trip distance distribution, number of visited locations, and radius of gyration. Although these components of *individual* human mobility is challenging to forecast, travelers' overall patterns are highly predictable [70]. The manuscript focuses on this task in Chapter 3 to analyze the categorization of traffic flows, and in Chapter 5 to discover the traffic density temporal changing patterns.

An important method for discovering human mobility is clustering the trajectory data. The method can emphasize the traveling behaviors and discover implicit patterns from the massive random trajectories. Wu *et al.* [86] propose the TelCoVis by facilitating the bi-clustering techniques that allow users to explore the co-occurrence behaviors in the two regions, such as many humans traveling from one region to another at the same period. Andrienko *et al.* [87] analyze the group movement behaviors by calculating the central trajectories of the group and transforming the group members into the group space created by the group movement. Kalamaras *et al.* [88] develop multiple functions traffic visual platform system, which contains road clustering, anomaly detection, and traffic prediction by combining the relevant algorithms in this system. Von *et al.* [89] propose a graph-based method using spatial and temporal simplifications with the Twitter and mobile phone datasets. The visualization system introduces a spatial graph clustering algorithm, allowing users to change the parameters to optimize the graph. Yao *et al.* [90] propose the spatio-temporal clustering method to explore mobility by creating a spatial and temporal similarity of measurements, which not only distinguishes the movement distribution but also discovers significant spatio-temporal trends.

Topology network is another alternative for discovering group behavior. Luo *et al.* [91] propose a new algorithm based on the local coherence of the sparse field (LCSF) algorithm to calculate the separation behaviors in the irregular and sparse topology network. They design an interactive visual system to dynamically divide the sub-regions to reveal human patterns based on cell phone data. Otten *et al.* [92] expand the topology method to allow users to discover human mobility with long-term traces. It can reflect the relationships of objects and their relations with spatial information.

A density map is an aggregation method for visualizing geographic information based on the kernel density estimation [93], which is an alternative for discovering human mobility. The density map aggregates the subsets of trajectories to help users explore risk analysis and anomaly events. Sheepens *et al.* [94] use density maps to visualize multiple attribute trajectories, and it combines with a widget to allow users to interactively define subsets to explore mobility. After that, they improve the computation [18] of the density field and add a varied radius to enforce the expression. Moreover, they combine a partial system with density maps [76] to solve the trajectory cluttering problems. It emphasizes traffic flow behaviors by designing a widget to help users dynamically select and filter directions and additional information on traffic flow. Cristie *et al.* [95] create CityHeat using Unity3D and Cellular Automata(CA), which helps users observe the traffic heat distribution and thus discover the human mobility through the distribution of vehicles detected by the emission of engines.

Apart from the introduced methods in the previous paragraphs, there are still other visualization techniques that can assist users in realizing the task pattern discovery and clustering, such as the space-time cube [13] for visualizing the trajectories and traffic flow to describe how they change over time and hence understanding the objects' mobility.

2.3.2.3 Situation-Aware Exploration and Prediction

This section introduces the task **situation-aware exploration and prediction** and relevant visualization techniques. We address the task in Chapter 4, which aims at exploring the spatial relations of geo-entities (*e.g.,* moving taxis or humans). The task is essential for helping traffic planners and urban planners optimize the traffic and support a better convenient lifestyle for citizens. It not only focuses on the exploration of mobility but also on the relations between traffic and other domains' knowledge, such as the relations between traffic densities and billboard locations.

Human movements usually relate to a specific purpose, such as working and shopping. Studying the relationship between human movement and activities is the foundation of traffic and transportation planning. Researchers in the traffic domain have studied this issue for many years. In the visualization community, researchers utilize visualization techniques to help traffic managers realize the task. Zeng *et al.* [97] explore the relation between human movement and activities using human mobility data and the POI data (points-of-interest) in Singapore. They design a visual system for traffic experts and conduct case studies to interview traffic experts about the finds of mobility-interest relations. Al-dohuki *et al.* [98] propose Semantic-Traj to help domain and public users understand what happened in massive taxi trajectories by combining taxi trajectories with the street or POI (Point of interest) information. It offers an intuitive and efficient visual presentation for both domain and public users to understand human mobility and its relations with potential activities. Zhao *et al.* use public smart card data to analyze how a passenger differs from or connects with other passengers. The analysis explores the mobility correlations of passengers' interest in group- and individual-based ways [99]. Ferreira *et al.* [11] focus on the OD (Origin-Destination) data. They design a visual system that allows users to manipulate the OD data to compare different regions over time. The most contribution of the system is the comparing query characteristics, which allows users to select the regions and hierarchical period to compare their temporal relations and understand the OD mobility.

Apart from the relation between mobility and the interest of purpose, traffic experts have also studied other topics related to relations between traffic events and trajectories. Sagl *et al.* [100] utilize the mobile phone datasets to analyze human mobility and conduct a case study to explore the relationship between mobility and soccer matches. Chu *et al.* [62] create a semantic transformation visual system to help traffic planners, administration, and travelers explore travel patterns. It makes the connection between the trajectories' coordinates and the road names. By doing so, a topic document related to trajectories is built to help users understand the implicit domain knowledge. Liu *et al.* [69] design SmartAdP to help advertising planners optimize the placements of billboards by combining the data mining method and visual techniques. SmartAdP is an interactive system for finding the solutions for selecting the placement of billboards and comparing these solutions with intuitive ways to help planners make decisions.

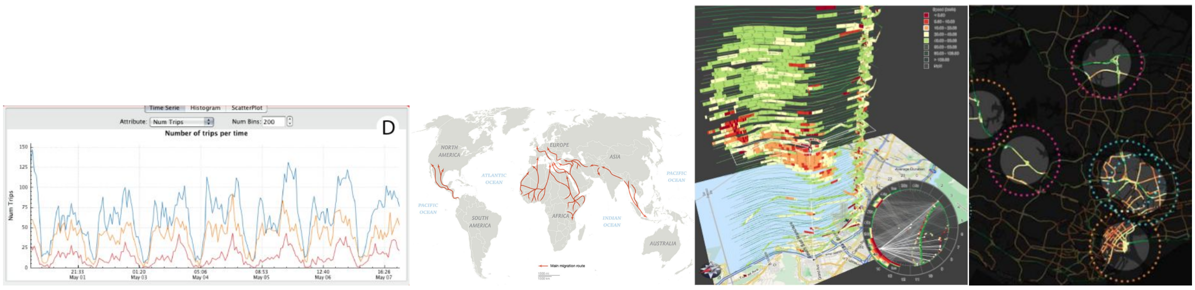
The section introduced three tasks related to the road traffic domain. With the development of visualization techniques, more tasks could be addressed, such as *route planning and recommendation*. Also, tasks mentioned in the section not only concentrate on the traffic domain but also on the other domains, such as exploring the moving pattern (*task:* pattern discovery and clustering) of soccer players (we introduce it in Chapter 6).

Visualization types	Visualization methods	Related works	Datasets
Temporal Visualization	Linear Layout	[104] [15] [17] [87]	Trajectory data, Passenger journey data
	Branching Layout	[105] [106] [107] [108] [16]	Taxi trajectory, Metro passenger data, Metro schedule data
	Circular Layout	[109] [103] [11] [25] [110] [111]	Taxis trajectory, Urban temperature data
Spatial Visualization	Point-based Design	[16] [20] [21] [112]	Metro schedule, Taxi trajectory data
	Line-based Design	[22] [76] [15] [17] [87]	County-to-county migration data, Moving object data, Microscopic traffic trajectory data, Human movement data
	Region-based	[25] [113] [23] [24] [114] [19]	Urban temperature data, Taxi trajectory data, metro passenger RFID card data
Spatio-temporal Visualization	Space-Time Cube	[13] [96] [115] [116]	Trajectory data, Human mobility data
Event Visualization		[26] [25] [27] [107] [117] [14] [118]	Urban temperature data, Road accidents data, Traffic speed data, Air pollution data, Taxi trajectory data, Traffic trajectory data

Table 2.2: Summary of visualization types, methods, and corresponding datasets used in this manuscript, as well as related work. Visualization types and their methods guide our visualization techniques design in the manuscript.

2.4 Visualization Types

The section introduces data visualization techniques while exploring information from massive and complex road traffic data. Based on different needs and tasks [102] in traffic domain, researchers in the visualization community classify visualization techniques and visual systems into four groups: temporal visualization [103], spatial visualization [25], spatio-temporal visualization [27] and event visualization [71]. Figure 2.4 shows the basic visualization types of road traffic data, and Table 2.2 shows the related works corresponding to visualization types. Additionally, the manuscript involves deploying traffic control centers (multiple wall-display visualizations for monitoring traffic situations). The deployment may employ multiple coordinated views (MCV) that we review in this section to combine different views or visualizations in a dashboard of traffic control centers.



(a) (b) (c) (d)

Figure 2.4: The manuscript reviews four visualization types that focus on temporal, spatial, spatio-temporal, and event data. (a) shows the temporal data visualization with a line chart, from the paper [11]; (b) shows a 2D map to display the spatial information, from the paper [12]; (c) shows the spatio-temporal data visualization, from the paper [13]; (d) shows the event-based visualization, from the paper [14].

2.4.1 Temporal Visualization

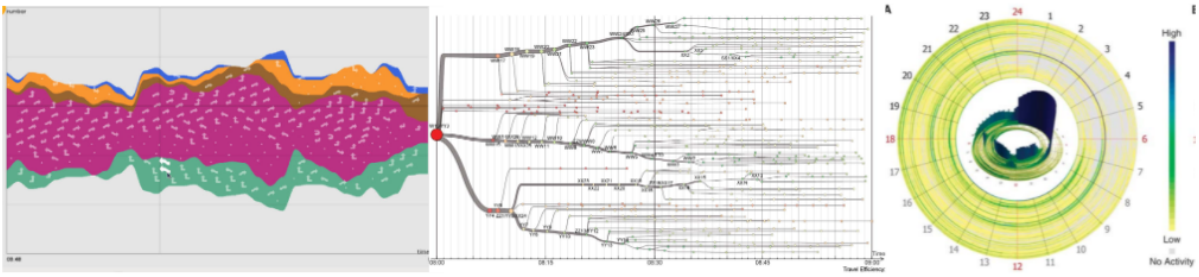
As introduced by Shneiderman, temporal data is one basic data type [61], representing states changing over time, such as traffic flows in different timestamps. The visual representation of temporal data in the traffic domain contains three types: linear layout, circular layout, and branch layout. We introduce corresponding visualizations in this section.

2.4.1.1 Linear Layout

The basic visual representation of temporal data is the linear layout (*e.g.,* line chart) that is among the oldest representation and conveys the structure of raw data for visual inspection, such as slop chart [119] displaying only the first and last element to compare. ThemeRiver is another typical linear time method to visualize the changes over time, as shown in Figure 2.5 (a). The colors of ThemeRiver refer to the individual themes, which are used in many visualization techniques by combining them with other visual representations. In the road traffic data visualization domain, ThemeRiver is usually used to visualize how different traffic flow categories or vehicle speeds change over time. Furthermore, Wang *et al.* [104] utilize the ThemeRiver to assist the users in exploring the trajectories distribution during a day by combining it with other spatial and statistic visualization. However, the ThemeRiver cannot visualize the directions of traffic flow and vehicle trajectories. For this problem, Guo *et al.* [15] improve the ThemeRiver by combining it with a glyph (encoding additional attributes as a compact visual element embedded in the streams) to present the vehicles at road intersections and the directions.

2.4.1.2 Branch Layout

Branch layout acts as a dynamic point-in-time interface for user actions within the branch. As shown in Figure 2.5 (b), the branch layout describes events or stories with nodes of the tree-like branch. Zeng *et al.* [105] create a branch-time (isotime) representation to explore



(a) Linear representation (b) Branching representation (c) Circular representation

Figure 2.5: Three visualization layouts for temporal data: (a) is a linear layout (ThemeRiver) visualizing the traffic situations, from the paper [15]; (b) is a branch layout to represent the subways' schedules in Boston, from the website [16]; (c) refers to the circular layout of two-dimensional ringmaps to visualize the timestamps in different levels, from the paper [17].

the mobility of humans in transport systems by displaying positions with nodes and timelines with edges. Compared to isochrone, the advantage of isotime is that it can visualize more mobility information by sacrificing spatial coordinates. Isotime also visualizes the timeline of the mobility of the flow map to explore accurate spatio-temporal information.

Similarly, the storyline [106] is another stabilizing technique for expressing the temporal changes and correlations, which presents the spatio-temporal changes through multiple lines with one axis referring to time and another axis referring to spatial marks. The storyline has the advantage of visualizing the timetable schedule for public transport. Doraiswamy *et al.* [107] propose a group index with stabilization for querying the events along with the time steps. Palomo *et al.* [108] create TR-EX to support the transport analysis of planned and accurate service, facilitating the linear time representation and stabilization method. It represents each transit with a polyline based on its station and stopping time, where its horizontal axes refers to the time of a day, and the vertical axes refers to the stations. Besides, Boston's Massachusetts Bay Transit Authority (MBTA) introduces a storyline [16] to represent four subways' schedules in Boston, where it displays four subways with horizontal axes and time of the day with vertical axes.

2.4.1.3 Circular Layout

Circular time refers to the repetition under a specific period (*e.g.,* days, weeks or months). It has cyclic properties that can be used for the visual representation using a cyclic layout, *e.g.,* a circle or calendar. Many recursive things happen within a specific period, such as the iterations of the morning peak of traffic flows. Visualizing these recursive things with a circular layout makes the recursive patterns more intuitive for users, as shown in Figure 2.5 (c).

The calendar view [109] is a time system for organizing circular periods, such as the months and weeks. Xu *et al.* [103] study the sequential pattern mining method with visualization techniques focusing on time series data. They mention that the calendar view could be used for exploring the temporal information in the time series data and this technique creates a whole life cycle data mining of time series data. Ferreira *et al.* [11] design a spatio-temporal data visual exploration system to allow users to compare the temporal and spatial relations. In

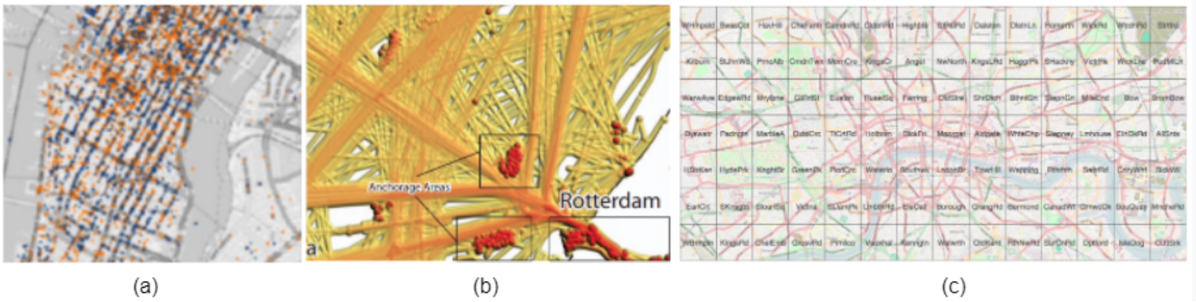


Figure 2.6: Spatial visualization for displaying the spatial data, which inspires the technique design in Chapter 4. (a) Point-based visualization of taxi trajectories, from the paper [11]; (b) Line-based visualization of vessels' trajectories, from the paper [18]; (c) Region-based visualization, from the paper [19].

their work, a calendar panel selects the specific days and hours, which means users can interactively operate it to select the different time dimensions. Seebacher *et al.* [25] design a visual system to analyze the urban heat island. In this work, they use a calendar view to visualize how the heat islands change over time, and the colors in the calendar view represent the amounts of hotspots.

Pu *et al.* [110] label the spatial information with a spiral view, which is also a typical visualization layout for circular periods. The spiral view is a ring-map-based radial layout design and is efficient at visualizing and analyzing the circular time, such as the 24 hours per day and seven days per week. The spiral view usually has two axes: radial direction and clockwise direction. In order to utilize the spiral view in traffic data exploration, Pu *et al.* [111] introduce the spiral view to visualize how the road traffic flow changes over a week with a hierarchical time level. This view contains two-time dimensions, which are days and hours. It represents the hour dimension with radius direction and the day dimension with the clockwise direction.

2.4.2 Spatial Visualization

This section introduces spatial visualization for spatial data, which generally contains the longitude and latitude coordinates representing the objects' positions in a specific time step, as shown in Figure 2.6. Spatial data analysis plays a vital role in the traffic domain to help traffic experts better know travel patterns and movement behaviors. This is what the spatial visualization techniques can do — displaying the key spatial information and critical components.

2.4.2.1 Density-based Visualization

Point-based visualization displays discrete traffic data samples with a point-relevant visual presentation. Each point visually encodes a unit (*e.g.,* a traveler or a vehicle) and their density reveals spatial patterns either statically or dynamically. Boston's Massachusetts Bay Transit Authority (MBTA) [16] presents the metro with points, as shown in Figure 2.7 (a), where the animation of moving dots offer an overview of the subway system.

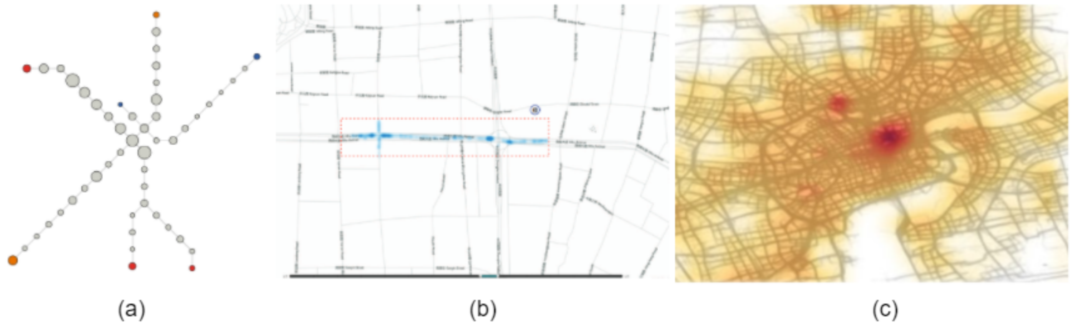


Figure 2.7: The density-based visualization that describes objects' position information and avoid visual clutter. (a) Boston's Massachusetts Bay Transit Authority (MBTA), from the website [16]; (b) Data-driven Transport Assessment, from the paper [20]; (c) Visual Analysis of Route Diversity, from the paper [21].

The advantage of point-based visualization is encoding the individuals' information, such as the movement directions and speeds. It can describe the individuals' characteristics as much as possible, but it would be inefficient when too many data items are in a specific space. In order to solve this problem, researchers use the cluttered method to present the movement group, such as using the heatmap. Wang *et al.* [20] propose a visual system to estimate the real traffic situations based on taxi trajectories, as shown in Figure 2.7 (b), where the heatmap introduced in this visual system can visualize traffic jams. Also, Liu *et al.* [21] create a heatmap view to present the road diversities of hotspots in a city, as shown in Figure 2.7 (c). The red colors in this heatmap show the locations where many vehicles pass by. Similarly, Liu *et al.* [112] create the road map overview with the heatmaps in an interactive real-time visual system.

2.4.2.2 Line-based Visualization

Line-based visualization can present the dynamic status of objects, such as their moving directions and speed, as shown in Figure 2.8. Compared with density-based visualization, the advantage of line-based visualization is in visualizing the moving objects and their moving process. Line-based visualization usually encodes the lines with different colors and widths to present the data attributes. Additionally, there is a range of visualization techniques using icons to display the directions of objects, as shown in Figure 2.8 (a), using the arrows to show the flow's direction. Traditional line-based visualization aims to visualize every single entity (*e.g.,* trajectories of every taxi) as much as possible. However, the amount of data is becoming enormous since various sensors and data storage capabilities have been developed. As a result, the line-based visualization tends to clutter because lines overlap. In order to overcome this problem, researchers visualize the data with groups. Guo *et al.* [22] create the flow mapping to visualize how the objects flow, such as the human migration among counties in the US, as shown in Figure 2.8 (a). In order to avoid messiness while visualizing all the flows, they aggregate the regions into different groups and then visualize the flows with arrows among these aggregated regions. Scheepens *et al.* [76] develop a visual system based on the lines to

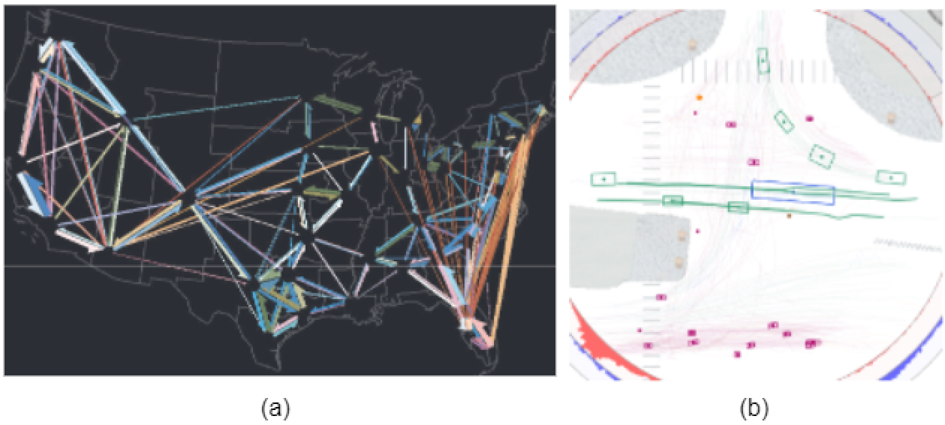


Figure 2.8: The line-based visualization that expresses objects' movement information, *e.g.,* direction or speeds. (a) A multivariate flow map, from the paper [22]; (b) Traffic trajectory at a road intersection, from the paper [15].

help operators select, filter, and compare traffic information using a combination of a particle system and a particular selection widget.

Moreover, line-based visualization efficiently represents the trajectory information (multiple individual points of the same objects connect based on time sequence) using different colors and symbols. Guo *et al.* [15] define the different symbols to present the vehicles while visualizing their trajectories, as shown in Figure 2.8 (b). It labels different vehicles with different shapes of rectangles and trajectories with different color lines. Zhao *et al.* [17] visualize the human trajectories to explore the human behaviors (*e.g.,* work, travel, and leisure) where they represent the movement patterns with different colors.

Additionally, researchers introduce the calculation method to assign the lines with new relative positions to reduce the clutter. Andrienko *et al.* [87] propose an approach to analyzing the group movement by calculating the trajectories' relative positions at each time point. The new relative positions respect the movement direction and group center to help users analyze the trajectories by reducing the clutter.

2.4.2.3 Region-based Visualization

Region-based visualization shows the traffic situations in every individual region, such as the administrative areas of cities. It is a highly aggregated type of visualization compared with line-based and point-based visualization, as shown in Figure 2.6 (c), where it visualizes the traffic flows as summaries using spatial structure, *e.g.,* administrative divisions and grids structure.

Generally, region-based visualization calculates the specific events (*e.g.,* traffic jams) based on the regions and then shows them with icons or other visual encodings on maps. Seebacher *et al.* [25] label the urban heat islands with circles and charts to visualize the detailed information in the specific regions having higher temperatures and describe the thematic changes over time with an extensive collection of documents. Xu *et al.* [113] propose the Geo map view to display the geographical distribution of user-specific topics based on the geographi-

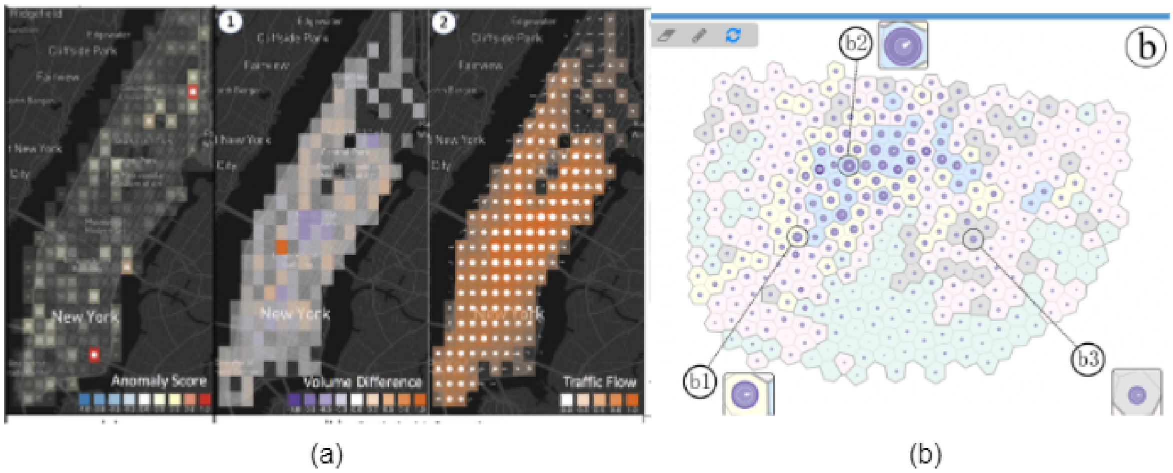


Figure 2.9: The Region-based visualization helps users have a perspective of the traffic based on the subdivision regions. (a) Voila, an overview of the anomalous information in the form of a heatmap, extracted from the paper [23]; (b) MobiSeg, a region segmentation visualization, is extracted from the paper [24].

cal heatmap. After combining with the text topic view, this Geo map can help users analyze the urban characteristics. Cao *et al.* [23] design the *Voila* visual system to display abnormal events (*e.g.,* traffic incidents) in specific regions. They create rectangles to split the urban region shown in Figure 2.9 (a). The rectangles are encoded with different colors and shapes to visualize the events and the abnormal percentage. This system can meet two requirements in real-world applications: online monitoring and interactivity. The results indicate that the system is robust and points out the possible research direction in the traffic domain, such as developing new algorithms with forecasting and prediction capability.

Moreover, another region-based visualization comes from particular region division methods, such as those with Voronoi-based methods. Wu *et al.* [24] create the Voronoi-based texture map to reflect region characteristics of the urban-based human movement activities. They separate the city regions with similar activity patterns, as shown in Figure 2.9 (b). Other relevant region-based visualizations in the traffic domain are the grids technique and the tree map technique. Wood *et al.* [114] have worked a lot on the OD visualization based on the grid technique. They introduce the treemaps into the spatio-temporal visualization for exploring the large multivariate spatio-temporal dataset. Also, they use treemaps to explore the traffic speed distribution in London and the relations of other attributes, such as the vehicle type, day of the week, and hour of the day, concurrently.

2.4.3 Spatio-temporal Visualization

Spatio-temporal visualization focuses on how the objects move along with the timestamps in a single space. A typical approach is the Space-time Cube, which is a framework for representing how phenomena change over time within geographic space. It contains the geometry information as a plane and the time information using another dimension. In a space-time

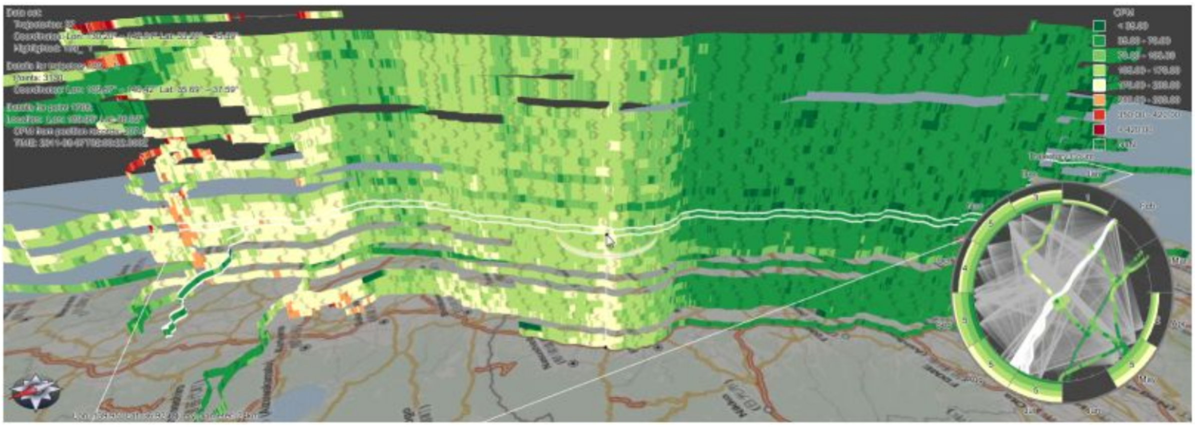


Figure 2.10: Stacking-based visualization of trajectory attribute data. A 2D map serves as the spatial context, and the stacked bands refer to the trajectories where the colors encode the data attribute values, from the paper [13].

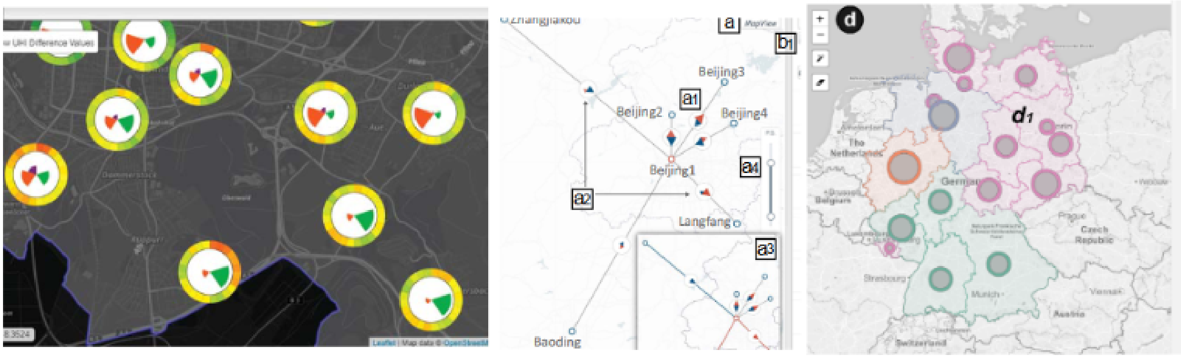
cube, each cube represents a slice of time (*e.g.,* top cubes have newer timestamps, and bottom cubes have older timestamps) [115]. Tominski *et al.* [13] utilize the space-time cube to visualize the trajectories and the traffic flow, to describe how they change over time and to understand the objects' mobility, as shown in Figure 2.10. Furthermore, Kang *et al.* [96] use the Space-time Cube to extract human mobility during the time of day and on different days from the millions of mobile phone users. It categorizes the users into different groups and analyzes their behaviors, classifying them based on age or gender.

2.4.4 Event-based Visualization

This section introduces event-based visualization. Event data usually contains the event lists that happened on the roads, such as incidents, traffic accidents, and road works. Generally, the event data involve the positions and time, which means it is spatio-temporal data. Thus, event-based visualization is a visual type that presents events in the spatio-temporal space.

Visualization techniques tend to highlight the event information on the map by using icons or specific shapes and colors. For instance, Deng *et al.* [26] design *Compass* to help users capture the dynamic urban causality in urban time series, as shown in Figure 2.11 (b). It uses the compass as icon displayed on the map to present the flowing directions. Similarly, Seebacher *et al.* [25] introduce the pie chart as an icon to present spatial and temporal information about urban heat islands, as shown in Figure 2.11 (a). Liu *et al.* [27] encode events on the map with circles, as shown in Figure 2.11 (c), where circles visualize quarter statistics for the normalized deviation from 0% to 100%.

Events or incidents are not identified in the raw dataset. Therefore, there are algorithms or methods aiming to detect events, *e.g.,* Doraiswamy *et al.* [107] propose an event-guided exploration system, which creates a time-varying data structure to make it automatically identify the events. It flattens the hotspot map in different time slices to compare how the event-based information changes. Tang *et al.* [117] introduce the Latent Dirichlet Allocation (LDA) model into the interactive visualization technique to extract the spatial and temporal information of



(a) (b) (c)

Figure 2.11: Three examples of event-based visualizations where (a) uses pie chart to represent the urban heat island, from the paper [25], (b) uses compass as icon to display the urban causality, from the paper [26], and (c) utilizes the size of circles to display data attribute values, from the paper [27].

the traffic events containing semantic information. After combining the LDA model, the interactive tool can display the topic analysis results.

Event-based visualization also aims to analyze the events' impact on traffic situations (*e.g.,* traffic jams). For instance, in order to explore what impacts the incidents have on the road and how the past events affect the traffic, Anwar *et al.* [14] design *Traffic Origins* to visualize the traffic situations and the incident's information, which employs an expanding circle to uncover the underlying traffic flow map and their temporal information.

A map legend is a description, explanation, or table of symbols printed on a map or chart to interpret traffic information and traffic events. The traffic events usually contain road situations, such as road works or traffic accidents. Also, it involves information on the map scale, such as the color scale reflecting traffic flows and traffic density (*e.g.,* free flow or traffic congestion). For the research of the legend in maps, Jason *et al.* [118] develop the guidelines for legend design in a visualization context that is derived from cartographic literature and the application from EDINA, which provides digital mapping services. We will introduce a quantitative value categorization technique for creating a suitable scale (*e.g.,* could be used as the legend in maps) of traffic flows or traffic density in Chapter 3.

2.4.5 Multiple Coordinated Visualization (MCV)

Most visualization techniques introduced in the previous sections are single views. However, they are limited when the data contain a large volume of information or many attributes, especially many visual representations that use different paradigms requiring their own space. Thus, we review multiple coordinated visualization (MCV) [120] in this section as the foundation of the research in Chapter 6. Initially, MCV focused on the model and the techniques [121]. After several years, researchers applied the MCV to different domains (*e.g.,* traffic control centers or business analysis). Multiple views have different meanings in different sentences. A study done by Roberts *et al.* [122] related to terminology and phraseology in the visualization community. The results of this study help users better understand MCV and use them in suitable ways to improve their writing. Besides, as introduced by Roberts *et al.* [121], MCV has four types based on the tasks: Overview & detail views use one view to visualize the whole dataset and another view to visualize the part of the dataset; focus and context views are similar to overview & detail, but the context does not display the overview of the whole dataset; difference views are achieved by merging several views; small multiples are the high density of the views or matrix.

The design space of MCV affects the availability and efficiency of realizing tasks. According to a study done by Chen *et al.* [123], the design space of MCV can be described in two aspects: composition and configuration. The composition defines how many views they use and which presentation types are in each view. Configuration describes the spatial arrangement of view layouts. Based on the two aspects, they summarize some guidelines for multiple view designs. Besides, the MCV combines with other visualization to construct new design spaces and improve visual presentation. For example, Roberts *et al.* [120] link the multiple views to the 3D visualization to explore the deeper patterns since it is not easy to 'see inside' in the 3D visualization when dealing with too much data.

As introduced in the previous paragraph, a good design space can improve the availability and efficiency of MCV. However, there is an issue — how to evaluate the design space of MCV? For this issue, Shao *et al.* [124] create the model based on Bayesian probabilistic inference to evaluate the effect of design factors, including views, coordination, and designers. This calculation introduces the maximum area and weighted average aspect ratios as the geometric metric. The maximum area ratio reflects the ratio of different views' areas. The weighted average aspect ratio can effectively clarify a presentation. Besides, Langner *et al.* [125] report on the critical consideration for the multiple views designs for wall-sized displays. They develop full-function interactive tools with 45 views based on criminal activities. The user study aims to learn how people use multiple view wall-sized displays with closed and overview distances. The results show that users would like to interact with the visual elements freely and would also like to walk close to each other.

Apart from the theoretical method for improving the design space, there are also empirical studies reviewing the MCV in the visualization community. Al-Maneea *et al.* [126] present the analysis of layout patterns with multiple views by collecting 491 images from the visualization community conferences and journals. They analyze the topology in juxtaposed views and eventually provide guides in designing multiple views. Also, Lyi *et al.* [127] study the layout effects from previous information visualization by reviewing three comparative layouts: juxtaposition, superposition, and explicit encoding.

Designing multiple view systems requires specific design methods, which involves many steps, *e.g.,* brainstorm, the layout design and focus zone design [128]. Apart from the traditional design method, novel MCV design methods rise along with the development of visualization techniques, such as automatic layout generation techniques. Cruz *et al.* [129] develop a visual system to assist designers in automatically generating MCV. First, this system can process the data component to create the relations among these different data sources, and then, the heterogeneous data help users create the views with their preferences. Finally, they enable users to construct integrated visualization with multiple views. Wu *et al.* [130] propose a deep learning-based method to help users create a suitable dashboard for analyzing the data. Xu *et al.* [131] provide automatic optimization of the layout for multiple views to make it beautiful when there are ambiguous layout problems. Al-maneea *et al.* [126] create a tool to help users easily create juxtaposed view layouts. This tool can help users change the layout types and replace the views in the bounding box of the layout. Similarly, Boukhelifa *et al.* [132] propose a model for creating the coordination of multiple views in the exploratory visualization. Eichner *et al.* [133] report a display model in a multiple-view environment. It helps users automatically generate the layout in the design space. This model considers three components: ”What”, ”When”, and ”Where”. ”What” refers to the contents that should be discussed, ”When” refers to the sequence, and ”Where” refers to how we display the content in the display environment.

2.4.6 Conclusion

To summarize, we reviewed the visualization techniques for road traffic data, including temporal, spatial, spatio-temporal, event-based, and multiple coordinated visualizations. Three typical layouts of temporal visualization support us with the background knowledge to design a temporal visualization technique in Chapter 5 by developing a set-based approach to avoid overplot generated from line charts. Spatial visualization techniques generally highlight the spatial data features through clustering, heatmap, and region-based visualization approaches. We focus on spatial visualization in Chapter 4 by developing a nest-based visualization technique to explore both the explicit relations (*e.g.,* the trajectory positions) and implicit relations (*e.g.,* travel distance and speed) of geo-coded entities. Event-based visualization aims to visualize the implicit traffic events (*e.g.,* traffic congestion). Mostly, traffic events are hidden in the raw data. In order to find an efficient way to define the traffic events in raw data, we develop a quantitative value categorization technique in Chapter 3. Spatio-temporal and multiple coordinated visualizations describe two research directions (3D and multiple coordinated visualizations) while visualizing both temporal and spatial information from heterogeneous data. This manuscript utilizes multiple coordinated visualizations to deploy the traffic control centers in Chapter 6.

Chapter 3

Univariate Visualization

Contents

Any use of ”we” in this chapter refers to Liqun Liu and Romain Vuillemot. In this work, we have a paper published as follow:

- Liqun Liu and Romain Vuillemot. ”Categorizing Quantities using an Interactive Fuzzy Membership Function,” In *The* 12*th International Conference on Information Visualisation Theory and Applications*, P. 8, On-line, Feb 2021. (Link)

3.1 Context and Motivation

This chapter focuses on the problem of analyzing single attribute data, commonly called *univariate* data (introduced in Section 1.2 and Section 2.2.1). Such problem is frequent with road traffic data, such as when analysts seek to *categorize traffic flows or vehicle speeds*. This categorization can benefit traffic operators of traffic control centers (introduced in Section 2.3.1) in monitoring the traffic task (introduced in Section 2.3.2) by providing a suitable traffic flow categorization scheme. In this case, analysts tend to categorize, for instance, vehicles speeds as `LOW`, `MIDDLE`, and `HIGH`. However, the `HIGH` speed could have different values from very

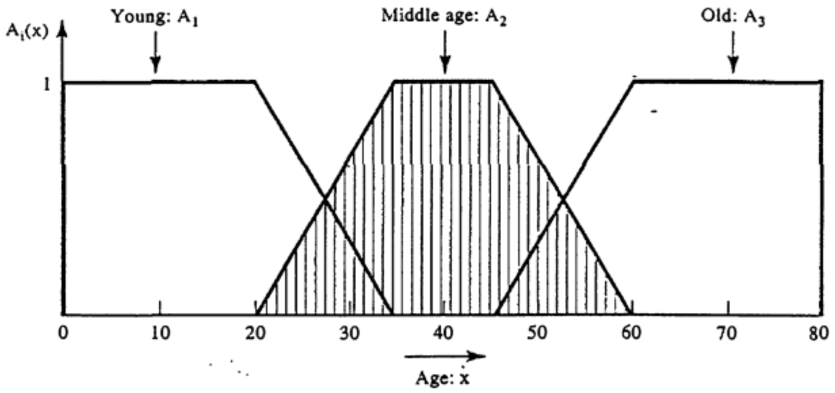


Figure 3.1: The membership function categorizes the age into `Young`, `Middle-aged`, and `Old`, extracted from the book [28]. We extend this function in this chapter to analyze univariate traffic data.

Name	Link
Online Prototype	`https://observablehq.com/d/0820d2ad9cfa734d`
Website for user study	`https://observablehq.com/d/45001390e4f1b08f`
User study results	`https://observablehq.com/d/9479ec50c448978d`

Table 3.1: Supplementary materials. It includes an online prototype with a feature for users to upload their data, a website for user study, and a website for the user study results.

`HIGH` to moderately `HIGH`. To capture such nuance, there is a need to mimic the logic of human thoughts and reasoning that is often subjective using the domain or prior knowledge. Such a process raises the following needs:

- *An explicit mapping of those categories:* the mapping function between quantities and categories should be clearly defined.
- *A customize-able mapping to categories that can vary across analysis sessions and analysts:* the mapping could be personalized and change based on the task to achieve.
- *The transfer between analysts and a user should be possible:* by some means of communication like a legend or a visual encoding that explains the current categorization scheme being used.

To meet those three needs, this chapter designs an interactive tool (FuzzyCut) to make the univariate data categorization process explicit and flexible to obtain a better traffic flow categorization scheme. Our approach relies upon fuzzy logic [134] created in the 1960s by Zadeh to model domains with imprecise information [135], which we argue provides the theoretical framework to address the above issues. In particular, we rely upon a visual representation from this theory called the *membership function*, as shown in Figure 3.1, which is a line chart of the mapping function between the univariate data value intervals and fuzzy categories. Using FuzzyCut, users can adjust the shape of the membership function to generate the fuzzy categories belonging to a specific set with confidence.

We provide an implementation demonstrating how it supports the categorization of multiple datasets in Observable notebook, a reactive, web-based framework compliant with ES6 JavaScript modules, using library D3 [136]. Table 3.1 lists all the prototypes, code, and study results presented in this chapter.

3.2 Related Works

This chapter focuses on categorizing quantities with fuzzy logic theory and quantifying these generated categories with a membership degree. This section reviews papers in probabilistic classification, fuzzy visualization, and uncertainty visualization.

3.2.1 Visualization of Probabilistic Classification

The general probabilistic classification visualizations focus on the quantitative and multiple attribute data. A visual system proposed by Seifert *et al.* [137] allows users to understand the process of classification and results, which handles multiple attributes data formats. For the multiple attributes data classification, Rheingans *et al.* [138] develop a technique to visualize high-dimensional predictive results with richer representation, *e.g.,* confusion matrices (a specific table layout that allows visualization of an algorithm performance) to help users understand high-dimensional data space. Besides, there are studies on visualizing probabilistic classifications generated from machine learning methods. A visual interactive analysis technique proposed by Alsllakh *et al.* [139] evaluates the effectiveness of classifiers to help machine learning experts discover the possible reasons for incorrect classification. This tool emphasizes the classification probabilities of items and make relations with a false negative and a false positive. Also, UnTangle Map proposed by Cao *et al.* [140, 141] using connected triangles to represent the set of data items can make efficient relations between data items and their probabilistic labels.

3.2.2 Fuzzy Visualization

There exist visualizations using fuzzy logic theory to capture the ambiguity categorization (we call it fuzzy visualization), where *fuzzy* represents that the truth value may range between completely false and completely true. For example, the Disk diagrams [142] proposed by Yeseul Park *et al.* can visualize fuzzy set (a class of objects with a continuum of grades of membership). It describes the complexity of fuzzy sets by showing the degree among sets with the layout of star coordinates. Besides, Zhu *et al.* [143] extend the circular disk diagram layouts to improve sets membership analysis by using color opacity and optimized layout. It conveys fuzzy set membership and reveals the uncertain owner-member relationship (*i.e.* the relationship between a value and a set).

There are several mixed methods for visualizing fuzzy sets and fuzzy clustering (each data point can belong to more than one cluster). The typical one is the combination between radial coordinate and parallel coordinate plot, introduced by Zhou *et al.* [144, 145]. This method reflects real-world clustering scenarios and improves the understanding of fuzzy clusters. Similarly, RadViz proposed by Sharko *et al.* [146] visualizes the fuzzy clustering of multiple dimensional datasets using radial visualization and the dimensional reduction method. Also, Rose Diagram proposed by Buck *et al.* [147] uses vectors of fuzzy attributes to visualize a fuzzy weighted graph. This method is an extension of the standard polar area diagram (a type of pie chart) and is helpful for decision-makers to choose a better way between several options while estimating the potential trade-offs.

With the development of visualization techniques, fuzzy visualization has become more abundant. Hall *et al.* [148] introduce parallel coordinates to fuzzy visualization and transform 3D or more than three-dimensional data into two dimensions without losing information. Similarly, Pham *et al.* [149] introduce the 3D parallel coordinates to visualize fuzzy data. Its advantage is that it is easier to distinguish the core and support in fuzzy sets. Furthermore, visualization techniques use novel methods to render fuzzy relations. Berthold *et al.* [150] propose a model to visualize the collection of fuzzy points in parallel coordinates. The parallel coordinates visualize the degree of membership function with different degrees of shading. Besides, Caha *et al.* [151] propose a hue saturation lightness (HSL) method to depict some essential values of fuzzy surfaces. This approach can be utilized in visualizing information of fuzzy numbers, vector data, and uncertainty data.

3.2.3 Uncertainty Visualization

Research on uncertainty visualization also offers efficient methods to convey ambiguity during the categorization process. As introduced by Brodlie *et al.* [152], the uncertainty of visualization exists in any data type, *e.g.,* point data, scalar data, multi-field scalar data, and vector data. Also, the visualization technique on uncertainty is challenging for various reasons, such as the complex status of uncertainty and the uncertainty information appearing in different ways. In order to address the challenges in uncertainty visualization, Skeels *et al.* [153] propose a classification of uncertainty for information visualization. It includes five categories: disagreement uncertainty from the difference of multiple times measurements, completeness uncertainty from the missing values, inference uncertainty from the model and prediction, measurement precision uncertainty from imprecise measurements, and credibility uncertainty from the conflict of the different data sources. Much effort has been devoted to designing visualization techniques, such as what Dong *et al.* [154] did, an interactive tool to help users recognize the situations and comprehend the ambiguity.

In conclusion, these works have presented several probabilistic uncertainty classifications with visual encoding methods. However, these works only support the communication of already created categories and not their generation by users to capture uncertain information.

3.3 Defining Crisp and Fuzzy Membership Functions

The challenge we tackle is the explicit mapping between quantities and categories. While most visualization techniques and tools usually address it during the data pre-processing steps (if not

ID	Time	Lat	Lon	Speed (km/h)	Speed Category
2618113	20130901000140	30.6199659	114.2990449	30	LOW
2618113	20130901000205	30.627190	114.2283287	57	MIDDLE
2618113	20130901000255	30.621745	114.2707959	65	MIDDLE
2618113	20130901000307	30.620711	114.260443	101	HIGH

Table 3.2: Dataset with quantities and categories. It introduces the data format of taxis trajectories, including the quantities (*Speed*) and categories (*Speed Category*)

yet in the dataset), it remains internal—or at best using a legend—without providing a fully explicit set of customization for this mapping.

To present our approach, we progressively introduce the definitions by first stating our challenge in finding the relationship between Q (Quantitative) and C (Categories) as a mapping function:

$$Quantity \rightarrow Category \tag{3.1}$$

Quantities are the measures of counts or values expressed by numbers (*e.g.,* $30km/h$, as shown in Table 3.2). On the contrary, categories are measures of type and can be expressed by a symbol, name, or label (*e.g.,* LOW or HIGH, as shown in Table 3.2). The mapping function connects quantities (Q) and categories (C), *e.g.,* the connection between quantity *"Speed"* and category "Speed Category" (listed in Table 3.2), shown as $[0, :] \rightarrow < LOW, MIDDLE, HIGH >$. Thus, we define the mapping from quantities to categories as Equation (3.2):

$$x \rightarrow \begin{cases} LOW & if\ speed(x) \leq 30 \\ MIDDLE & if\ 30 < speed(x) \leq 80 \\ HIGH & if\ 80 < speed(x). \end{cases} \tag{3.2}$$

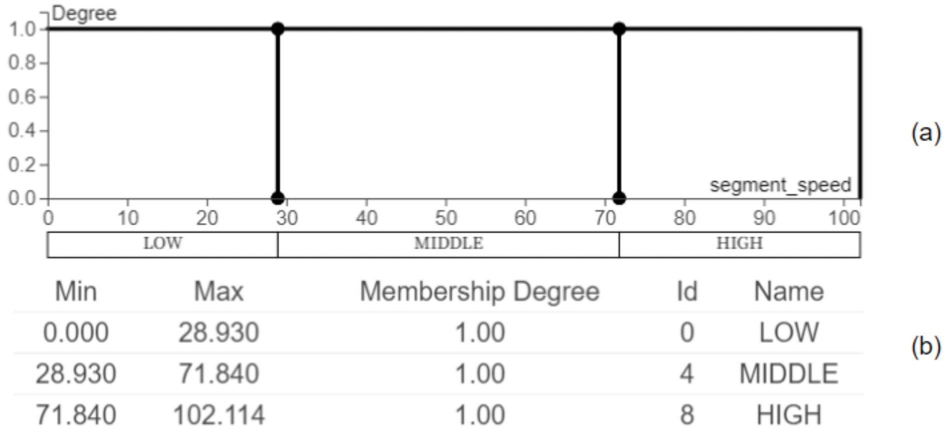
In order to address this mapping problem, we design a visualization technique based on the line chart, mapping quantitative scales to a domain of user-defined categories. Figure 3.2 illustrates the user interface that represents the mapping of each value to categories. The table in Figure 3.2 indicates the categorization result for each category.

This mapping type relates to the classical sets theory, where categorization is a *crisp* process that splits quantities into categories with a binary function: accepting or rejecting the object belonging to a category [155]. As a result, an element x either belongs to a category or not. If there is a set W that is not empty and a set $S \subset W$, the characteristic function of S shows as follows:

$$f_S(x) = \begin{cases} 1 & if x \in S \\ 0 & if x \notin S \end{cases} \tag{3.3}$$

where $f_S(x)$ is the function and the domain of $f_S(x)$ is W. The value of $f_x(x)$ is in set $\{0, 1\}$. If $f_S(x) = 1$, it means element x belongs to set S; if $f_S(x) = 0$, it illustrates element x does not belong to set S so that this mapping function $f_S(x) \rightarrow \{0, 1\}$ are able to completely represent the relationship between element x and set S.

However, the previous mapping function cannot capture ambiguous sets. For example, it cannot represent the MIDDLE, RELATIVELY HIGH, and HIGH because RELATIVELY HIGH



Min	Max	Membership Degree	Id	Name
0.000	28.930	1.00	0	LOW
28.930	71.840	1.00	4	MIDDLE
71.840	102.114	1.00	8	HIGH

Figure 3.2: Categorization using a crisp mapping function, there are three crisp categories generated from the function shown in (a) and the detailed information of categories in (b).

overlaps with `HIGH` and `MIDDLE`. Thus, it is not any more suitable for separating quantitative values when there is ambiguity or when there is more than one quantitative scale.

In order to solve the problem that does not have sharp boundaries while categorizing, Zadeh *et al.* propose the membership function. It is a line chart of the mapping function between the univariate data value intervals and fuzzy categories, presenting the degree of truth (the membership degree) as an extension of valuation [40]. The membership function can be described as $f_a(x) \rightarrow [0, 1]$. The value of $f_a(x)$ means the membership degree of the membership function. With $f_a(x) = 1$, it represents the complete belongingness while $f_a(x) = 0$ shows the complete non-belongingness. The membership degree has the interval in $[0, 1]$, where this value represents how many possibilities element x belongs to set A. The membership degree $f_a(x)$ can be calculated in multiple ways, *e.g.,* triangular, trapezoidal, Gaussian, and sigmoidal functions. In this chapter, we select the trapezoidal membership function (we will implement other membership functions in further work), and it is given by:

$$f(x) = \begin{cases} 0 & if \quad x < a \\ (x-a)/(b-a) & if \quad a \leq x < b \\ 1 & if \quad b \leq x < c \\ (d-x)/(d-c) & if \quad c \leq x < c \\ 0 & if \quad x > c \end{cases} \tag{3.4}$$

where, a, b, c, d are the parameters of the trapezoidal membership function. The membership functions can be displayed over a line chart in which the x-axis is the quantitative value, and the y-axis is the membership degree. Each line is a category whose membership degrees are from 0 or 1.

To implement the membership function into interactive categorization, we design a visualization technique named FuzzyCut. It can map quantities to categories with membership degrees as an extension of categories, as shown in Figure 3.3. FuzzyCut involves three parameters: 1) *Core*: it refers to the elements completely or fully belonging to one membership, and the membership degree is equal to 1; 2) *Support*: it refers to the elements that belong to

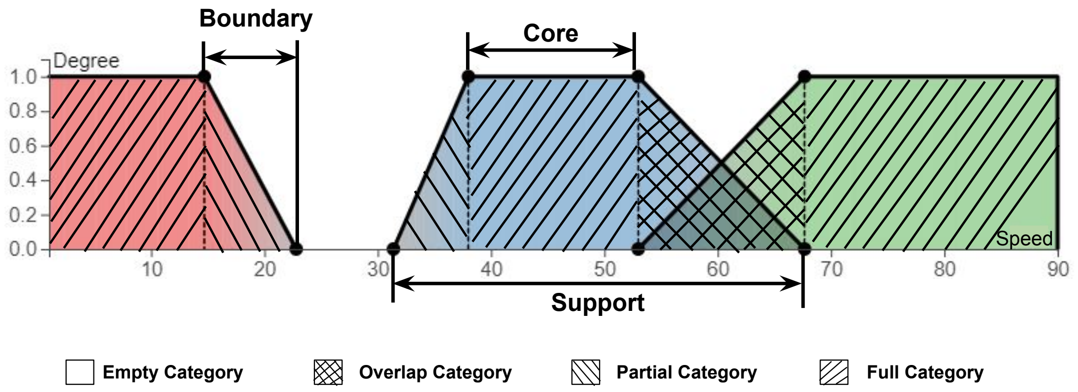


Figure 3.3: Parameters and categories of membership function. The parameters contain *Core*, *Support*, and *Boundary*. Based on these parameters, there are four specific categories generated: *Full Category* with only one membership degree is equal to 1, *Partial Category* with only one membership but the membership degree is less than 1 and more than 0, *Empty Category* without any membership, and *Overlap Category* with two memberships.

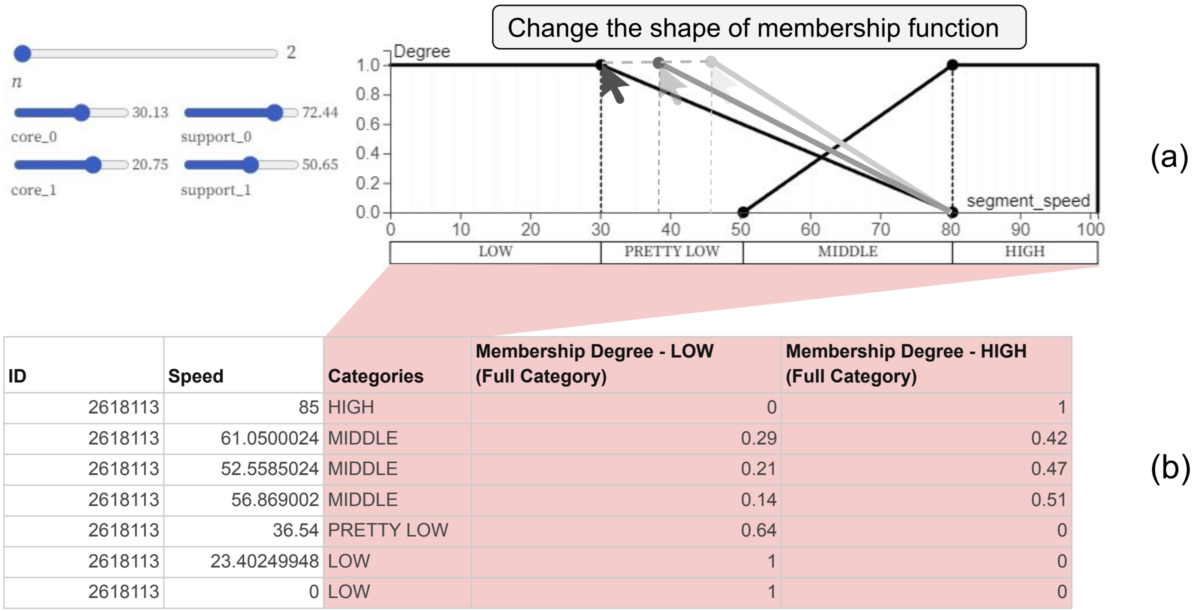
one membership with membership degrees more than 0; and 3) *Boundary*: the categories that contain elements that have non-zero memberships and incomplete memberships.

FuzzyCut generates the categories based on the different combinations of parameters in the membership function. Figure 3.3 includes the generated categories: *Partial Category*, *Empty Category*, *Full Category*, and *Overlap Category*. *Empty Category* refers to the categories that do not belong to any category, and their membership degrees are equal to 0. *Overlap Category* refers to the categories that belong to two memberships simultaneously, with membership degrees more than 0 and less than 1. *Partial Category* refers to the categories that only belong to one membership but with membership degrees less than 1 and more than 0. *Full Category* corresponds to parameter *Core*, with membership degree equal to 1.

3.4 Interactive Membership Function

We have implemented FuzzyCut as an interactive prototype (link) in Observable notebook [136] (introduced in Section 3.1). Figure 3.4 shows the interface for users to adjust parameters, thereby generating the categories they want. In (a), the x-axis represents quantitative values, and the y-axis shows the membership degree (μ). In (b), the table shows the data and derived attributes. Users can generate the categories they want in two steps:

- **Change the shape of the membership function.** Users can adjust the shape of the membership function by dragging the blue sliders (on the left of Figure 3.4) to adjust the parameters (*e.g.*, *core*, *support*, and *n*). Moreover, users can adjust these parameters by dragging small black circles (on the membership function).
- **Define the name of generated categories.** Users define the name of generated categories by inputting texts in the rectangles below the membership function. The generated membership values and derived attributes are in Figure 3.4 (b).



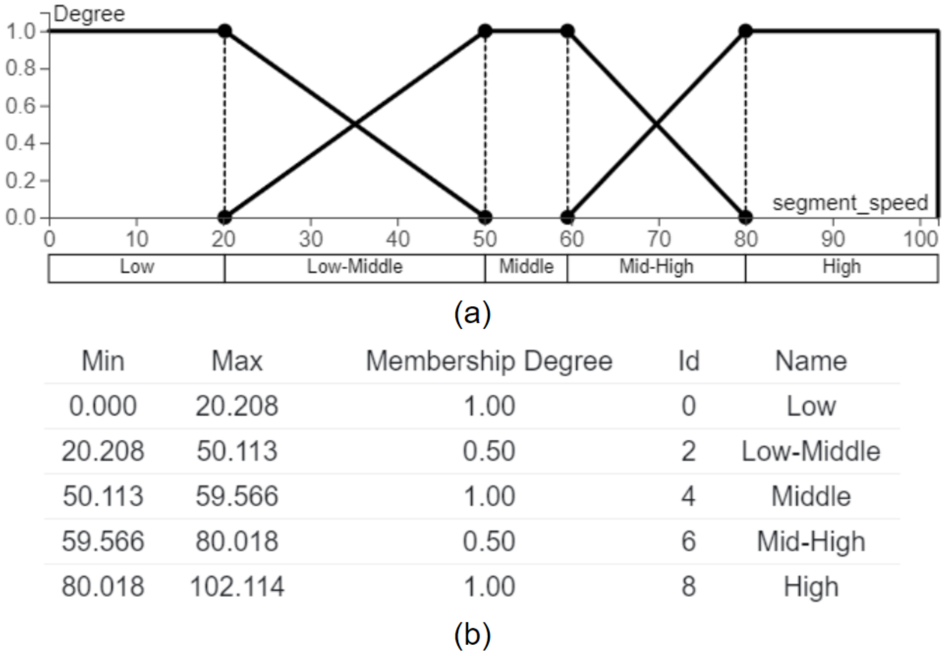
ID	Speed	Categories	Membership Degree - LOW (Full Category)	Membership Degree - HIGH (Full Category)
2618113	85	HIGH	0	1
2618113	61.0500024	MIDDLE	0.29	0.42
2618113	52.5585024	MIDDLE	0.21	0.47
2618113	56.869002	MIDDLE	0.14	0.51
2618113	36.54	PRETTY LOW	0.64	0
2618113	23.40249948	LOW	1	0
2618113	0	LOW	1	0

Figure 3.4: The illustration of interaction on FuzzyCut. Users can adjust the shape of the membership function by dragging the black points on the membership function as illustrated with three arrows or dragging the parameter sliders (blue) shown on the left in (a). Based on the membership function shapes, it can create categories with different labels (`LOW`, `PRETTY LOW`, `MIDDLE`, `HIGH`). The data format and the derived new attributes are in table (b). The raw data includes the quantitative data *speed* and derived attributes, including *Categories* and the *membership degrees*, such as the columns *Membership Degree - LOW (Full Category)* and *Membership Degree - HIGH (Full Category)*.

3.5 Illustrative Examples with Taxi Speed Dataset

We have implemented FuzzyCut with several data types, such as taxi speed, temperature, and age datasets. This section introduces how users categorize the taxi speed values using FuzzyCut to generate the categories (We introduce other dataset implementations in Section 6.2.1). In this example, the speed data implemented are half-bounded intervals (speed value $x \in [0, +\infty]$) and continuous value types (numeric variables that have an infinite number of values between any two values), which have been introduced as the taxi trajectory data in Section 2.1.5. While analyzing the taxi data, traffic analysts usually are interested in characterizing taxi driving behaviors (*e.g.,* drunk driving or fatigued driving). In this analysis process, speed is an essential parameter to reflect these driving behaviors.

Figure 3.5 shows FuzzyCut implemented with the taxi speed data. (a) shows the interface of the interactive membership function, and (b) refers to the generated categories. In this example, FuzzyCut separates the speed into five categories, three of which are main categories named *Low*, *Middle*, and *High*. Also, there are two *Overlap Categories* with membership degrees less than 1, named *Low-Middle* (overlap between *Low* and *Middle*) and *Mid-High* (overlap between *Middle* and *High*). Other categories, more domain-specific, could have been used, *e.g., Slow*, *Fast*, by editing the label input field in the prototype.



Min	Max	Membership Degree	Id	Name
0.000	20.208	1.00	0	Low
20.208	50.113	0.50	2	Low-Middle
50.113	59.566	1.00	4	Middle
59.566	80.018	0.50	6	Mid-High
80.018	102.114	1.00	8	High

Figure 3.5: FuzzyCut separates taxi speed values into five categories. There are two categories with membership degrees less than 1 (fuzzy categories), named *Low-Middle* and *Mid-High*.

3.6 User Study

In order to verify the effectiveness and limitation of FuzzyCut, we conducted a user study. Our main goal was to investigate how easily users could use the technique and how effectively users could categorize quantities with different data types. We conducted this study with *Age* and *Temperature* datasets, to have more participants with quasi homogeneous expertise in a domain than road traffic. By interviewing these users who had the experience of categorizing quantities and exploring the habits humans resonate on quantities, we investigated three hypotheses in this user study:

H1 The interactive technique would help people categorize quantities and affect their categorization results.

H2 The data size would affect the categorization results, which means that the same user will create different categorizations with different data sizes.

H3 Users would name the generated categories with comparative and descriptive words (*e.g.,* `HIGH` and `RELATIVELY HIGH`).

3.6.1 User Study Setup

In order to test if the hypotheses were `Valid` or `Not`, we designed the user study in three steps. We first recruited participants in the computer science field or not, and then we invited participants to categorize the quantities, *i.e. age* and *temperature* data. Finally, we measured the categorizing quantities results.

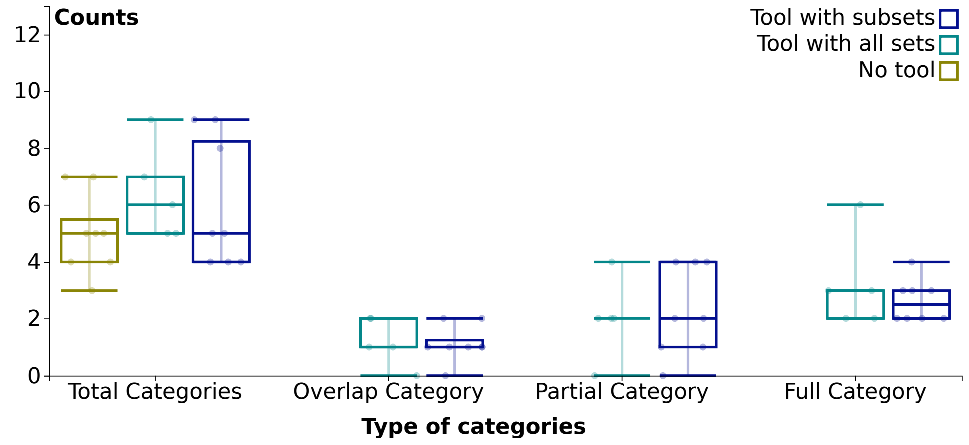


Figure 3.6: The statistic of categories generated from eight participants with the age dataset. The x-axis represents the type of categories, such as *Overlap Category*, *Full Category*, and *Partial Category*, which are introduced in Figure 3.3; *Total categories* refers to all the generated categories; the y-axis represents the number of each category type; the different colors refer to the categorization without using FuzzyCut, the categorization using FuzzyCut but with subsets and the categorization using FuzzyCut with the entire dataset.

Participants: We recruited eight participants (three females) to participate in this study. Their ages ranged from 26 to 32 and the mean age was 28.6. Two of them had computer science backgrounds and rich experience in reasoning on quantities. None had used our technique before. We invited all participants to fill in a pre-study questionnaire relating to basic information (*e.g.,* Whether they are similar to visualization and have a computer science background). In this pre-study questionnaire, participants were also required to categorize the *age* and *temperature* data based on their background knowledge.

Tasks: We invited all participants to operate FuzzyCut at least once and collected the data log, including their operating activities and the categorization results. We designed a web page (link) where users could operate FuzzyCut following the introduction. On this web page, we explained FuzzyCut with examples and then implemented two prototypes using *age* and *temperature dataset*. Through users' activities on this web page, we recorded the operating results and saved these results to a remote server. This operation included three steps:

- In **Step. 0**, participants watched a brief introduction to FuzzyCut, which included some basic information and workflow introduced by a video.
- In **Step. 1**, we invited participants to create a specific categorization configuration (we pre-supplied) in aspects of name and range of categories. The configuration helped participants understand how to manipulate FuzzyCut quickly.
- In **Step. 2**, we invited the participants to categorize *age* and *temperature* datasets with different data sizes. Firstly, participants categorized the data with a small amount and then with a greater amount if the participants thought the categories should be updated when the amount increased.

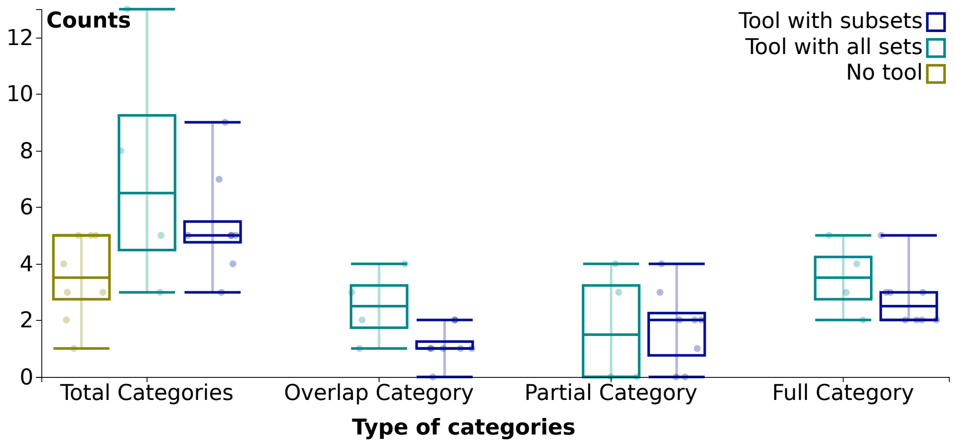


Figure 3.7: The statistic of fuzzy categories generated from 8 participants with the temperature dataset. The x-axis represents the type of categories, such as *Overlap Category*, *Full Category*, and *Partial Category*, as shown in Figure 3.3; *Total categories* refers to all the generated categories; the y-axis represents the number of each type of categories; the different colors refer to the categorization without using our tool, the categorization using our tool but with subsets and the categorization using our tool with the entire dataset.

After this operation, participants needed to fill in another questionnaire with qualitative feedback about FuzzyCut. During the study, we encouraged participants to think aloud about what they were doing, how they thought, and why they operated the parameters. We took notes about these questions and assisted if it is necessary.

Measures: We recorded the information to a remote server, which was related to participants' operating activities and the categories generated by users. Activities were how users changed the shape of the membership function (introduced in Section 3.4). The categories included maximum values, minimum values, names, and membership degrees. We utilized these attributes to calculate the participants' categorizing behaviors and reasoning. For example, the first-time and second-time categorizations could reflect whether participants were disturbed by the data size.

3.6.2 Results

We visualized the results of categories for *age* and *temperature* datasets in Figure 3.6 and Figure 3.7, respectively. The x-axis represents the category types (have illustrated the concepts of these categories in Figure 3.3), which contains: *Total Categories* for all categories generated, *Overlap Category* for the categories having two memberships, *Partial Category* for the categories only having one incomplete membership (the membership degree is more than 0 but less than 1) and *Full Category* for the categories only having one complete membership (the membership degree is equal to 1). The different colors of the boxes show the categorization in different ways: categorization without using any tool, first-time categorization using FuzzyCut with the sub dataset, and second-time categorization using FuzzyCut with the entire dataset. The y-axis represents the number of each category type and the points in the boxes represent the individual participants.

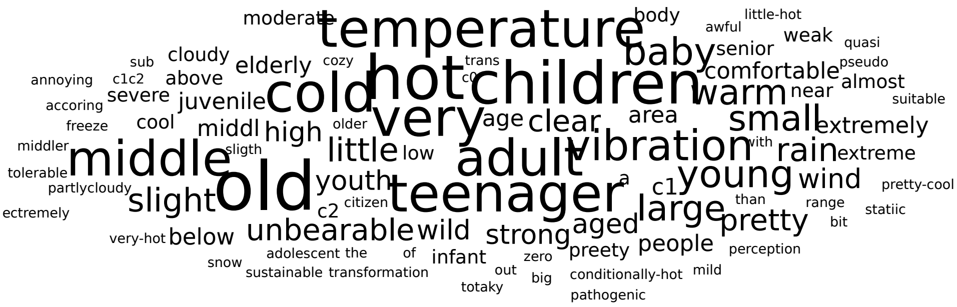


Figure 3.8: Usage frequencies of categories' names. The bigger size of the text, the more participants used such names.

As shown in Figure 3.6 and Figure 3.7, the categorizations without using FuzzyCut do not generate *Overlap Category*, *Full Category*, and *Partial Category*. Also, the two figures show that compared with the categorization without using FuzzyCut, the categorizations using FuzzyCut with either the sub dataset or the entire dataset generate more *Total Categories*, which indicates that FuzzyCut has a particular effect on participants while categorizing quantitative values because the number of categories increases significantly while using FuzzyCut, which verifies **H1**. It also indicates that participants can understand the data deeply after using the technique. Besides, compared with the first-time categorizations using FuzzyCut, we find that the second-time categorization generally has more counts in *Total Category* and *Full Categories*, meaning that users create more categories in the second-time categorization while using the entire datasets. Therefore, the data size affects the categorization results, which verifies **H2**.

Now, we analyze the categories naming patterns. We count usage frequency of all the words (*e.g., very*, *old* and *hot*) while operating FuzzyCut. Also, we code qualitative properties of words, *e.g., Old* and *Children* referring to common words; *Very* and *Less* referring to comparison-descriptive words. Figure 3.8 shows statistical results where the words *Hot*, *Children*, and *Old* show more frequently than other words, indicating that the participants usually prefer to use common words as labels for new categories. Besides, the words *Very*, *Large*, *Small*, and *Middle* also appear more frequently, from which we can conclude that the participants usually prefer to use comparison-descriptive words combined with nouns to represent the categories' degree. These two findings verify **H3**.

The post-study questionnaire collected qualitative feedback from the participants, and the results are shown in Figure 3.9. The x-axis represents the six questions related to the performance of FuzzyCut, and the y-axis represents participants' scores for each question. The score is Likert scale from 1 to 5, which ranges from very negative to very positive. Most of the scores are more than 2. The question related to *Easy to interact* has the most positive feedback from the participants, reflecting that the participants can easily engage with the interactive functions. However, the questions related to *Creating categories* and *Parameters understanding* have relatively lower performances than other questions, which might be because these participants are unfamiliar with the fuzzy theory, so it takes time to understand the membership function parameters.

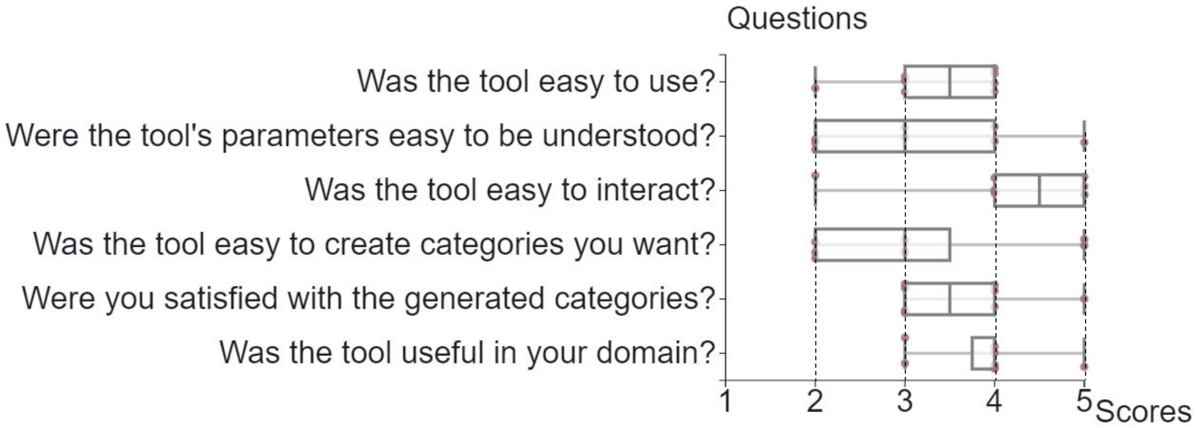


Figure 3.9: The results of the post-study questionnaire. The box plot represents a Likert scale for eight participants answering six questions. The scores are from 1 to 5, with 5 representing the most positive feedback and 1 representing the most negative feedback.

3.6.3 Other Feedback

We have also collected all participants' qualitative feedback while using FuzzyCut. In general, the feedback is positive. All participants think FuzzyCut is easy to understand and manipulate. Two of them think this technique is very useful in many domains, especially those that do not have a clear mathematics relation. In that case, people make the decisions with their own experience, so an interactive technique is highly necessary to help people understand the fuzzy relations among data.

Moreover, there are issues with the current version. A participant thinks it would be better if the membership degree could connect with the distribution of the dataset. In addition, there are issues while adjusting the membership function shape, such as category names overlapping when there are many categories. In terms of parameters, one participant says they cannot easily check the values of *core* and *support*.

3.7 Conclusion and Perspectives

We proposed an interactive membership function (FuzzyCut), which allows users to comprehend the quantitative data by mapping them into categories. It creates fuzzy categories combining membership degrees to convey how confident a quantity belongs to a set. We illustrated its use with taxi speed data. The evaluation results show that it supports users with understanding the quantities in an interactive way where they can observe how the categories change when they create different membership function shapes. The data size also affects users making decisions in the categorization process. This work opens research areas at the intersection of visualization and fuzzy logic, which is currently under-explored. The interactive membership function can help traffic experts create a suitable scale of traffic flows or densities.

We expect this technique to be used as a categories-generation tool in traffic-relevant data and other domains (*e.g.,* the temperature and age data). In particular, as we have released our prototype and code as an open-source project, it facilitates further application to more domains, improvement, and evaluation. We will discuss how to deploy FuzzyCut and other visualization techniques (which will be introduced in the following chapters) in traffic control centers. We will also discuss the applications in other domains in Section 6.2.1.

Chapter 4
Spatial Visualization

Contents

In this chapter, any use of ”we” in this chapter refers to Liqun Liu, Romain Vuillemot, Philippe Rivière, Jeremy Boy and Aurélien Tabard. In this work, we have a paper under review as follow:

- Liqun Liu, Romain Vuillemot, Philippe Rivière, Jeremy Boy and Aurélien Tabard. ”Generalizing OD-Maps to Explore Multi-Dimensional Geo-Coded Datasets,” In *The Cartographic Journal*, P. 26, 2022. (Under review, Link)

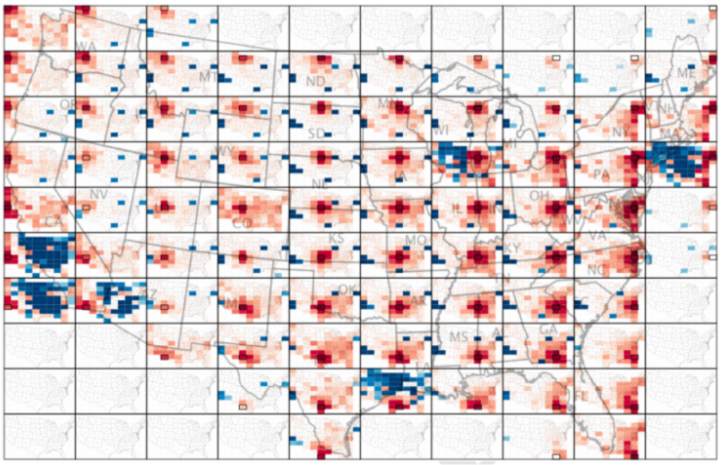


Figure 4.1: OD map (which stands for Origin-Destination map, a geospatial visualization technique introduced in [29]) represents the US migration among countries. This chapter is based on this technique that we generalize, to explore spatial data.

4.1 Context and Motivation

Road traffic is inherently spatial, thus mobile entities produce massive amounts of geo-coded data through GPS signals, mobile phones, or sensors (introduced in Section 2.1). Analysts can use them to understand patterns of human mobility or road traffic. Such data usually enable the analysis of both entities (*e.g.,* taxis or public transport) and their *relationships* either with themselves (if entities move over time) or across them (if entities communicate with each other). Such relationship data is often called Origin-Destination (OD) data: it refers to the movement of objects in a geographic space from one location to another, **O** refers to the original position, and **D** refers to destination position, and is deeply geo-coded. In this chapter, we explore OD data in both [156] *explicit* and *implicit* relations, where:

- *explicit* relations refer to the spatial trajectories of links between geo-coded entities (*e.g.,* taxi trajectories);
- *implicit* relations refer to the abstract attributes of those trajectories (*e.g.,* speed or moving direction of taxis).

Exploring OD data requires ways to rapidly navigate through implicit relations while preserving references to explicit relations. OD *Map* (Figure 4.1) is a geospatial visualization technique proposed by Wood et al. [29], to encode the explicit relation of origin and destination of entities. A first level of the map encodes the origin, and a second nested level encodes the destination in cells nested on the map. Visually, this technique generates grids of maps that preserve the original map at a lower scale. This technique has proven efficient in analyzing any geo-coded data in many domains. Our goal in this chapter is to generalize it to encode implicit relationships, *i.e.* showing relations between entities in the spatial dimension and eventually help transport planners (introduced in Section 2.3.1) explore the OD patterns, which is one typical task of situation-aware exploration and prediction (introduced in Section 2.3.2).

To address this problem in an operational manner, we introduce a generalization of the OD maps techniques called GRIDIFY, as shown in Figure 4.2. We implement it as an interactive Exploratory Data Analysis (EDA) tool for geo-coded data—the approach of analyzing datasets by summarizing the data characteristics with visualization methods [36]. We demonstrate the expressiveness and effectiveness of GRIDIFY through several case studies, and we show how it enables the discovery of structural properties of typical datasets — taxi and transit datasets, which are introduced in Section 2.1.5 as taxi taking passengers trajectory dataset and transit dataset. We then discuss the main limitations and challenges of the generalization framework and the tool, primarily related to data pre-processing and scalability.

We have implemented GRIDIFY using Observable Notebook (introduced in Section 3.1). We used the D3 [136] version 5.9.2 library for data manipulation, and in particular the `d3.group` function for calculating nesting. We also used the `d3-gridding` toolkit [157]–based on D3– for grid partitions. All supplemental materials are in Table 4.1.

Name	Link
Online prototype	`https://observablehq.com/d/10a0f8527c21dcd3`
Datasets	`https://github.com/LyonDataViz/oddata`

Table 4.1: Supplementary materials. It lists an online prototype of GRIDIFY implemented with Observable notebook and datasets description (data format and how to use one's datasets).

4.2 Related Works

Our focus is primarily on developing an Exploratory Data Analysis tool that relies upon the general concept of OD matrix. OD matrix is a description of movement in a certain area, with rows referring to origins and columns referring to destinations, which has been investigated by [158, 159, 160, 161]. As we aim to generalize OD matrix visually, we investigate grid-based and faceted approaches to visualize the trajectories of different geo-coded entities and their abstract attributes. We consequently review previous work on faceted visualizations and small multiples, as well as work on constructive Exploratory Data Analysis tools developed by the Infovis community.

4.2.1 Faceted Visualizations and Small Multiples

A recurring theme in Exploratory Data Analysis is the necessity for multiple perspectives on *faceted views* of given datasets—something Tufte emphasizes in his praise of small multiples as a very efficient exploration mechanism [119]. Munzner [162] outlines a design space for multiple faceted views, which covers the use of small juxtaposed visualizations to compare individual *perspectives* (*e.g.,* abstract attributes) across many *collections* (*e.g.,* small multiple views of different entities) in the data, or multiple perspectives across few collections. Beecham *et al.* [163] expands this design space by proposing a theoretical framework and an implementation of *faceted views with varying emphasis*, which attempts to tackle the issue of simultaneously visualizing multiple perspectives across many collections. Our work builds

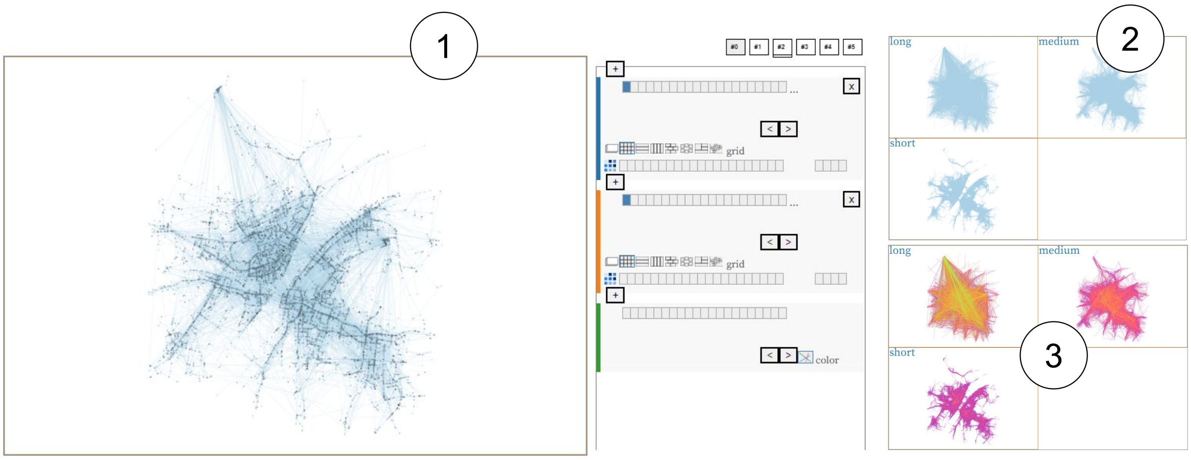


Figure 4.2: Example of visualizations built with GRIDIFY. GRIDIFY relies on the combination of simple data grouping, aggregation, and grid patterns, to reveal implicit relationships in geo-data (*e.g.,* speed of taxis) while keeping explicit ones (*i.e.* positions of taxis) visible. ① shows the taxi trajectories (explicit relationships) between pick-up points (the places where taxis pick up passengers) and drop-off points (the places where taxis drop off passengers). ② extends the same encoding as ① to nest more dimensions, such as the distance of the trajectories (implicit relationships). ③ renders the trajectories based on the trajectories' attributes.

on this idea. However, where Beecham *et al.*'s work explores superimposing perspectives on each collection, ours breaks down the collection, *e.g.,* the dense, spatially anchored trajectories of a geo-entity, according to the different perspectives, using a variety of grid patterns [157]. As such, our work is also related to the use of small multiples as a navigation mechanism in multivariate datasets [164] to separate subsets of the data using non-visual properties.

Google Facets[1], or the Geofacets R module[2], tie together facets and (shallow) hierarchical layouts by organizing the different views of the data in a specific spatial layout, or *grid*. However, these techniques only allow the display of a limited number of abstract attributes since they do not support deep hierarchical nesting within facets or *cells* of the grid. Matrix-based layouts have shown many benefits to organizing entities into rows and columns [165]. PivotGraph [166] builds on 1D or 2D matrix-like grids to group nodes of multivariate graphs and let users analyze node's properties. Pivot slices [156] demonstrate how a faceted approach with multiple heterogeneous views could support the exploratory analysis process.

Finally, to better qualify the distinction we make between the inherent geo-coded attributes of OD data and their more abstract attributes, we rely on Zhao *et al.*'s [156] distinction between implicit and explicit relations in datasets. While the distinction they make is not necessarily intended for primarily geo-coded data, we use the idea of explicit relations to qualify the spatially anchored aspects of geo-coded entities, and implicit relations to qualify their more abstract attributes.

[1] https://pair-code.github.io/facets/
[2] https://github.com/hafen/geofacet

4.2.2 Constructive Exploratory Data Analysis Tools

While the underlying motivation for faceted views is well known—to enable rapid scanning over items or attributes using descriptive visualizations–the various ways to achieve this is grounded in different strategies. Interaction is key. However, analysis attention is often split between data operations (*i.e.* section, queries) and visual mapping operations (*e.g.,* graphical encoding, choice of chart). While this offers much flexibility, the two operations are overloads and distractions. As Tufte suggested, exploratory processes should bear on *data variations*, rather than on *design variations* [119]. We subscribe to Tufte's view and set graphical encodings to position—the most efficient encoding [167]. Our approach relies on a top-down, gridded approach to create facets: the display space is first divided according to the values of a first, specified implicit relation (*e.g.,* time or any categorical values) into a gird of cells, each showing a relevant subset of the explicit relation, and this procedure is then repeated (up to) as many times as there are implicit relations in the data.

This progressive construction of the visual display alongside the Exploratory Data Analysis process has the potential to facilitate sensemaking activities [168]. Using a code-driven approach, construction can build upon rich visualizations grammars, like Vega [169], Grammar of Graphics [170], or ATOM [168]. These toolkits enable the creation of sophisticated visualizations, but they can be tedious to specify, and they provide limited feedback. However, these code-driven specifications also lend themselves to constructive interfaces. Tableau, building on Polaris [171], maybe the most prominent one. These tools enable analysts to rapidly create and explore multi-dimensional data. More recently, Voyager [172], building on the Vega-lite grammar [173], provides a demonstration of how construction mechanisms can be used to create multiple views and can be augmented with automated design recommendations.

4.2.3 OD Visualization

OD (Origin-Destination) visualization is becoming a popular researching topic [31, 76]. Graph [174] is widely used in visualizing the traffic flows, which represents the positions of the geo-coded entities with nodes and relations between every two geo-coded entities with edges. However, the graph limits visualizing a large number of data items since it could cause the serve clutter. In order to avoid the overplot, Holten *et al.* [174] bundle the node-links if they have similar patterns and the results showed significant clutter reduction and visible high-level edge patterns. Similar to the graph visualization, the flow map [175] is another method for visualizing the OD data by arrows or bands between places to present the from-to information. Given that the data is rising fast, visualizing OD data with a flow map cannot avoid the clutter. There are some improved flow maps created to reduce the clutter impact, such as Zhu *et al.* [176] presents density-based flow map generalization method to extract similar OD patterns.

4.3 Gridify Framework: Decoupling Data and Visual Abstractions

The core of our generalization of OD maps is decoupling data transformation and visualization construction operations to answer more flexibly Exploratory Data Analysis questions related to spatial analysis problems (*e.g.,* how do humans commute in a city?) The InfoVis pipeline generally implies following a data-to-display analytic approach [157]. We argue that exploratory processes can benefit from going back-and-forth between display-to-data *and* data-to-display approaches to refine and answer sophisticated questions [177]; and that ideas on how to best transform the data to answer given questions can be formed at a purely visual level, by manipulating and progressively constructing the elements of the display. Essentially, this means that analysts should be able to break down the explicit relations in OD data either by grouping subsets of the explicit relations in data space according to specified implicit relations, or by isolating aspects of the explicit relations directly in the visual space.

We consider the Infovis pipeline as the combination of two abstraction levels: a *data abstraction* level, and a *visual abstraction* level. We use *data abstraction* as an umbrella term for all the manipulations and transformations an analyst can perform in the data space—typically to transform large datasets into (often smaller) sets with derived attributes—before trying to map them to the graphical space; and *visual abstraction* as an umbrella term for all the manipulations and constructions an analyst can perform in the graphical space, before trying infer the necessary transformations in the data space. In this section, we detail the specific data and visual abstractions we propose in Gridify, and we discuss how they can be joined through nesting operations.

4.3.1 Data Abstractions

We consider each geo-entity as the triplet $E = \langle d, t, A \rangle$ with $d_i = (x_i, y_i)$ their explicit relation (a position in a 2D space S), ordered over time $t_1 < t_i < t_n, t \in T$ a time period, and a set of implicit relations A (*e.g.,* speed of taxi). Connections are two entities connected with each others $C = \langle (E_o, E_d), A' \rangle$, in most case they represent the endpoints of a trajectory, which also contain implicit relations A' (*e.g.,* distance between entities). Those two sets of implicit relation, A and A', are our focus of attention as they are responsible for the main data abstractions in Gridify, as shown in Figure 4.3. A and A' are composed of:

- **A dimensions list** that consists of all the implicit relations available for entities and connections $D = \{A_1, A_2, .., A_n\}, A_i \in \{A, A'\}$. Some dimensions are available permanently (*static dimensions* like weather, time), while others are only available during, or after given data transformations (*dynamic dimensions* like speed or acceleration).
- **Grouping methods** that group values, for a dimension, based on criteria (*e.g.,* number of expected groups) or a property of the dimension: *Ex: grouped by values (categories), bins or buckets (quantities).*
- **Aggregation methods** that derive one or multiple values from the grouped group: *Ex: count, sum, mean, median, average, distinct, min/max, unique, and value.*

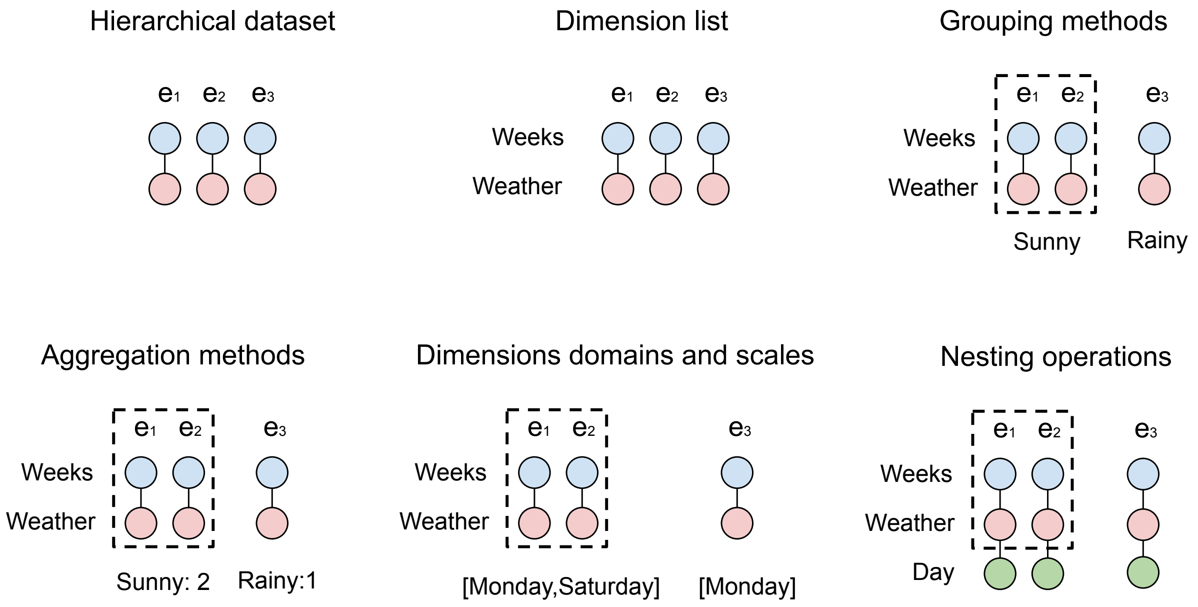


Figure 4.3: Data abstraction in GRIDIFY. Dimension list consists of two implicit relations (Weeks and Weather). Grouping the geo-coded entities (*e.g.,* e_1, e_2, and e_3) based on the *Weather* dimension achieves two groups. One group contains e_1 and e_2 that only has *Sunny* value. In contrast, another group contains e_3 that only has *Rainy* value. Aggregation method estimates the statistic of grouped groups (*e.g.,* one group has two elements and another has one element). Dimension domain method shows the value scales from other dimensions (*e.g.,* one group has a domain in $[Monday, Saturday]$). Nesting operations generate another level nest (*e.g.,* adding the *Day* dimension).

- **Dimensions domains and scales** that return a list of all unique values for a given implicit relation A: $\{a1, a2, ..., am\}, a_i \in Dom(A_i), i \in [1, m]\}$ or the extent of a scale $[Min(a_i), Max(a_i)], i \in [1, m]$ for a quantitative dimension. *Ex: Dom(Years) = 2011 or 2012.*
- **Nesting operations** that consecutively pass down operations from one level to the next one. *Ex: Select a* `Country` ⋈ *Divide by* `Year` ⋈ *sum(*`Exports`*).*

Grouping and aggregation methods provide an array of options [178, 179, 157] well documented in *e.g.,* [180], and nesting operations are central, as they enable increasing the number of implicit relations A and A' in the visualization.

4.3.2 Visual Abstractions

The main visual abstractions in GRIDIFY are space partitions we call *cells*, which are organized according to *grid patterns*. Cells respect a 1:1 mapping with the underlying data abstractions: there are as many cells in a grid as there are groups in a nested data partition. As such, cells and grids also follow a nested, hierarchical structure, starting at the canvas (or *root*) level, and progressively breaking down each cell into smaller partitions (or *leaves*). We use the following notation (originally proposed in [157]) to describe grid properties:

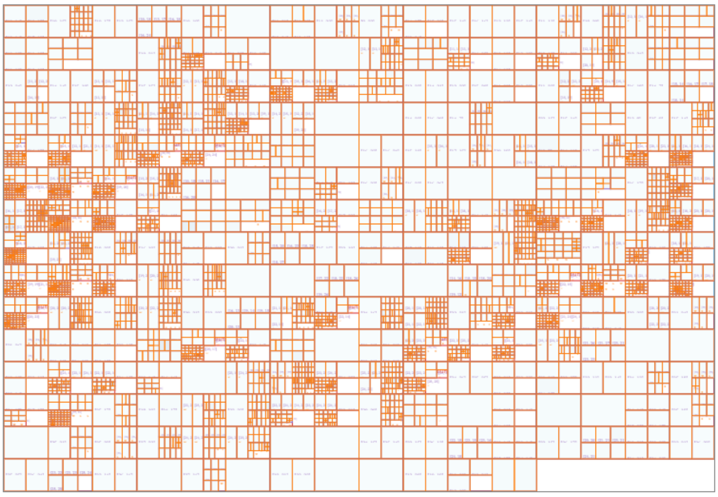


Figure 4.4: Cells become smaller and smaller when the number of implicit relations A and A' increases.

- **Grid patterns** are methods defines the space division method that will create **cells** which are a space partition with coordinates and dimensions.
  Parameters: grid ▦, *horizontal* ▤, *vertical* ▥, *coordinates* ⁘, *treemap* ◫, central ▢, radial ⁙, brick ▤.
- **Visual mapping** is the customization of cells using an attribute mapped to its properties, based on the grid type.
  Parameters: coordinates, height, width, order, color, filling with marks such as circles and lines.
- **Nesting** represent all grids can be nested with each others and children inherit from the parent's placeholders positions and dimensions.
  Ex: ⁘ $\bowtie$ ▦ *(scatterplot of matrices).*

Cells are bound to become smaller and smaller, as the number of implicit relations A and A' increases, as shown in Figure 4.4. They will converge towards mark-like representations, if represented as an empty rectangle, or glyphs [181], if elements are visible, or a special encoding is being used.

We advocate for this decoupling of abstractions to be more widely supported by interactive techniques. As far as we know, there are currently no tools that allow operating on both directions of the Infovis pipeline at an abstract level, in particular by providing grid patterns and advanced nesting capabilities for deep hierarchical layouts (a fundamental operation at both the data and visual levels). The closest approach to generically unifying these abstractions is graphical notations like [177, 173, 157].

4.4 Gridify Tool

Our Gridify framework is implemented available as an online web application (link) designed to help analysts build abstract, multidimensional gridded generalization of OD maps. It presents all the parameters for building the necessary data and visual abstractions on a compact *query panel* (Figure 4.6-left), which analysts can use to rapidly explore implicit relations in the data, while preserving references to their explicit relations in a *main view* (Figure 4.6-right). In this section, we describe the design principles we adopted for the query panel and the main view. We describe their implementation, as well as the development of simple cues for navigating between pre-set and historical configurations. Finally, we present a typical workflow using GRIDIFY.

4.4.1 Typical Workflow

Figure 4.6 shows a simple workflow for a user starting with an empty query sequence. One can create the main view divisions and progressively refine them in a few steps, by:

① *Browsing* all the **dimensions** of a dataset and *picking one* which defines a **grouping**. For each element of the **domain** cells are created on the main view;

② *Customizing grouping and grids patterns* consists in defining the dimension domain, unique values (if categorical dimension) or its bins (if quantitative dimension). The cell will change **position** and **dimensions** based on this gridding pattern;

③ *Nesting* by repeating the sequence on another dimension. This enables the iterative construction of complex queries and grids by leveraging **dependencies** that occur since they are chained together.

In order to support fast exploration, all the elements can hover with an immediate update in the visual space. The states are transient and only made persistent on click. This scrubbing-like interaction is an instance of sequential feedforward, enabling instant preview and supporting the reflective construction on complex queries. We use the widget's color and position to convey the query's state. When scanning the complete list of widgets used for the nesting, one can read the transformation and the resulting query sequentially.

4.4.2 Query Panel Design

The query panel (Figure 4.6-center) enables querying the implicit relations of the data, and associates a grid pattern with each query. Its design builds on common EDA tool design principles (see *e.g.,* [172, 119, 177]), which derive and summarize as follows:

P1 Make nesting chainable, and show them in a compact way: the query panel should facilitate building sequential queries that can be chained and viewed together, as well as allow for their tweaking and reorganization, to help analysts better understand the state of their query.

P2 Expose the data and visual abstractions: all data dimensions, grouping and aggregation parameters as well as the visual abstraction parameters should be readily visible in the query panel, and not hidden in menus or further folded panels, immediately exposing the breadth of options offered to the analyst.

P3 Enable constructive selection and constant preview: the query panel should enable a tight action–feedback loop, and the main mode of interaction should favor continuous actions (e.g. hovering or brushing) over sequential ones (e.g. dropdown selection) providing instant feedback.

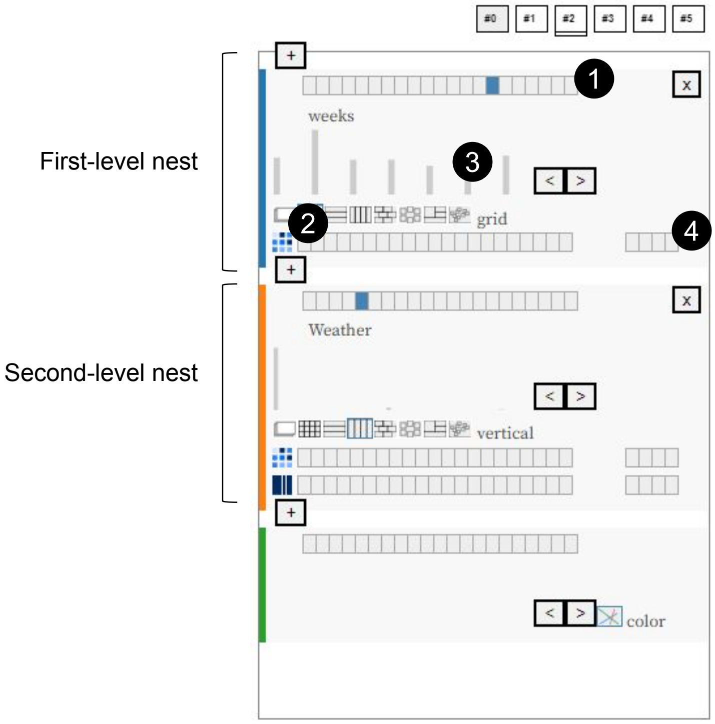


Figure 4.5: Query panel overview. ① shows the rectangular boxes view all the implicit relations. Grid pattern selection generates different space divisions (*e.g.,* grid, horizontal, or vertical) in ②. The histogram displays the univariate distribution of each quantitative implicit relations' value in ③, and ④ displays options of quantitative implicit relations calculation.

P4 Ensure generic and independent manipulation of the data and visual abstractions: analysts should be able to query implicit relations of the data independently from their visual abstraction, and reciprocally, they should be able to build visual mappings even if the necessary data abstractions are not defined.

The query panel is composed of rows controlling each level of nesting (shown in Figure 4.5), both in the data and visual abstractions (**P1**), which display a list of all implicit relations (shown as small rectangular boxes in ①) and of grid patterns that can be used to encode them visually (shown as ②). This ensures that all data and visual attributes, operations, and parameters are immediately exposed (**P2**). Selecting a grid pattern, and simply brushing across the list of implicit relations will automatically update the main view (**P3**), breaking down the base OD node-link diagram (*i.e.* the visualization of explicit relations) into a multitude of cells, related to the selected implicit relation.

To enable Grouping methods in the data abstraction, we display a univariate distribution of each quantitative implicit relations' value (in the form of a histogram shown as ③), as well as an option for changing bins values. Note however that the Aggregation operations can only be applied to quantitative implicit relations (shown as ④). This encoding is relatively compact (**P1**), so many nested operations can be listed in the panel.

There is no pre-defined flow for query construction. The user can start with the visual abstraction and then continue with the data abstraction (**P4**) or vice-versa. The operations done at a nesting level (*e.g.,* first-level nest and second-level nest) are chained with the levels preceding it (**P1**). As a result, the query panel plays an essential role in the EDA process (beyond exposing data abstractions): it indicates all current implicit relation selections and their visual mappings, and these can be dynamically updated, like in an (interactive) legend [182]. This means there is no need for additional navigation elements in the interface.

4.4.3 Main View Design

The main view consists of a canvas on which the explicit relations, and all their consecutive subdivisions are rendered. Its design follows one main principle, summarized as follows:

P5 Maintain consistent encoding: A consistent encoding should be preserved (*e.g.,* using position) to allow a focus on attributes' structure and generate interesting sub-sets. Also, browsing should be informative on the attributes space and domains and provide flexibility, especially when binning or clustering is needed.

The canvas initially displays the explicit relations. Once a query is added using the query panel, the canvas turns into a hierarchical structure, in which the visualization of explicit relations is broken down into cells, each containing a subset of the explicit relations, relevant to the selected implicit relation. Additional encodings can be set, such as a bivariate color scale [183] to convey **aggregation** values (*e.g.,* `MEAN`, or `COUNT`). Color encodings of the whole cell can also be used as the level of nesting increases, for performance reasons to reduce the number of elements to draw.

Analysts can then select cells for grouping purposes by clicking and dragging in the main view. Selecting cells updates a `SELECTED` attribute in the dataset. This attribute can be used to 1) show only the selected values, 2) hide them, or 3) group purposes in the query pipeline, *e.g.,* computing aggregate values on the selection or adding a level of nesting.

4.5 Case Studies

In this section, we demonstrate the use of Gridify using real-world, geo-coded datasets. It has been tested with up to 14 datasets, and we report on two traffic-relevant datasets — taxi trajectory datasets and public transport datasets (introduced in **Section 2.1.5**), to showcase the expressiveness of Gridify, and to identify interesting exploration patterns and shortcomings of our design and implementation. We focus on these specific case studies for the diversity of their explicit and implicit connections and because authors and their collaborators have expertise in the domains studied and have previous experience creating visualizations with the data.

Taxi trajectory datasets. The first dataset contains large-scale taxi GPS records, which are now publicly available such as the one collected by the city of Wuhan, China, which a co-author of the work already pre-processed (introduced in Section 2.1.5). It contains 7271 entities over a month (Sep. 2013, $145,789$ trips). Trajectories for each taxi are available as

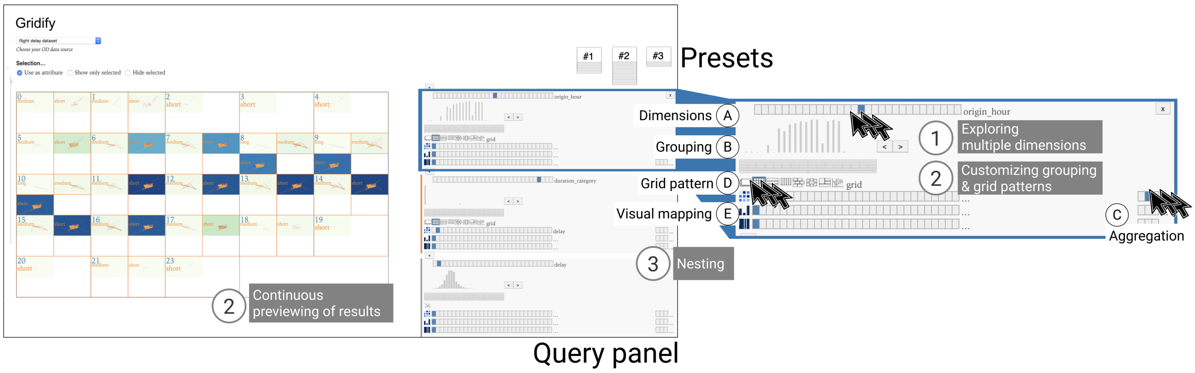


Figure 4.6: An overview of Gridify implementations and interactions to construct data and visual abstraction (a) **dimensions lists** from which (b) a **grouping method** such by domain (if category) or by binning (if quantity), (c) **aggregation types** for each division such as counting or averaging, (d) **grid patterns** and (e) **visual mapping** for grids customization. The left part shows the gridded space where each cell encodes **dimensions domains and scales** according to **grid types** and **visual mappings**. Widgets can be vertically chained to create **nesting** for both the data and visual abstractions: the widget will inherit from **dynamic dimensions**, and the new grids will be created in the **placeholders** created by the parent.

recorded at a frequency of $1-4$ times per minute, with unique ids and a STATUS occupied/vacant/not working/invalid along with fare value when occupied.

Figure 4.7 (0) shows the overview of taxi trajectories while taking passengers. An interesting region is highlighted — the airport, which displays long segments while most of the city usually has short trips. An attribute in the dataset flagged the trajectories that corresponded to *going back to pick up point* during a weekday and Figure 4.7 (1) displays them by hour, showing that first there is a peak of trajectories at midnight, and second overall most going back trajectories are short. Figure 4.7 (2) displays asymmetry between pickups and drop-off for a region of interest during a month: Wuhan's railway station. Visually, it is not possible to spot significant asymmetry between those two, but showing pickups and drop-offs side by side, by days and time slots reveals the asymmetry of pickups and dropoffs Figure 4.7 (3). The trade-off here is that aggregation provides better comparison but without the trips' origins or destinations.

This exploration enables to better grasp the dataset dimensions distribution, a preliminary step to then develop complex machine learning models to predict hotspot that maximize incomes, which is a research focus of one of the authors of the chapter.

Public transport accessibility. The second dataset contains transit data, which are heavily used today for trip planning, but also by urbanists and decision-makers to understand how well transit networks serve the population. We have collected a dataset of 45,520 trips in Paris (introduced in Section 2.1.5) at every hour of a given day Figure 4.8 (0). The trips start from 3 distinct locations (origins) and destinations are all reachable areas surrounding the origin for a given time (*e.g.*, 5min) by mode of transport (walk or public transport). We collected various dimensions such as CO_2 emissions.

The first step is to plot the data in a familiar way when analyzing locations accessibility:

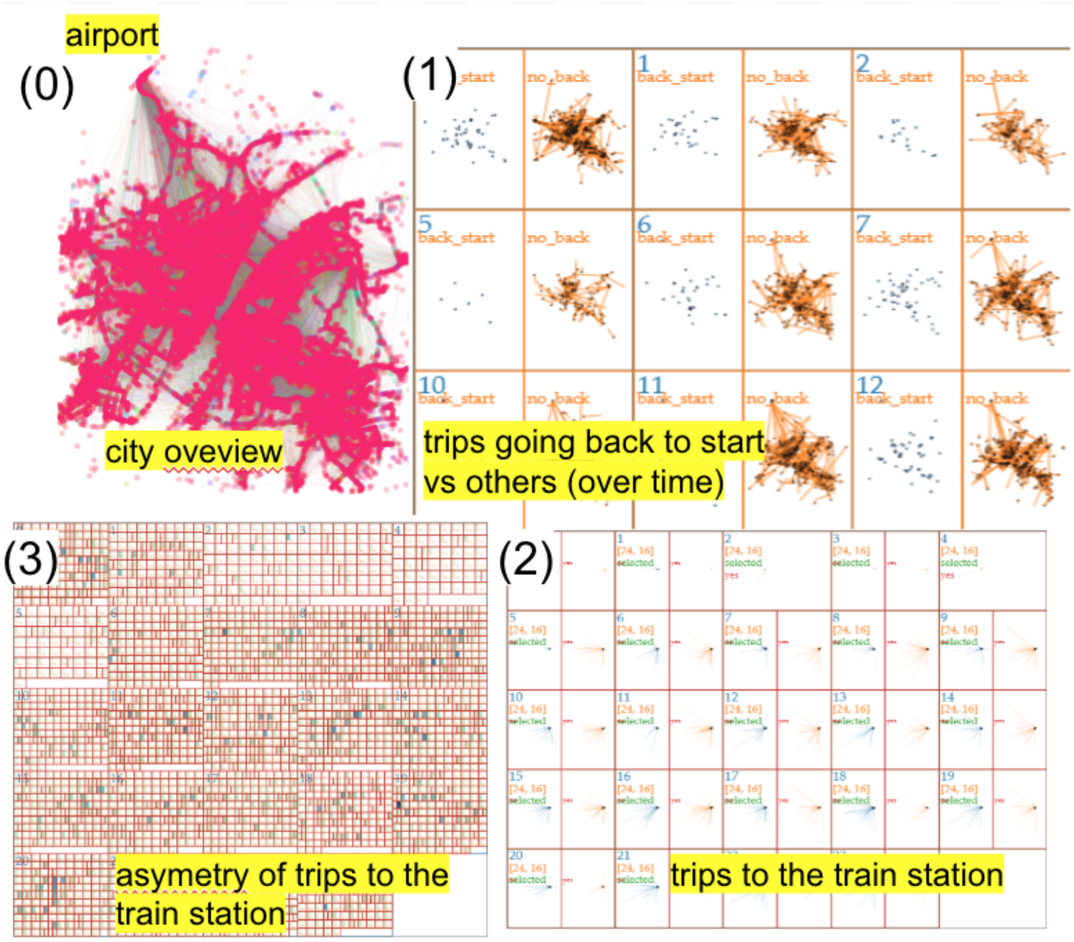

airport
(0)
city oveview
(1)
trips going back to start vs others (over time)
back_start
no_back
(3)
asymetry of trips to the train station
(2)
trips to the train station
[24, 16]
selected

Figure 4.7: Taxi dataset of 7, 271 taxi drivers in a Chinese city during one week (145, 789) records. We identify interesting patterns related to the asymmetry of trips. (0) refers to the overview of the taxi trajectories while taking passengers and we bin them based on hours in (1). Then, we display an interesting asymmetry between pickups and drop-off for a region of interest (Wuhan railway station) during a month in (2). Finally, we show pickups and drop-offs side by side, by days and time slots reveal the asymmetry of pickups and drop-offs in (3).

as an *isochrone maps* Figure 4.8 (0, 1). Isochrones enable to grasp areas reachable for a given time budget, *e.g.,*with a 5 or 10min walk. However isochrones are complex to build as they usually require an underlying graph data structures along with efficient breadth-first search algorithms. Using Gridify, isochrones can be built by filtering segments originating from a given location and by changing the radius of the destination mark (circle) to emphasize them, which is close to how isochrones are built.

We then abstract space using several grids to understand the temporal patterns. Figure 4.8 (2) groups each of the locations as a row and a vertical grid divides by hour. The central part of the grids shows a (small) isochrone for every hour. The background colors the SUM of trips so we can quickly spot the time of the day the public transport network is most active (during morning and afternoon rush hours), but also the lack of rapid transit mechanism during

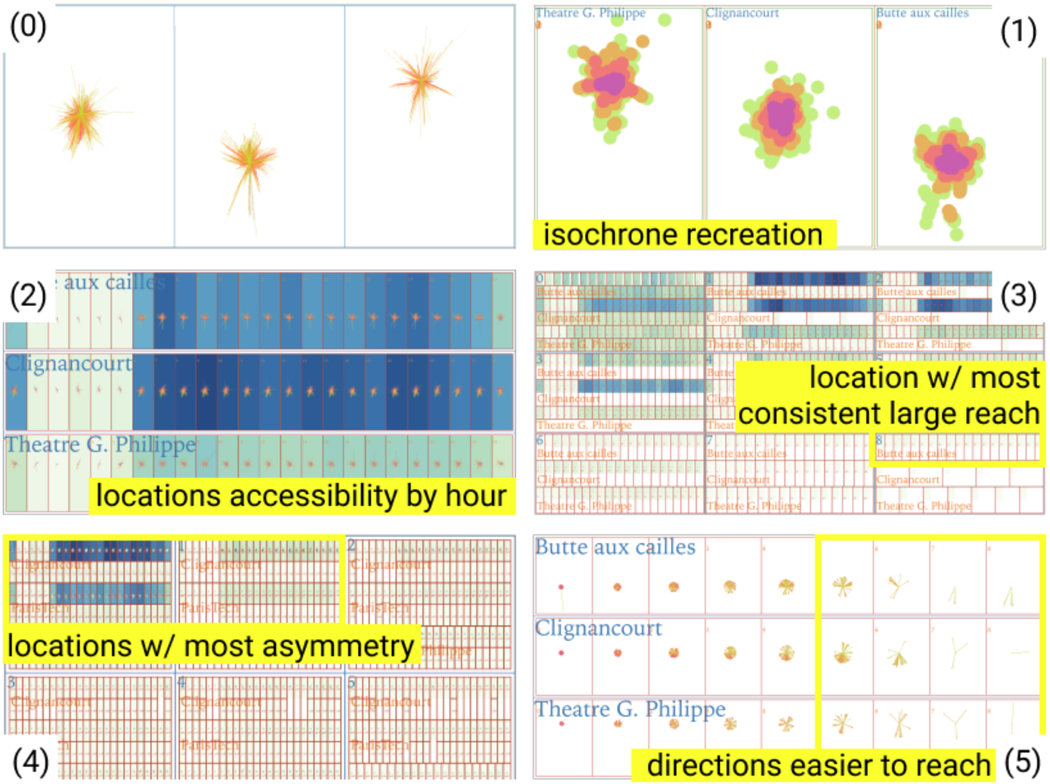


Figure 4.8: Comparing public transport accessibility for three locations throughout the day. An overview of trips divides journeys into three grids based on origin locations in (0) and (1). After that, grouping the origin locations as rows and dividing hours as vertical grids create the *location accessibility* by hours in (2). We divide again by distance to create the *most consistent large reach* in (3) and by the time difference at the root to create the *most asymmetry* in (4). Finally, we divide the grids by length of journeys and origin locations to receive *easier-to-reach information* in (5).

the night. We can refine the query by removing the walked journeys (another division), and dividing again by distance, this leads to locations having the best and most consistent large reach (something not conveyed by isochrone maps) Figure 4.8 (3). By modifying two steps of the pipeline from Figure 4.8 (3), we compare the duration of outgoing and incoming journeys (another division), and identify which location has the most asymmetry (we add division on the time difference at the root using BINNING. It is a way of grouping method, and used when the value is quantitative, as shown in Figure 4.6 (b)) in Figure 4.8 (4). Finally, starting from scratch again, partitioning the trajectories by the length of journeys (distance) and origins, and coloring the marks by journey duration, reveals the directions that are easier to reach than others, and highlights the difference between distance and travel time Figure 4.8 (5).

4.6 Discussion

Our case studies indicate that the current implementation of Gridify enables complex EDA using a simple set of data and visual abstractions. We discuss here the implications of our abstractions, their implementation in Gridify, and the challenges it raises.

4.6.1 Expressiveness and Applicability

Gridify is already a very expressive tool despite encoding simple grouping and aggregations mechanisms and a consistent visual encoding (using circle/marks or color for leaves, and position otherwise as we follow **P5**).

Gridify could handle more data and visual abstractions, such as partitioning, aggregation, and grid patterns. Data aggregation can be integrated on the fly during analysis in the Observable Notebook by writing functions in an advanced mode. Grid partitions rely on an open-source toolkit–`d3-gridding` [157]–that offers a modular approach to easily add new **grid patterns**. However, in our current approach, grids should have a recursive construction mechanism. Hexagonal grids could be an improvement [184] as they sometimes provide a better binning estimation, but Gridify only supports rectangular cells. Finally, Gridify is compliant to include Vega [173] specifications at the leaves nodes to render aggregation charts (instead of the marks or the colors). Design implications on the query panel are related to adding chart templates either by the queries or as pre-set configurations.

So far, we only encoded leaves using color for aggregation or circles and rectangles for explicit connections. Lines that have a rich design space [185] and curve design [186] could be used, *e.g.,* to encode local connections properties such as speed for finer-grain representations. This would extend the **visual mapping** section of the query panel and would probably require a specific legend. Those could be useful at an occupation stage, but for exploratory tasks, one would recommend adding better dynamic opacity techniques [187].

Gridify can be applied to any type of geo-coded data beyond simple connections. For instance, geo-trajectories can be re-constructed, grouped by `TRAJECTORY_ID` and ordered over time using the current version of Gridify (see taxi case studies in Section 4.5). New aggregations will be needed to calculate distances and other trajectories properties [188] dynamically. Figure 4.7 already shows some cases where a grouping of trajectories can benefit from Gridify. However, this approach limits the number of segments to display a full resolution trajectory. Also, some metrics on trajectories are relative to the sequence of segments (*e.g.,* sliding window speed) and cannot be derived using aggregation.

4.6.2 Scalability

Gridify's scalability in the number of items is limited by the number of cells and marks it can draw simultaneously. Our prototype handle up to 100k data points (marks) and thousands of rectangles/placeholders. This limit is set by the Observable reactive framework we picked, as it facilitates prototyping and re-use of visualization libraries. Switching to GPU rendering is a classical step that would help but would be limited at some point.

We argue most promising approaches to tackling scalability issues are related to strategies (*e.g.,* domain aggregations and marks aggregation.) to first display aggregation of cells for immediate feedback, and then progressively render details such as nested cells and marks: only the statistical properties of dimensions need to be known in advance (*e.g.,* distribution)—the data points can be loaded later. Regarding the scalability in the number of dimensions, a first limit is on the display of implicit relation rectangles and nesting chaining: it has been tested with up to 20 implicit relations, but beyond a more compact design should be used. For datasets with thousands of implicit relations, adaptive exploration strategies should be developed to suggest/re-order these relations according to the current view (similarly to what is done in EvoGraphDice [189]. Dimensions reduction techniques could also be used as an **aggregation** method.

4.6.3 Perspectives

Our future work is oriented toward matching modern exploratory analysis features available in the community, introducing multiple views and animated transitions.

Multiple views. Instead of the single view, exploratory analysis often requires multiple coordinated views (MCV). We have studied the MCV in Section 2.4.5, which could provide the context of the exploratory analysis. Such context is helpful to provide a dataset overview constantly available, *e.g.,* all countries, while a specific country is selected (instead of filtering out all other countries). At the moment, the exploration strategies match the *small multiple, large single* [164] approach, where each view is followed by small multiples that provide navigation options.

As Gridify technically and conceptually relies on [157], multiple static views can be defined in a data-driven manner. So adding a 2-view, Focus+Context layout (introduced in Section 2.4.5) would require two steps: first at some point to inject a data array with those views properties (which are cells); the linking (shared selection between views) would be provided by the SELECTED attribute. And then branching the abstractions based on those views (*e.g.,* to assign particular abstractions to the static cells). The impact on the query view would be important as it will not be linear anymore. Techniques like ElasticHierarchies [190], VisBricks [191], TPFlow [192] and Baobabview [193] are design candidates to improve the sequence with branching. Recently the use of Virtual Reality [194] has been proposed to explore Origin-Destinations—but with explicit relations encoded as lines—offering brand new spaces to further explore using a grid-based approach.

Animated transitions. Entities (and their connections) have a continuous representation in Gridify. They can be animated when cells change positions. Similarly, the transition between grid patterns [31] provides benefits to users, even though sometimes cells change shape. However, in most cases, many parameters change at once: attribute, grid pattern, and aggregation method. Cells may then go through multiple states: appear, disappear, update (*i.e.* change position and shape). Communicating the change of the cell is an open challenge as similar grids or cells (position and size) do not necessarily encode similar data. A simple approach could be a tree-based animated transition that collapses/expands nodes based on a hierarchy [195].

4.7 Conclusion

In this chapter, we have presented Gridify, a generalization of OD maps and its implementation as an exploratory data analysis (EDA) tool for primarily geo-coded data, which extends previous work on grid-system approaches to visualization design ([157]). The tool builds on a core concept of decoupling data transformations and visualization construction mechanisms. We have demonstrated its expressiveness and effectiveness through several case studies. Its main advantages include 1) maintaining an explicit encoding of both data and visual parameters, which provides analysts with an overview of all their options; 2) chaining nesting operations that are rendered in a grid view that provides constant feedback, allowing analysts to select elements on which they want to zoom in; and 3) exporting and loading pre-set configurations, which provides a mechanism for transitioning between views in a step-by-step way, for pre-defining the exploration charts and patterns, and potentially for recommending future explorations steps. To finish, while our design and implementation of Gridify have focused mainly on OD data, essentially because authors work a lot with this type of data, we strongly believe the tool can handle a wider variety of data and visual abstractions and transitions. We intend to explore these possibilities in the future.

Apart from the OD data in the traffic domain, we implement the prototype with other spatial data in section 6.2.3, such as sports datasets (*e.g.,* to explore soccer players position during games) and trade datasets (*e.g.,* to explore trade patterns across countries).

CHAPTER 5

Temporal Visualization

Contents

Any use of ”we” in this chapter refers to Liqun Liu and Romain Vuillemot. We have a paper published in this work as follow:

- Liqun Liu and Romain Vuillemot. ”GROUPSET: A Set-Based Technique to Explore Time-Varying Data,” In *EuroVis 2022 - Short Papers*, the Eurographics Association, Roma, Italy, P. 5, June 2022. (Link)

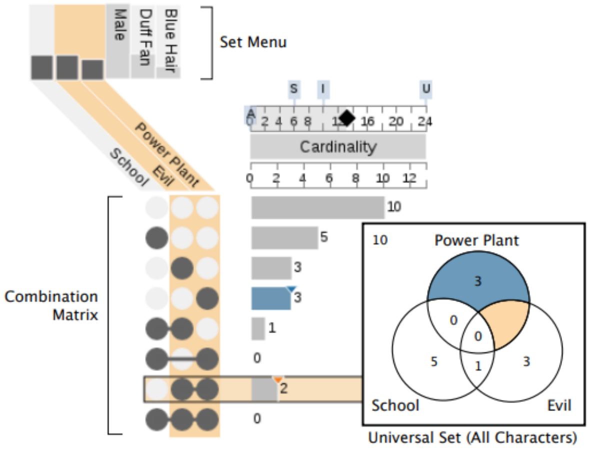


Figure 5.1: The Upset [30] sets visualization technique is an efficient alternative to Venn diagrams, where each sets intersections are encoded as rows of a sets combination matrix. In this chapter we build on this technique to explore time-based road traffic data.

5.1 Context and Motivation

Temporal analysis problem refers to analyzing the time-varying data (*e.g.,* traffic density changing over time). Analyzing such data is key to exploring the temporal aspects of traffic congestion, which can help transport planners (introduced in Section 2.3) optimize public transit, hence releasing the pressure on specific roads. We have studied the related visualization methods for analyzing temporal change patterns of traffic data in Section 2.4.1, such as a line chart—is usually the main representation to explore temporal change in datasets [36]. However, such a chart usually generates an overplot and hides patterns as datasets get increasingly large.

We introduce GroupSet, a temporal exploration technique to avoid the overplot, inspired by Upset [30], a set-based technique. Figure 5.1 shows the Upset to visualize the set intersection of Simpsons characters (an American animated sitcom) using the combination matrix. In the combination matrix, the column refers to the sets (*e.g.,* working at Power plant or evil), and the row refers to the set intersection (*e.g.,* the characters working at Power Plant and evil). Each row corresponds to an area in the Venn diagram. The highlighted row shows that two characters are evil and work at the power plant, but are not School students.

By improving the combination matrix, we propose GroupSet, a set-based visualization technique, to explore changes within large temporal datasets (task **pattern discovery and clustering**, introduced in Section 2.3) using line charts. The technique relies on set-based tasks [196] to understand the relationships between lines, using partial memberships (fractional membership in multiple sets) and change-related metrics. This approach reveals temporal similarities of elements by categories (sets) memberships, usually hidden by overplot. We demonstrate the applicability of the technique to traffic density data (introduced in Section 2.1.5) and report on usability feedback of an interactive prototype implementing the technique.

A prototype is available online, the datasets used to collect feedback from the tool, and its code published as an open-source project. It uses JavaScript and D3.js. Datasets used in this chapter are pre-loaded and stored in a remote server deployed using Heroku platform. GroupSet is the prototype as an Observable notebook to facilitate sharing early versions of the design, but due to code complexity, performance, and the need to use more screen real estate, it is re-implemented as an independent application. All the supplementary materials are listed in the Table 5.1.

Name	Link
Online prototype	https://llqsee.github.io/groupset/
Description of prototype	https://github.com/llqsee/groupset
First version prototype	https://observablehq.com/d/9efe0f3d70a90a63

Table 5.1: Supplementary materials. It lists two online prototypes (one is a version implemented on Observable notebook; another is a web-application) and a document to illustrate the prototype in detail (*e.g.,* how to implement one's own datasets).

5.2 Related Work

Our work relates to temporal visualizations where an independent variable (time) is plotted along with a dependant one (value over time). As we discretize the dependant value, we also relate to categories-based visualizations and set-based visual analysis. In contrast, other domains such as time series analysis and modelization–and recently machine and deep learning–have been extensively investigated to automate detection such as frequent patterns or anomalies detection. Our focus is on exploratory stages where few assumptions can be made about data quality and distribution.

5.2.1 Time-based Visual Exploration

Time-oriented data visualization is an important domain in visualization and has been widely investigated [197]. The standard visualizations to address this challenge use line chart designs and their variations. The line chart is among the oldest representation and conveys the raw data structure for visual inspection. Design variations such as the slop chart [119] provide a simplification, for better comparison, by displaying only the first and last elements to compare. The steepness of the slope indicates the trend, but it requires picking the right beginning and end of the time interval to compare. Univariate charts such as spark-lines [119] or temporal glyphs [198] are designed for a single object and variable at once. Changing graphical properties such as opacity [187] is an interesting approach for local identification of lines but does not allow connecting all segments across the temporal interval efficiently. Using polar coordinates, such as in radar charts or ChronoLenses [199], instead of Cartesian coordinate system, enables more compact visualization but does not scale to a large number of elements.

Extensions of the layout such as with Stack Zooming [200] and ChronoLenses [201] enables zooming on certain dense areas. A modulo-based approach like Horizon Charts [202] provides repetitions of the line chart once it reaches a threshold, to analyze high values time series. Further layouts are proposed to cluster similar episodes using a linear layout with StoryFlow [203] or self-organized ones like in Timecurves [204]. Again, scalability to a large number of elements is not addressed. The closest work to our approach using layouts is LineUp [205], to compare Top-N elements using a horizontal, temporal layout compare rank changes over time and multiple dimensions.

Beyond graphical properties, novel encoding or glyphs have addressed change detection, such as RankExplorer [206] and RankEvo [207] but are specific to local tasks and not suited to compare global trends. Candlestick patterns convey change within a trading day, with the candle's height and color encoding the price range and the stock's performance during the day; it is close to our approach by aggregating and grouping interval values to indicate a behavior. However, this is designed for local behaviors as we are interested in global ones.

User interaction is also an approach to better identify trends and patterns. Early work has investigated natural user queries that draw patterns [208] but local ones. Direct manipulation of time series can enable quickly navigating time [209] of a single temporal object and identifying peaks. Textual annotations [210] by users can also be an input modality to identify interesting patterns and retrieve similar sequences. Again all those techniques are suited to manipulate a handful of time series.

5.2.2 Categories and Set-based Exploration

Category exploration offers similar challenges to comparing time-varying data. The flagship example is the parallel coordinates chart [211], where the X-axis displays the various dimensions but also generates clutter and over-plot. Additionally, it needs to order the categories carefully. Building on this approach, ParallelSets [212] displays groups of elements similar to a Sankey diagram to reveal trends by cohorts of elements. Temporal categories [213] are presented using a flow chart that emphasizes changes across pairs of categories. Both approaches aggregate elements and thus hide individual patterns within each group.

Sets are a natural way to analyze categories [196] using sets data models. Among the many recent tools PowerSet [214] is a scalable technique that enumerates all the sets and represents them as a treemap to identify sets intersections distributions. Techniques to characterize sets intersections, such as UpSet [30], are also scalable but use a matrix approach. Radial Set [215] organizes sets on a circular layout. However, none of those techniques support time-varying sets or sets creations from time-varying data, assuming sets already are in the datasets and for a single time point. AggreSet [216] uses aggregation of elements to show their global membership to a category, but the aggregates are also based on a pre-defined category. TimeSet [217] shows group changes over time using contours shapes but is not scalable to many data elements.

The closest works to our approach using set-based analysis of changing categories is primarily Set Streams [218]. It supports set-based tasks and displays the changes as a flow chart. [219] is also very closely related to our work by displaying set changes over time using layered set intersection graphs, which represent intersections between all sets. Our work also relates to one of the earliest set visualization systems [220] that aimed at including set visualizations within existing visualizations (*e.g.,* bar charts) to explore additional data attributes and relations.

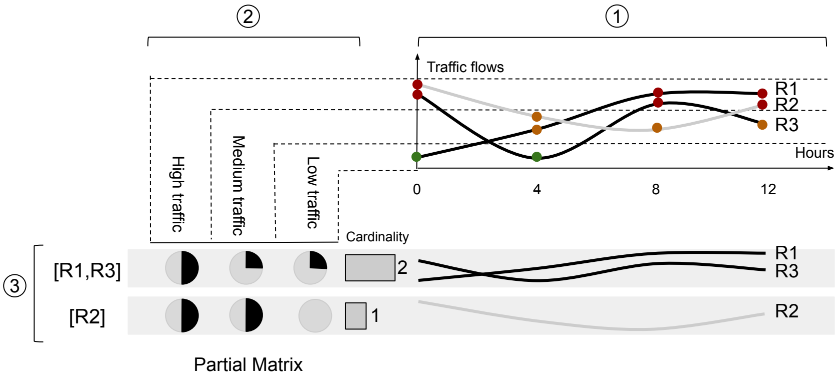


Figure 5.2: GroupSet technique overview. Sets are created from time-varying data (line chart) in ①. The same sets are organized as columns to indicate partial membership as pie charts in ② and set intersections are listed as rows to represent groups of similar lines (along with their partial membership representation) in ③.

5.2.3 Fuzzy Categories Creation

In Chapter 3, we have mentioned the categorization of quantities (*e.g.,* taxi speeds), which is important in general for road traffic data. However, set creation (quantity categorization) remains an under-explored area in the visualization community [196]. In particular, sets capture humans' thoughts, which are not binary, thus making elements members of multiple categories. The Fuzzy sets [134] community has introduced such data models to categorize into sets with a confidence value ranging from 0 to 1. Visualizations have been dedicated to fuzzy membership representations such as the Disk diagrams [142] to convey both categories and confidence with a linear scale. However, it does not apply to multiple categories or instances. Set creation tools inspired by fuzzy sets have been built to create categories from quantitative scales [221]. While they are not suited for visualization for the exploration of time-varying data, the underlying idea of splitting a quantitative interval into categories and partial membership is related to our work.

5.3 Data Model

Our general goal is to group multiple road-traffic temporal *elements* based on temporal similarities over a particular attribute (*e.g.,*speed). We operate as follows: we first discretize temporal values into *sets* (*e.g.,* `LOW traffic, MEDIUM traffic, HIGH traffic`) as in Figure 5.2; then, we count every time an element reaches one of the sets (*e.g.,* `LOW` x 2, `MEDIUM` x 5, `HIGH` x 3). Each element reaches as many sets as time steps (*e.g.,* 4). We further use percentages (*e.g.,* `LOW` x 20%, `MEDIUM` x 50%, `HIGH` x 30%) so memberships are normalized within a $[0, 100]$ range as we represent it as a pie chart ◗. We now provide definitions of these data models and the set-based metrics we derive.

5.3.1 Elements, Categories and Membership

The *elements* $E = \{e_1, .., e_n\}$, usually are the rows or items from a finite dataset of size n. Each element has a quantitative value $q_t \in Q$ function of time, within a temporal interval we consider uniformly sampled $t \in T = \{t_1, .., t_p\}$, with $p = |T|$ time steps. The temporal axis is usually represented as the X-axis of time series of rankings, and the q_t value as Y-axis.

The *categories* creation can be achieved dynamically by the user (Figure 5.2), which basically is the split of the quantity range $Q = [min(Q), max(Q)]$ into intervals or *sets* $X = \{X_1, .., X_i, .., X_n\}$ ordered by i as $x_i < x_j$. Thus, each x_i is associated to an interval dividing Q. We now define explicit rules that define elements *membership* to each set as a function $f_X(q_t) : X \rightarrow [min(Q), max(Q)]$ for every time steps t (sets are named to be more human readable):

$$x \rightarrow \begin{cases} LOW & if q_t \leq 20 \\ MEDIUM & if 20 < q_t \leq 60 \\ HIGH & if 60 < q_t. \end{cases} \tag{5.1}$$

5.3.2 Membership Calculation

Membership is calculated for each element e over time, for each set. Thus e may be part of multiple sets X_n, so the set degree D for an element e over $t \in T$ is $D_e = \{d_t, ..., d_t\}, t \in T$ and all the values of degrees are equal to 1 as $\sum_{t \in T}(d_t) = 1$. For a single set x_i, the normalized membership calculation is (*e.g.,* for `LOW`):

$$\sum_{t \in T} \frac{f_{\{LOW\}}}{|T|} = ◑ \tag{5.2}$$

To facilitate the comprehension, we introduce a pie chart representing total membership to a given set x_i. The black wedge of the pie chart encodes the % of membership for a given set x_i. If a complete set membership would be the following ●. However, we usually represent all the sets X to represent the distribution of membership across all sets, so for an element always `LOW` [●, ○, ○]. If the membership is evenly distributed across all the sets [◑, ◑, ◑]. Note at this point the changes are aggregated across all the time steps. An important property is that total membership is always 100%.

$$● = [◑ + ◑ + ◑] \tag{5.3}$$

5.3.3 Changes Patterns

We introduce a metric to calculate the magnitude of the change Δv, which is the difference of value between two snapshots $[t_i, t_j]$ over both values of q_i and q_j. This magnitude usually is the *slope* of the corresponding segment on the visual representation:

$$slope = \Delta v_{qt_j, qt_i} = \frac{qt_j - qt_i}{t_j - t_i}$$

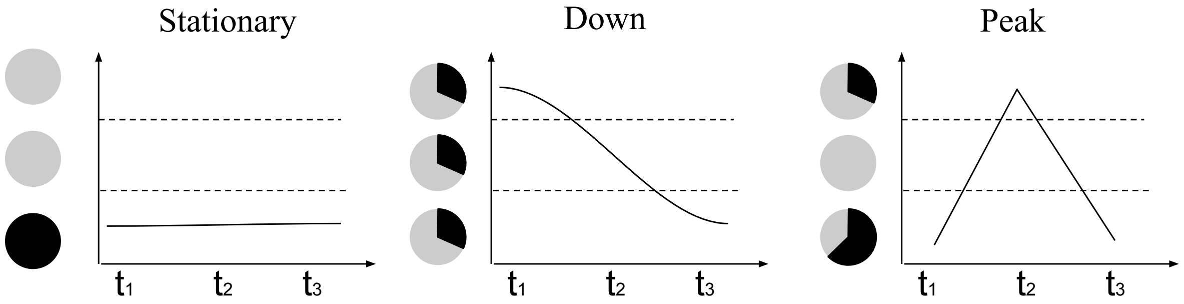


Figure 5.3: Changes patterns with both their temporal representation and encoding using our set-based approach the typical changes patterns. If the membership is always to the same group we denote it as follows: ●, if it never belongs to a group it is defined: ○ and finally partial membership like 24% is as follows: ◔.

We introduce *categorical change*, the set membership change between t_i and t_j. Indeed, changes that generate a different set membership are considered more important for the analysis and are represented as an additional category of change (Figure 5.3). Such change has been implemented in [206, 213] using respectively glyphs and color scales. Figure 5.3 illustrates the typical patterns types of changes related to *increase* and $\Delta v > 0$, and *decrease* $\Delta v < 0$ (but not limited to) but accross dimensions:

- *Stationary* ($\Delta v = 0$) *:* where the set membership degree is $|S| = 1$, with 100% of membership for this intersection, represented as horizontal an line. It is translated as an intersection of degree 1.
  E.g.: [●, ○, ○]

- *Up (down):* a constant raise or fall trend. Generally, it passes through all the degree. So the degree is 3.
  E.g.: [◔, ◔, ◔]

- *Peak:* a single peak is observed, for a short period so degree is 2, but with not a lot of membership to the non-majority sets.
  E.g.: [◕, ○, ◔]

5.3.4 Set Aggregation

We now introduce a general mechanism to aggregate sets based on their properties. Aggregation means that all the intersections share a categorical property such as the set membership, change pattern, or degree. Using set definitions, this is translated as a union ∪ of, *i.e.* all the intersections with a given membership degree. The second level of aggregation can be achieved by repeating the aggregation to similar sets, and then a series of subsets ⊂ is created. While aggregation operates on categories, it also can operate on quantitative values, such as membership degree, which needs to be split into groups to create those categories. Figure 5.4 provides a compact representation of the elements after they have been aggregated. Each row

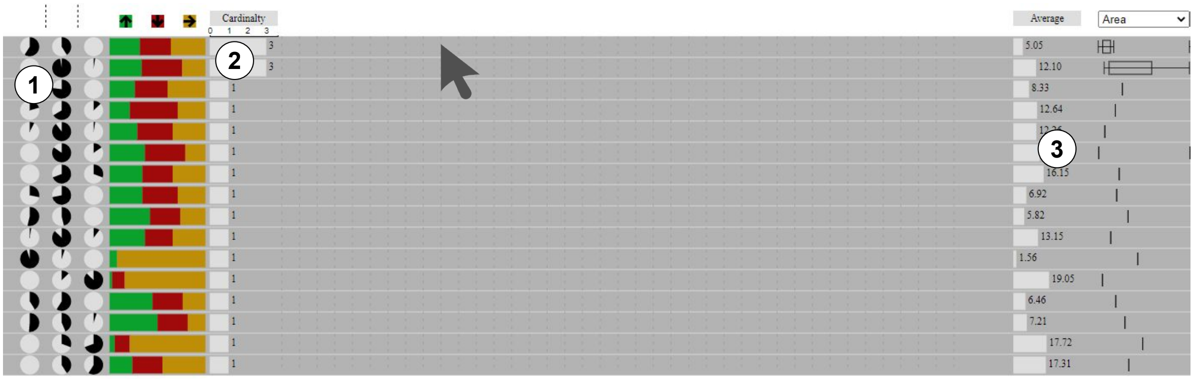


Figure 5.4: A compact view of aggregates (where no details of their content is visible). It summarizes all the groups of elements using the pie chart (①) and changes sorted by cardinality (②), followed by the statistic of group attributes (③).

encodes a group of elements and displays statistics related to the sets and the changes across instances.

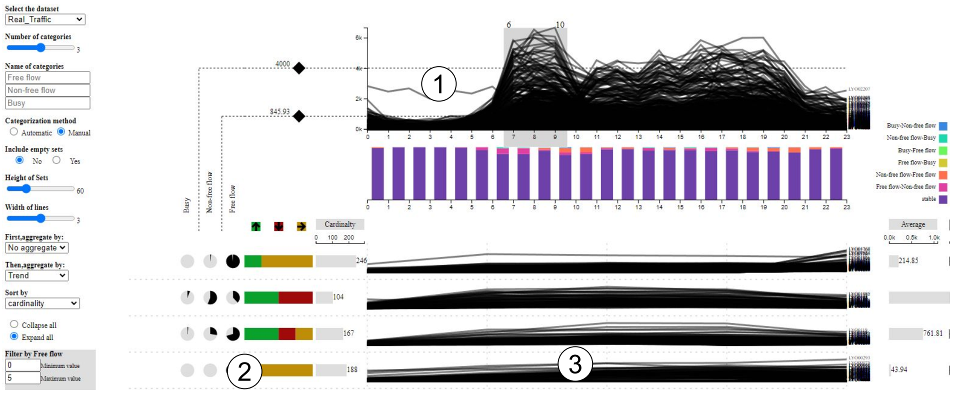
5.4 The GroupSet Technique

Our technique addresses the characterization of temporal elements (*i.e.*line variations along the Y-axis) in large datasets, which categorize lines based on their time-varying values. The core of the GroupSet technique is a partial membership metric that captures temporal changes.

5.4.1 Design Rationale and Tasks

The ultimate goal of GroupSet is to generate views of groups of elements that can be inspected in detail, without clutter, using simple set creations, sorting, and aggregation. The technique operates using the following workflow: 1) sets are created from the Y-axis of a line chart representing elements, encoded as a line over a time interval T, 2) each set are represented using an set matrix [196] inspired by UpSet [30] where each row is a set intersection of elements sharing similar membership properties (*e.g.,* [●, ●, ●]), 3) details of each intersection is detailed as a filtered view of the main line chart. GroupSet addresses the following set-based tasks [196]:

- **T1**: Create new sets.
- **T2**: Find elements with their set memberships and specific time intervals.
- **T3**: Find intersections with specific time intervals.
- **T4**: Analyze intersection relations with time intervals.
- **T5**: Analyze elements distribution with time intervals.



Select the dataset: Real_Traffic

Number of categories: 3

Name of categories: Free flow, Non-free flow, Busy

Categorization method: ○ Automatic ● Manual

Include empty sets: ● No ○ Yes

Height of Sets: 60

Width of lines: 3

First,aggregate by: No aggregate

Then,aggregate by: Trend

Sort by: cardinality

○ Collapse all
● Expand all

Filter by Free flow
0 Minimum value
5 Maximum value

Figure 5.5: GroupSet explores the changes of traffic density in Lyon, France, during a day (24 hours). In ① traffic densities are categorized by rank into 3 sets (`Busy`, `Non-free flow`, and `Free flow`). Users can then explore the traffic densities based on their memberships to those sets over 24 hours and identify temporal patterns by manipulating a combination matrix of set intersections ②. The final view provided by GroupSet is a filtered line chart corresponding to each set intersection showing teams with similar patterns ③ along with other attributes (*e.g.,* change).

- **T6**: Filter time intervals.

Temporal changes are represented using pie charts where the black wedge encodes the % of membership (that we define it with *set membership degree*) to a category (that we now call *set*) which is one of the value intervals of the dependant quantity domain. Sets usually are an interval of values. For instance, if a temporal element is constant over time, and its value interval is divided into three categories, its membership is the following: [●, ○, ○] (black circle indicating 100% membership to a single set, gray circles 0% membership to the two others). If the membership is spread across multiple sets, the representation can be as follows *e.g.,* [◑, ◑, ◑]. An important property is that total membership across sets is always 100%. The pie chart representation is key to characterize visually, order and filter groups of elements, which is the ultimate goal of GroupSet.

5.4.2 Sets Manually Created

Sets are manually created by defining intervals over the Y-axis of the line chart (Figure 5.2, ①). By default, 3 sets are included, but users can add or remove sets, customize their value intervals, and name them. The interval customization uses direct manipulation of the Y-axis. Every time set properties are changed, the interface updates so the user can immediately see the result of those manipulations. We pre-loaded each dataset in the tool with sets, their frequent names, and value intervals. We finally include a way to automatically create sets based on the temporal value distributions using CKmeans [222] to automatically divide data into groups. If we would like to cluster n data to a p cluster, the equation of CKmeans is as follows:

$$D = \sum_{i=1}^{n} \sum_{j=1}^{p} \mu_{ij}^{m} d(X_i; C_j)^2 \tag{5.4}$$

where μ_{ij}^m refers to the membership degree of the i data sample belonging to j cluster and $d(X_i; C_j)$ refers to the distance between the X_i and center value C_j. The optimization goal is to minimize the distance among all data samples and cluster centroids.

5.4.3 Aggregation Exploration

Aggregation exploration is achieved using the set permutation matrix and aggregates all elements in a similar way to the UpSet matrix [30]. Each column is a set, and each row encodes a single type of intersection, where the pie chart shows the intersection's partial membership inspired by AggrSet [216]. It aggregates the elements if they have the same distribution among sets. Such as shown in Figure 5.2, $R1$ and $R3$ both have 2/4 time points in `High traffic`, 1/4 time points in `Medium traffic` and 1/4 time points in `Low traffic` respectively. Moreover, GroupSet also allows users to sort and filter the aggregated groups to change the vertical resolution. Finally, a horizontal bar chart encodes the intersection's cardinality (*i.e.* number of elements). Each set intersection embeds a stacked chart encoding a change across two sets (Figure 5.5 ②), which presents the average change values of all elements in each set intersection, where green indicates ups, red downs, and orange stable slopes. The stack height (horizontal) shows the most frequent changes for each category of change.

5.4.4 Elements Details, Change and Attributes

Each groups of elements is displayed as a filtered line chart on each row (Figure 5.5, ③). This enables a less cluttered representation of an individual inspection of trends. A stacked chart displays the changes for each time step t_i (Figure 5.6). Each set intersection also embeds a stacked chart encoding a change across two categories (Figure 5.5, ②) where green indicates ups, red downs, and orange stable slopes. We opted for this chart instead of flow charts [213] for simplicity and compactness. The height of the stack shows the most frequent changes for each category of change.

A brushing feature of the line chart is provided to select a sub-set of the time range T for the intersections visualizations (Figure 5.5, ①). This brushing will be valuable to discard some time steps that may not be relevant for the analyst (*e.g.,* due to high variability).

5.5 Case Studies on Traffic Density Data

We load a traffic density dataset from a public open data repository in Lyon, FR (introduced in Section 2.1.5), for a single day of observation (Figure 5.7). The datasets include 1334 road segments during a day (24h), monitored by sensors using inductive loops. The X-Axis represents 24 hours and the Y-Axis represents the traffic densities collected by the sensors. Generally, the traffic densities are low from 0:00 AM to 6:00 AM, and then start rising and reaching the morning peak around 8:00 AM. After the afternoon, it descends again to a lower traffic density. As [71] mentioned, one of the challenges in visualizing the traffic data is that it has too many data items to track. A general line chart cannot handle too many road segments without overplot. Thus, for the traffic dataset, we aim at achieving the followings:

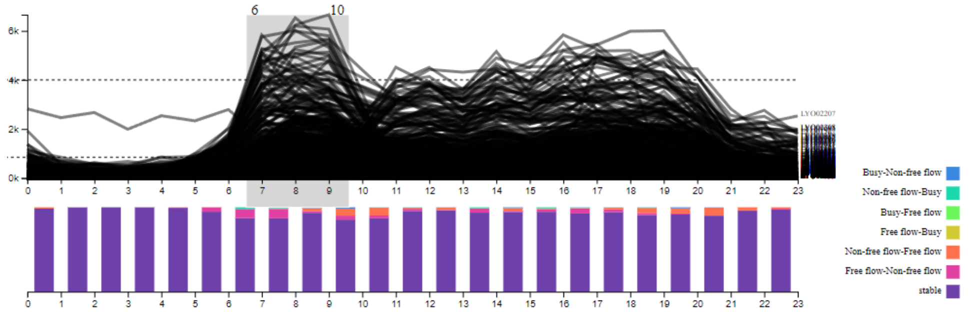


Figure 5.6: Stacked chart below the line chart encode the change between two adjacent time points (color scale indicates the change patterns, *e.g.*, `Busy-Free flow` hinting it changes from `Busy` category to `Free flow` category). The legend of changing patterns is on the right.

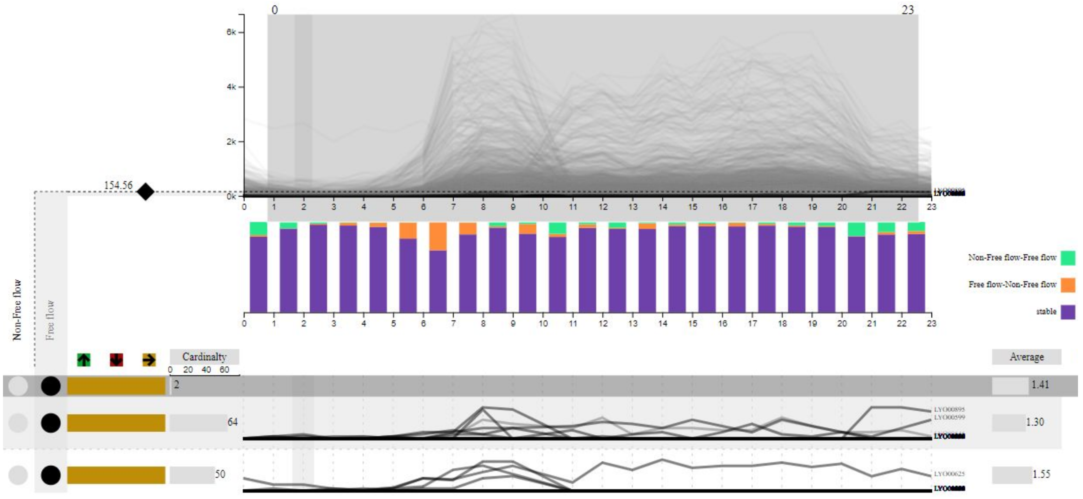


Figure 5.7: Road segments that are always not busy. An aggregated group with two subgroups contains 64 and 50 road segments that only belong to `Free flow` during 24 hours.

Detecting roads with low traffic. Finding the roads that are *not busy* (free flow) is critical work to help traffic planners optimize traffic load. These traffic flows can be re-planned from the roads where the traffic congestion always happens. Thus, assisting traffic operators find the roads that are always not busy (during 24h) is able to improve the traffic conditions. As shown in Figure 5.7, we focus on the entire period (24h) and create two sets (`Non-free flow` and `Free flow`) (**T1**), with a low density threshold to identify the free flows. By using the filter panel, we keep only the `Free flow` roads which are characterized with a membership to this set only (**T3**). The result is shown in Figure 5.7, and two groups exist (as we aggregated by change pattern): one contains 64 road segments and another one contains 50 road segments (**T2**). These road segments mostly have less traffic flow. Further investigations is needed for their relevance to bear more traffic.

Analyzing the traffic flow patterns with a specific time interval. Managing the traffic during peak time is very important for a traffic manager, usually during rush hours around 8:00 AM and 6:00 PM. To analyze this peak pattern of traffic density, we first brush the time interval

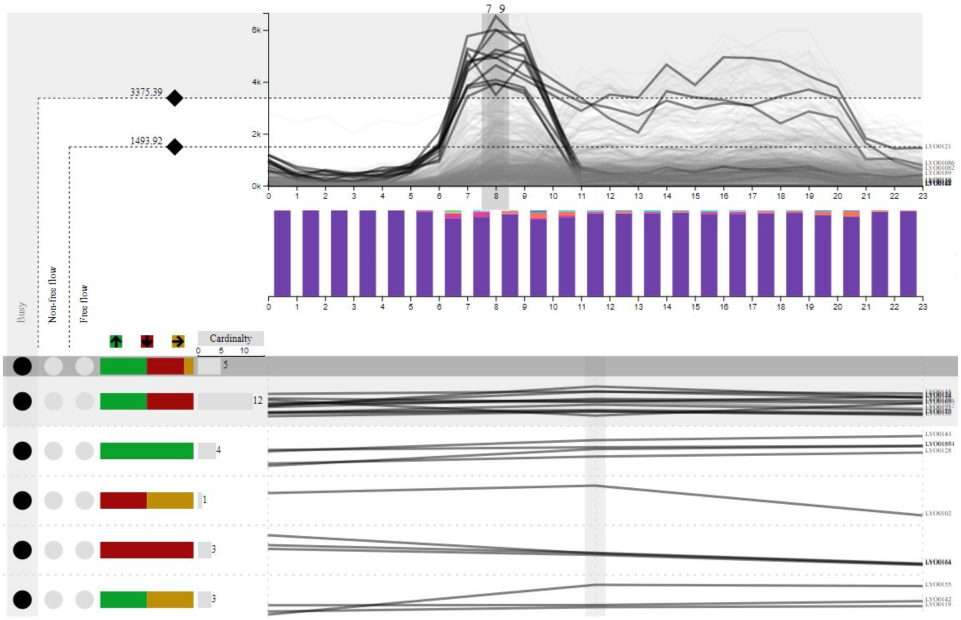


Figure 5.8: The traffic flow patterns during morning peak. An aggregated group with 5 subgroups reaches the peak time in the morning, but some do not reach the peak in the evening.

to the period which we are interested in (7:00 AM to 9:00 AM) **(T6)**. We then add `Busy` as shown in Figure 5.8 to capture the busiest roads at peak time (**T1**). We observe there is an aggregated group (containing 5 subgroups) only belonging to `Busy` (**T4**). A first insight is that the largest group has roads with a clear increase and decrease change pattern (green and red bars), which means these roads are not stable being either rising up or falling down. A second insight is that a certain number of roads peak in the morning, but they do not have peak time in the evening, as shown in the global line chart (**T5**). These two patterns are difficult to identify in the full dataset, and the roads that match them can further be explored in the tool using the attribute panel of GroupSet (*e.g.*, road segments average length).

5.6 Feedback and Perspectives

We conduct a preliminary study to validate the usability of GroupSet with four researchers that frequently use data visualization tools to validate the tool's usability and detect any major design issues. We present them with the tool loaded with the soccer dataset as it is the easiest one to understand as all researchers are familiar with soccer. We then demonstrate the standard workflow from categorization to detailed view exploration. We then ask them to use the tool and follow a think-aloud protocol to capture their thoughts and understand their intentions. We first ask them to reproduce the demonstrated workflow, then they conduct an open-ended exploration with any dataset of their choice regarding the tasks. All participants find the tool flow relevant to exploring such a dataset and logical, from the exploratory chart to the detailed view. The main remark we collected is preserving the sequence of events we will discuss as a tool's limit. They also notice some performance issues we will discuss in the next section.

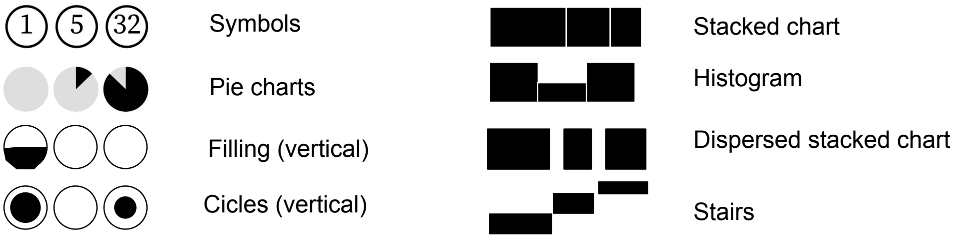


Figure 5.9: The suggestions of combination matrix. We will validate them in further experiments to compare which one is better by interviewing visualization designers.

Performance and scalability. The first feedback we collected during the usability study is the performance, especially when there are many elements and sets. For some application domains, it may be needed to include more than three sets either because it is semantically relevant or because a finer grain of analysis of changes is needed. Currently, GroupSet supports up to five sets which potentially generates an important number of intersections. However, as we only focus on analyzing a subset of the data (due to sorting and aggregation), we argue there is no need to calculate and visualize all the set intersections.

Applicability to other datasets and beyond line charts. Our tool is generic to support any temporal dataset (*e.g.,* University rankings) without any change in the design. Regarding the type of chart it supports, it is applicable beyond line charts to all charts with a single dependent variable (time points) plotted on the Y-axis to visualize the distribution of elements values in each time point among different sets (`Normal`, `Top 5`, `Bottom 5`, etc), such as histograms, density plots, or bar charts. 2D temporal charts, such as scatterplots or geo-map, will require some significant change in the design, but GroupSet could be used as a marginal plot to filter and group such charts over time.

Limits and perspectives. GroupSet currently does not capture intra-group variations and temporal changes in the pie chart: thus, some temporal elements may be included in the same set intersection, despite having different patterns (*e.g.,* one is increasing, the other decreasing). This issue could be addressed by adding a second level of aggregation to the intersections using a global trend indicator to group elements by either globally increasing or decreasing value. However, this may generate additional intersections and slow performances. Another perspective of our work is investigating alternative designs to the pie charts to encode intersections and memberships. While circles and pie charts are already used in [30, 216] for such encoding, there currently is no formal evidence that they are the best-suited representation. We plan to implement and formally evaluate alternatives using other statistical charts (*e.g.,* bar charts, box plots), as shown in Figure 5.9. Finally, the traffic density case only helps us analyze how traffic flows change over time. However, spatial information should be taken into account. In the next step, we will consider connecting GroupSet with a map to analyze the spatio-temporal information.

5.7 Conclusion

We have introduced GroupSet, a set-based visualization technique to compare multiple items across several instances. The technique relies on a data model that captures the set creation and the change of memberships of time-varying items across those sets. We have described a case study demonstrating the tool's applicability to traffic density changing over time. We have also reported on early usability feedback, and we plan to conduct a formal evaluation of the tool with domain experts. We believe GroupSet is applicable beyond the presented case studies as a generic tool to compare multiple elements over instances. As we provide an interactive prototype and its code as an open-source project, we expect to foster more research in the domain of set creation and time-varying analysis.

Apart from the traffic density data, we test the implementation in other areas such as the Machine Learning classification and the ranking data. For example, we use GroupSet to analyze the classification results of the **MNIST** dataset (A handwritten digits datasets for training imaging process system and algorithm [223]) and the ranking data of soccer teams. We introduce the implementation of these two datasets in Section 6.2.3.

Chapter 6

Deployment in Traffic Control Centers and Other Applications

Contents

Any use of ”we” in this chapter refers to Liqun Liu and Romain Vuillemot.

6.1 Deployment in Traffic Control Centers

This section discusses how to deploy our three novel visualization techniques (FuzzyCut, Gridify and GroupSet) in real-world traffic control centers. We first introduce road traffic control centers in Section 6.1.1, and then introduce the context and tasks in Section 6.1.2. After that, we take two examples with Lyon dataset to explain the deployment in Section 6.1.3, and eventually, we discuss the deployment challenges in Section 6.1.4.

6.1.1 Road Traffic Control Centers

We have introduced road traffic control centers in Section 2.3.2. Now, we explain them with detailed information, including their goals, limits, and related studies. This information provides the necessary knowledge for the deployment.

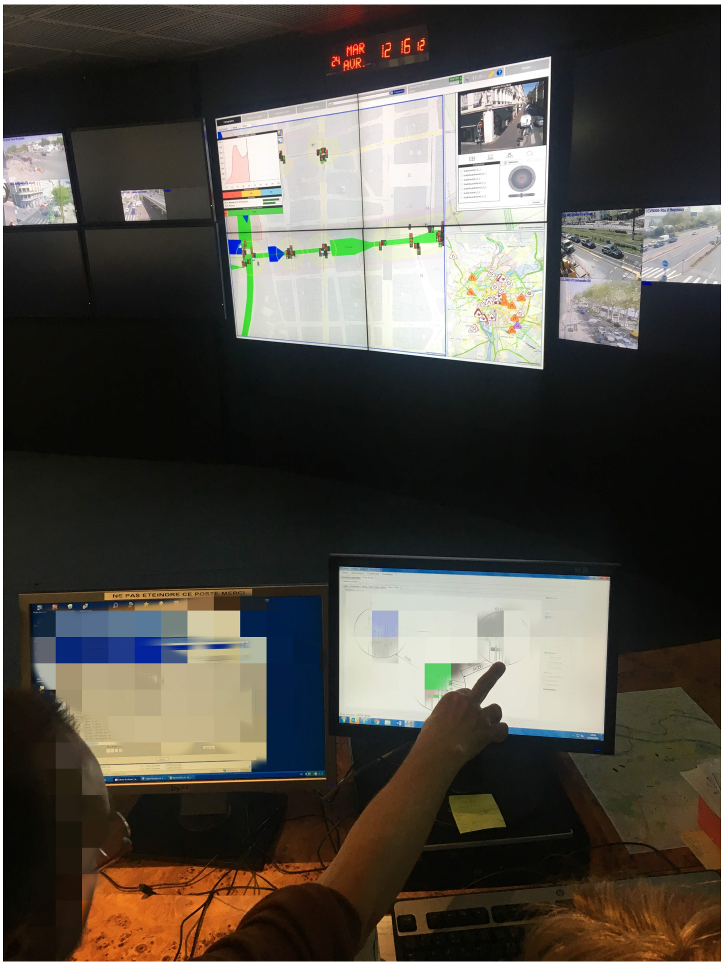


Figure 6.1: A picture we took in the traffic control center in Lyon, France, an indoor physical facility where operators monitor traffic on a large display from remote workstations. It offers the visual environment for merging multiple visualization techniques to analyze heterogeneous spatio-temporal traffic data.

We have visited several control centers in France, and investigated the current visualizations and workflow practices. Such centers are indoor physical facilities (Figure 6.1) with restricted access as they play a key role in managing traffic but also roadworks and road message boards (*i.e.* to announce congestion or closed road segments). Their main goals include: (1) maximize the available roadway capacity, (2) minimize the impact of incidents, and (3) assist in emergency services [224]. To this end, like the ones in Paris described by Prouzeau [225], they are composed of a large display that shows both a map of the monitored road network, each road colored in the function of the traffic density, and a matrix of video stream coming from CCTV cameras. Each operator in the room has a workstation composed of several screens on which they can access useful information, including a detailed version of the traffic map and a specific CCTV camera (which they can also control). Actions on the traffic are mainly required when incidents happen (*e.g.*, breakdowns, accidents) to warn the drivers on the road and, if necessary, reroute traffic and assist first-responders. Incidents can be frequent.

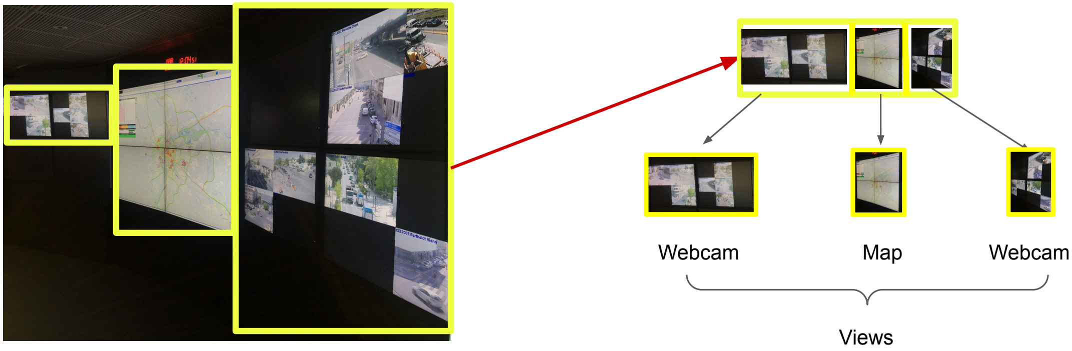


Figure 6.2: Dashboard and view examples. The left one shows the dashboard of the traffic control center (the picture was taken from the Lyon traffic control center). The right one refers to the views extracted from the dashboard.

For instance, around 22 of these daily incidents are in the peripheral ring in Paris, its busiest road. Operators can spot these incidents directly on the CCTV cameras or by noticing unusual traffic jams on the density map. They can also be warned by police onsite or by bystanders. In the latter, information regarding the location of the incident can be vague, and operators need to look for the precise location using CCTV cameras.

The management of these incidents significantly increases operators' workload, up to 40%, as suggested by Zeilstra *et al.* [226]. In their study in a road traffic control center in Grenoble, Starke *et al.* [227] show that operators must go back and forth between screens, including the traffic map and the CCTV videos, to solve the incident. They do not look at the video matrix in the large display as it provides too much information. Instead, they focus on the video from the CCTV camera that shows the incident and display it directly at their workstation. This allows them to avoid information overload and focus only on the incident. However, it also restricts their situation awareness to this specific part of the road at the expense of the rest of the network that could also be impacted by this incident or another unrelated, a phenomenon called intentional blindness [228].

To help with information overload but avoid unintentional blindness, Baber *et al.* [229] propose a road traffic control dashboard in which the information needed to do a task is in the same view. For instance, the traffic map also provides a temporal evolution of the traffic density for each road segment. Anwar *et al.* [14] show the impact area of the incident directly on the traffic using a lens that grows with time, and Prouzeau *et al.* [230] use a dragmag to show future traffic prediction directly on the map. Finally, Schwarz *et al.* [231] use a lens on the map to allow operators to access a detailed view of the traffic and CCTV cameras for a specific area of the map, which helps them associate detailed information with its context.

In order to avoid information overload in traffic control centers, researchers have been devoted to improving the traffic maps or enriching the interaction ways with maps. However, less work is related to the flexible deployment of traffic control centers.

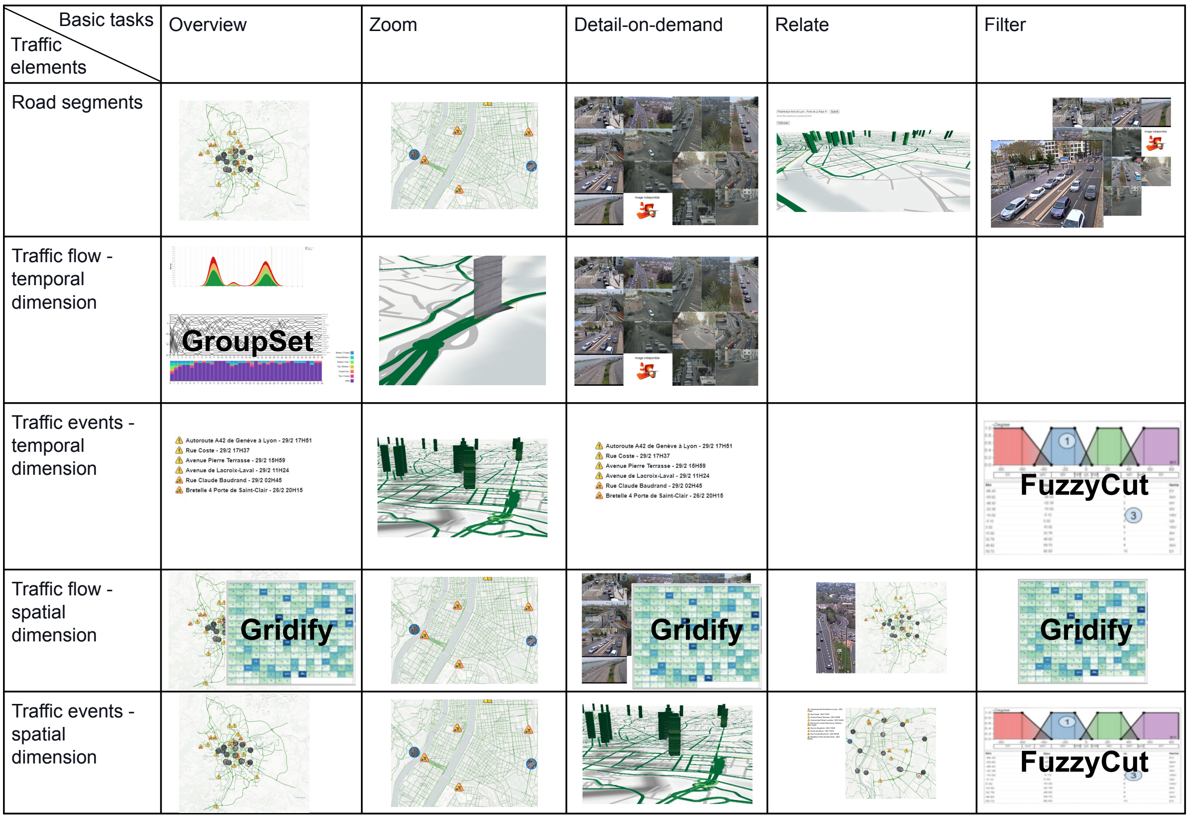

Basic tasks
Traffic elements
Overview
Zoom
Detail-on-demand
Relate
Filter
Road segments
Traffic flow - temporal dimension
Traffic events - temporal dimension
Traffic flow - spatial dimension
Traffic events - spatial dimension
GroupSet
Gridify
FuzzyCut
Autoroute A42 de Genève à Lyon - 29/2 17H51
Rue Coste - 29/2 17H37
Avenue Pierre Terrasse - 29/2 15H59
Avenue de Lacroix-Laval - 29/2 11H24
Rue Claude Baudrand - 29/2 02H45
Bretelle 4 Porte de Saint-Clair - 26/2 20H15
1
3

Figure 6.3: The overview of basic tasks, traffic elements, and visualization techniques. Basic tasks in the column and traffic elements in the row construct a matrix that consists of visualization techniques. It illustrates that a visualization technique in each point achieves a task for a traffic element.

6.1.2 Context and Tasks

In previous chapters, we have introduced three visualization techniques that focus on univariate data (FuzzyCut), spatial traffic data (Gridify) and temporal traffic data (GroupSet). As traffic data analysis tasks are complex (*e.g.,* traffic monitoring, pattern discovery and clustering, and situation-aware exploration and prediction, introduced in Section 2.3.2), and as data sources are heterogeneous (*e.g.,* traffic density data, taxi trajectory data, event data, and webcam data, introduced in Section 2.1.5), a visualization environment that can realize multiple tasks and visualize heterogeneous data is needed.

Traffic control centers already provide such an integrated visualization environment by using a wall-display multiple views dashboard. It helps traffic operators know what is happening at the city and road segment levels with webcam views and map views, as shown in Figure 6.2. We argue that combining our different visualization techniques in a dashboard to achieve more complex tasks is valuable. Results from this work were presented to the MI2 project (the research project this manuscript is in conjunction with) and will help shape future control center displays.

The basic tasks of visualization contain *overview*, *zoom*, *filter*, *details-on-demand*, *relate*, *history*, and *extract* [121] (we have introduced in Section 2.4.5), as shown in the columns of

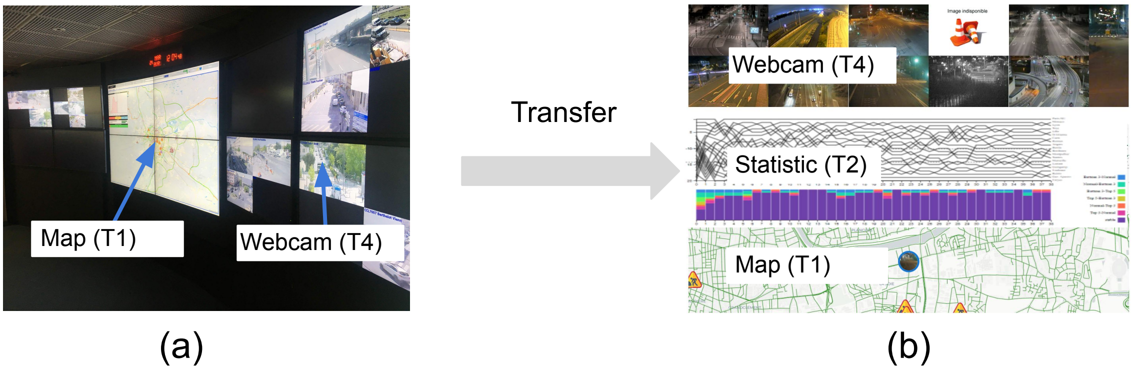


Figure 6.4: Transfer from a real-world traffic control center to a simulated dashboard visualization environment. (a) refers to a picture taken from a traffic control center in Lyon, France; (b) refers to a new dashboard including webcam, statistic, and map views.

Figure 6.3. *Overview* refers to the overview of the entire data. *Zoom* refers to the zoom on the data that users are interested in. *Filter* refers to the dynamic queries for filtering uninteresting data. *Details-on-demand* refers to selecting specific data or a group of data. *Relate* refers to the view of relationships among data. Also, traffic elements are shown as the rows of Figure 6.3, including road segments, traffic flow, and traffic events. Eventually, the basic tasks and traffic elements construct a matrix (Figure 6.3) that consists of different visualization techniques at each point of the matrix. In other words, visualization techniques can achieve one or several *control center-oriented tasks*, *e.g.,* GroupSet enables an overview of traffic flow in the temporal dimension. We summarize *control center-oriented tasks* as follows (based on[61]):

- **T1**: Overview of the traffic flow and traffic events in spatial dimension;
- **T2**: Overview of the traffic flow and traffic events in temporal dimension;
- **T3**: Zoom in the map to observe the traffic flow and traffic events in specific spatial dimension;
- **T4**: Details-on-demand of the specific road segments, traffic flow, and traffic events;
- **T5**: Relate the traffic events with the map;
- **T6**: Relate the specific traffic flows with the map;
- **T7**: Filter the specific road segments;

Multiple views or visualization techniques enable organizing a dashboard, shown as Figure 6.4 (a), a traffic control center in Lyon, France. The wall-display dashboard consists of a traffic map and webcams. The map enables to tackle **T1**, and the webcams enable to tackle **T4**. These two tasks are space-related and do not support time-related tasks. Thus, we have to transfer this dashboard to another by replacing, removing, or adding views if we want to achieve the temporal relevant tasks. Figure 6.4 (b) shows the solution by adding a statistic

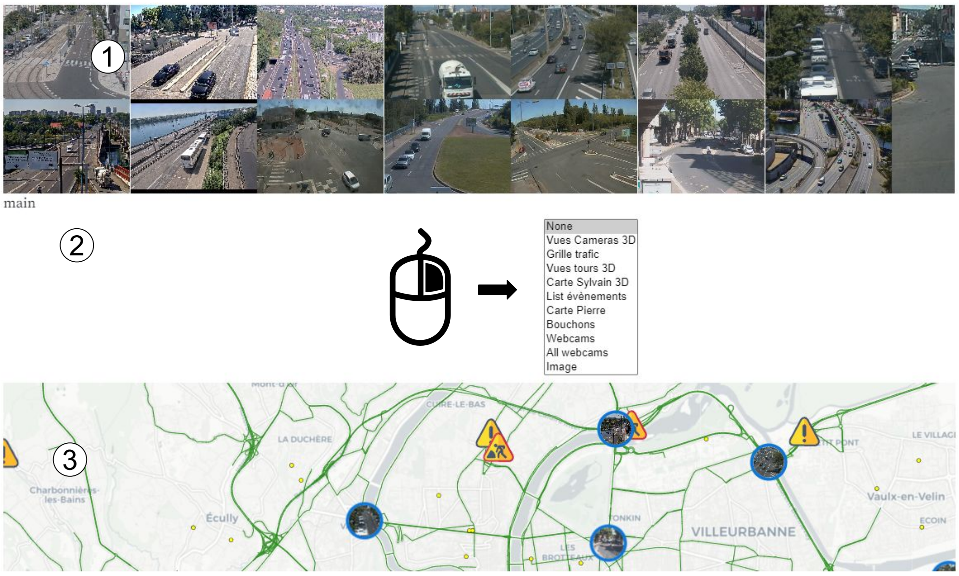


Figure 6.5: Interaction of ControlCenter. ① shows the `webcam view` to monitor the road intersections. Users can add a view by right-clicking and picking a view in point of view (②). A `map view` to overview the road traffic locates at ③.

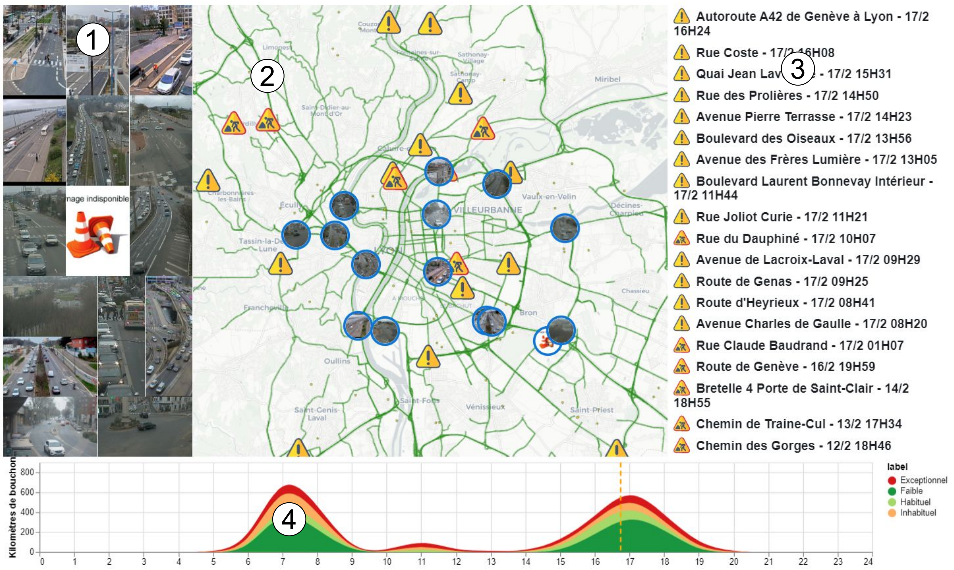
view (GroupSet) to achieve **T2**. To do so, an adjustable prototype that can change the views in the dashboard is needed.

Designing such a scenario is supported using a rapid visualization prototyping method named ControlCenter, to help traffic operators fast transfer from one dashboard to another by adding, removing, and replacing the views (*e.g.,* map, webcam, or statistic). ControlCenter is implemented in JavaScript using D3 [136]. The implementation choice is to use Observable notebooks as a coding and deployment environment. The code is released as an interactive prototype (link), as shown in Figure 6.5.

6.1.3 Scenario in Lyon

We explain the ControlCenter workflow with two examples. Using ControlCenter, we generate two dashboards based on the road traffic data from Lyon (introduced in Section 2.1.5). One of the two dashboards reorganizes the existing views, observing how traffic flows and traffic events are distributed on a map. Another dashboard is implemented with existing views and novel visualization techniques designed in this manuscript, aiming to explore how traffic flows change over time. This section introduces how ControlCenter implements the dashboard and achieves the control center-oriented tasks. Co-authors of this work are in charge of updating the reorganization of dashboards.

Reorganizing a dashboard using existing views. Arriving at certain places on time is a big concern for city commuters. Especially those who drive, they have to consider the traffic density distribution and whether the road to be passed is under roadworks. ControlCenter can combine multiple views as a dashboard to help traffic operators overview road networks and the traffic flow situations over road networks. This information from traffic operators can



Autoroute A42 de Genève à Lyon - 17/2 16H24

Rue Coste - 17/2 16H08

Quai Jean Lav - 17/2 15H31

Rue des Prolières - 17/2 14H50

Avenue Pierre Terrasse - 17/2 14H23

Boulevard des Oiseaux - 17/2 13H56

Avenue des Frères Lumière - 17/2 13H05

Boulevard Laurent Bonnevay Intérieur - 17/2 11H44

Rue Joliot Curie - 17/2 11H21

Rue du Dauphiné - 17/2 10H07

Avenue de Lacroix-Laval - 17/2 09H29

Route de Genas - 17/2 09H25

Route d'Heyrieux - 17/2 08H41

Avenue Charles de Gaulle - 17/2 08H20

Rue Claude Baudrand - 17/2 01H07

Route de Genève - 16/2 19H59

Bretelle 4 Porte de Saint-Clair - 14/2 18H55

Chemin de Traine-Cul - 13/2 17H34

Chemin des Gorges - 12/2 18H46

Figure 6.6: Reorganize dashboard using existing views. A webcam matrix view shows the overview of specific road intersections in ①. A map view shows the overview of the traffic status, road events, and webcam locations in ②. An event list view shows the detailed information of events in ③. A stacked line chart shows the temporal changes in traffic flows in ④.

be propagated to commuters, improving their traveling efficiency. First, we input the relevant traffic data in Lyon (introduced in Section 2.1.5 as CRITER datasets) in ControlCenter, including webcam data, traffic density data, and traffic event data. Four views are utilized in this dashboard: a webcam view, a map view, an event list view, and a stacked line chart view, as shown in Figure 6.6. A webcam (①) visualizes the specific road status (**T4**). It helps drivers avoid driving at these road segments where traffic congestion exists. Furthermore, the webcam also helps traffic operators observe whether these roads have traffic accidents (**T4**). The map view (②) can visualize the spatial information (the positions of objects) related to webcams and traffic events (**T5**). In the map view, users can also observe Lyon's traffic flow distribution on all road networks (**T1**). The traffic flow seems good because all the roads are green. Moreover, there are still many warnings and roadworks on the map view. In order to know the temporal information of these events, we use the event list view (③). As we can see from this view, all the warnings are on 15 February (**T2**), so we should make concerned about it if we travel on 15 February. It has roadworks on `Rue du Dauphine` at 09H31 as well. Thus, we should not pass through this road at this time. Finally, we select a stacked line chart view (④) to visualize the traffic status in the temporal dimension. As we can see from the line chart, there are both morning peaks from 7:00 to 9:00 and evening peaks from 16:00 to 18:00, which indicates that if we drive on the road at that time, it is highly possible to suffer traffic jams in Lyon.

Redesign map-oriented dashboard (Figure 6.2) by adding novel visualization techniques. Given that the current existing views in traffic control centers are difficult to provide detailed temporal changes information of traffic flows (*e.g.,* display how traffic flows of every road segment change over time), we merge the novel visualization techniques proposed in this

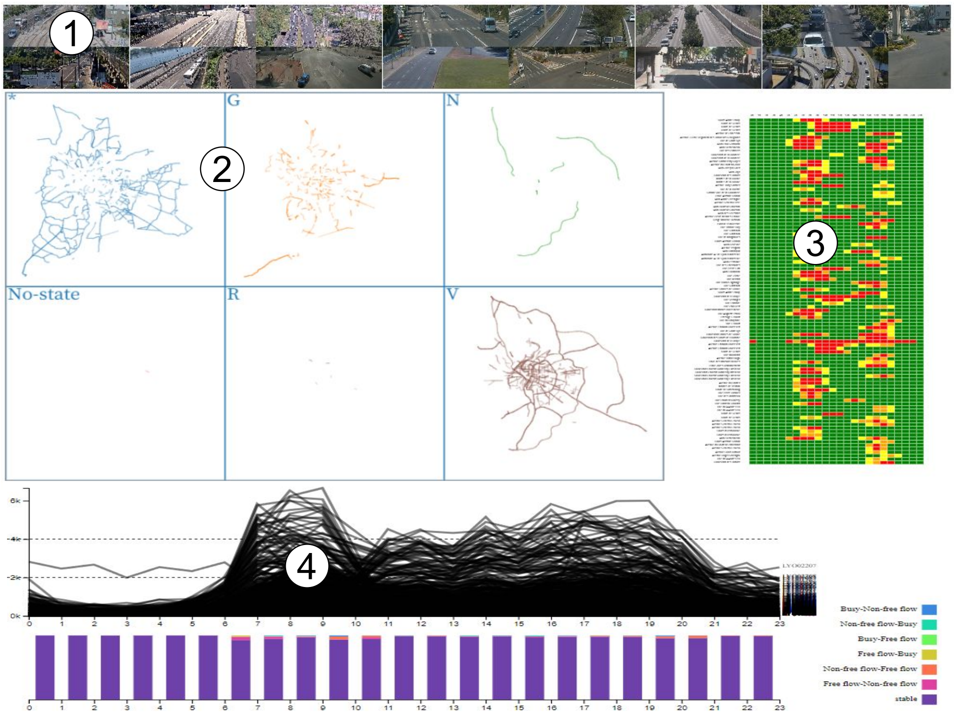


Figure 6.7: Redesign of a map-oriented dashboard (Figure 6.2) by adding novel visualization techniques. ①, a webcam view shows the road status in specific road intersections. ②, GRIDIFY shows the traffic status by dividing road segments in cells. ③, a rectangle matrix view shows the traffic flow changes over time for specific road segments. ④, GROUPSET shows how traffic flows change over time.

manuscript into a dashboard. To explore traffic flow temporal changing patterns, we utilize four views: webcam view, GRIDIFY (Chapter 4), rectangle matrix view, and GROUPSET (Chapter 5), as shown in Figure 6.7. In order to visualize the temporal changes in traffic flows, we first select GROUPSET (④) to visualize the traffic flow (**T1**) changing over time during a day. It shows that the traffic flows in Lyon have two peaks: morning peak around 7:00 and evening peak around 17:00 (**T2**). Nevertheless, we still believe some roads only have morning peak or evening peak so that we can guide the vehicles passing through the roads during their not busy time to make the road network have a balanced load. Thus, we select the rectangle matrix as a view (③) to visualize how traffic flows change during a day in specific road segments. In this view, the X-axis represents the 24 hours of a day, and the Y-axis represents the different road segments. Therefore, we can observe the major road status distribution over time. This view shows the high traffic density in red and the low traffic density in green. We can observe that some roads do not have morning peaks, and some do not have evening peaks (**T4**). It shows the opportunity to rebalance road traffic with some intervention. For example, traffic operators can guide more vehicles or public transport to these roads that do not have morning peaks from 6:00 to 8:00am. Finally, we choose GRIDIFY (②) and webcam (①) view. The webcam view helps traffic operators observe the traffic status among specific road intersections (**T2**). GRIDIFY shows the different traffic situations in cells (On Figure *N* refers to server traffic congestion, *R* to traffic congestion, *O* to not fluent, *V* to fluent, and *G* to unknown). The interesting thing is that three long road segments in cell *N* refer to traffic congestion (**T2**). It is different from our common sense since we usually think the city center easily happens traffic congestion. However, these three road segments are not in the city center of Lyon.

6.1.4 Discussion on Traffic Control Centers Deployment

The previous section has introduced two cases explaining how ControlCenter implements dashboards in traffic control centers to achieve specific monitoring goals. However, two challenges have not been addressed in these two cases: 1) how to coordinate the visualization techniques in the dashboard and 2) how to make connections among these visualization techniques. These two challenges could be our future research directions.

"How to guide traffic operators to create a helpful dashboard in traffic control centers?" Previously, we have introduced the views (*e.g.,* `map` and `webcam`) in the dashboard of traffic control centers. However, we do not discuss where these views should be put in a dashboard. This is a **layout** issue formulated as a problem — *"Which layout strategy is most beneficial to achieve the tasks?"*

In the InfoVis community, this problem is defined as the **Multiple Coordinated Views (MCV)** problem we have studied in Section 2.4.5. MCV is a visualization environment that consists of two or more distinct views to support exploring a single conceptual entity, which is a specific visualization technique in that users can understand the heterogeneous data and view them through different representations. The design space of MCV affects the availability and efficiency of realizing tasks. Chen *et al.* study the design space of MCV in composition and configuration [123]. The composition describes how many views they use and which presentation type each view is. They summarize some guidelines for multiple view designs. Besides, the MCV can merge other visualization to construct new design spaces to improve the visual presentation. Roberts *et al.* link the multiple views to the 3D visualization to explore the deeper patterns [120] since it is not easy to 'see inside' in the 3D visualization when dealing with too much data.

Thus, to address this challenge, we plan to conduct an empirical study to analyze the strategy of the layout of the current traffic control centers. By doing this, we will first collect the pictures of dashboards in traffic control centers on the internet or by taking pictures in the control centers. Then, we will annotate the layout strategies and the views contained in these dashboards. Eventually, we will analyze the layout strategies and design the guidelines for recreating the dashboard in traffic control centers.

"How to connect each visualization technique in the dashboard of traffic control centers?" In this manuscript, we have discussed how to deploy visualization techniques in traffic control centers. However, a challenge still exists while combining multiple visualization techniques in a dashboard since we not only put them in the same design space but also make the connection among them in a workflow. This means we should unify the data parameters to allow visualization techniques to interact with each other (*e.g.,* to share a highlighted element or time interval). We argue that unifying the configuration format is the most important step for aligning all visualizations states. Such configuration would also enable the rapid export and sharing of the parameters, which makes it possible to interoperate among the multiple visualization techniques in a dashboard.

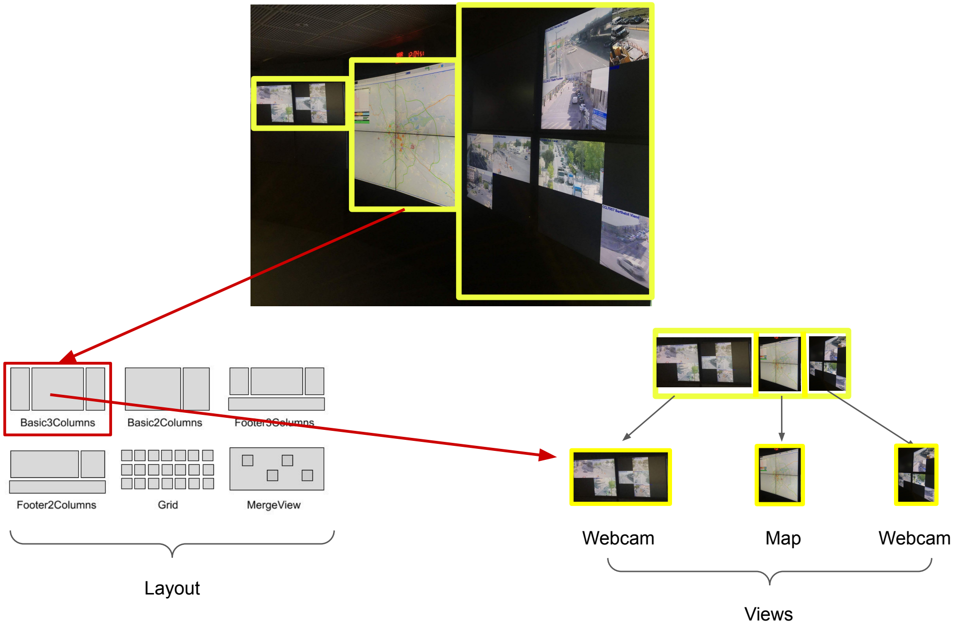


Figure 6.8: Dashboard, layout, and view examples. The up one is the dashboard of the traffic control center, edited based on a picture taken from the traffic control center in Lyon. The control center belongs to `Basic3Columns` layout type, containing two `webcam` views and one `map` view.

We explain how the visualization techniques can be connected by introducing the configuration format of FuzzyCut, which are as follows:

```
{
  "Title": "Taxi speed",
  "Attribute": "0",
  "Parameters": [{
      "n": "3",
      "Core": "13.6",
      "Support": "27.2",
      "Names": [
        "Low",
        "Middle",
        "Hig"
      ]}]}
```

where a configuration file of FuzzyCut captures all the design and visual abstractions parameters (*e.g.*, the `title`, `n`, `core`, `support`, and `names`). These parameters can adjust the display of the visualization techniques. For example, changing the parameter of FuzzyCut's configuration file can adjust the shape of FuzzyCut (the membership function). Thus, the configuration file is the key to communicating among different visualization techniques, which can help the dashboard in traffic control centers connect all the views and interoperate among these views.

To summarize, deploying the traffic control centers are facing two challenges, and we will address them as follows:

- Conducting an empirical study helps explore the layout strategies of traffic control centers, hence, concluding the guidelines for the control center design;
- Through the configuration, we will unify the input parameters of each visualization technique. Thus, the visualization techniques can communicate necessary information to each other, allowing interoperability among these visualization techniques.

6.2 Applications Beyond Road Traffic Data and Context

This section discusses the applications of our three visualization techniques (FuzzyCut, Gridify, and GroupSet) to other domains and contexts of use. We picked those domains based on

Visualization techniques	**Datasets**	**Data types**	**Data characterization**
FuzzyCut	**Taxi speeds**	Quantitative values	Half-bounded interval ($x \in [0, +\infty]$), continuous
	Temperature	Quantitative values	Unbounded interval ($x \in [-\infty, +\infty]$), continuous
	Age	Quantitative values	Half-bounded interval ($x \in [0, \infty]$), integer
	Displacement	Quantitative values	Unbounded interval ($x \in [-\infty, +\infty]$), continuous
	Computational time	Quantitative values	Half-bounded interval ($x \in [0, \infty]$), continuous
Gridify	**Taxi**	Spatial data	Geo-coded entities, geographic coordinate system [a]
	Transit accessibility	Spatial data	Geo-coded entities, geographic coordinate system
	Trade	Spatial data	Geo-coded entities, geographic coordinate system
	Soccer players	Spatial data	Geo-coded entities, Cartesian coordinate system [b]
GroupSet	**Traffic density**	Time-varying data	Half-bounded interval ($x \in [0, +\infty]$), continuous
	Soccer rankings	Time-varying data	Half-bounded ($x \in [1, +\infty]$), integer
	MNIST classification	Category-varying data	Bounded interval ($x \in [0, 1]$), continuous

Table 6.1: Summary of datasets implemented in visualization techniques proposed in this manuscript. The bold texts refer to the road traffic-relevant datasets. All the datasets implemented in FuzzyCut are quantitative values, and all the datasets implemented in Gridify are spatial data. However, the datasets implemented in GroupSet are either time-varying or category-varying data.

[a] A spherical or ellipsoidal coordinate system for measuring and communicating positions directly on the Earth as latitude and longitude.

[b] A coordinate system that specifies the positions of objects with a pair of numerical coordinates.

the similar challenges they offer (in terms of space, time and space-time visualizations), and the availability of the datasets (listed in Table 6.1). The context of use has been picked to show the techniques used beyond control centers and dashboards, *e.g.,* as an interactive legend in a visualization application.

6.2.1 Expanding the Applications of FuzzyCut

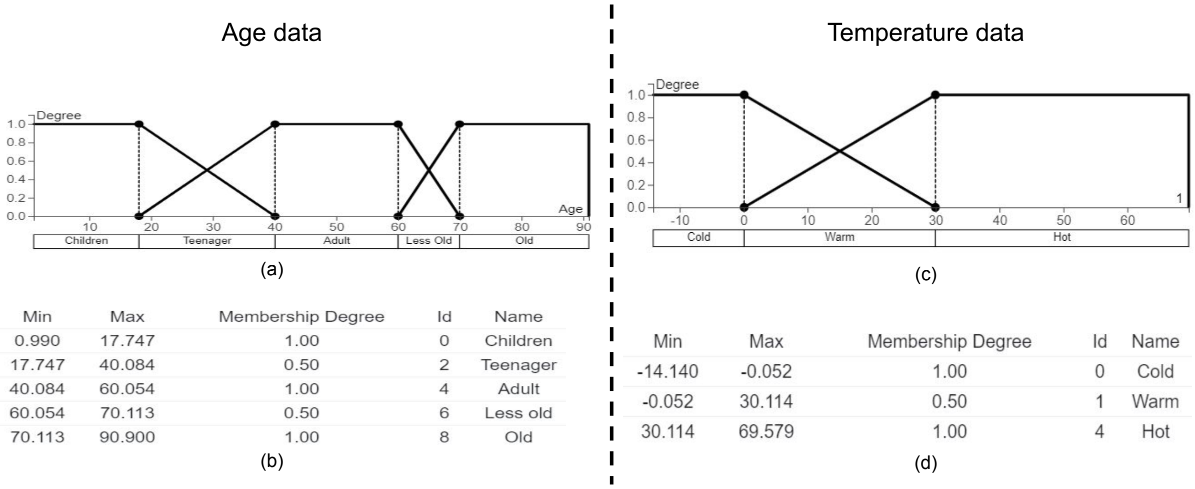
In Chapter 3, we have shown how FuzzyCut can be used to categorize taxi speed values. This section discusses the expanding applications of FuzzyCut with four datasets: age, temperature, engineering datasets, and computational time datasets (detailed information introduced in Table 6.1).

Age/ Temperature. This section introduces examples of how FuzzyCut categorizes the quantitative values except for traffic-relevant data — age and temperature data. Table 6.1 shows the detailed information of the datasets. Compared to the taxi speed data (introduced in Chapter 3), the age data type is discrete instead of continuous, and the temperature data has unbounded intervals instead of half-bounded intervals. This section explores the usability of FuzzyCut when the quantitative data differ from taxi speeds in data characterization.

We first implemented FuzzyCut with age data. As shown in Table 6.1, a typical age data consists of a half-bounded interval (*e.g.,* $[0,+\infty]$) and rounded values (*e.g.,* 3, 5, 7). There exists a minimum value but no fixed maximum value. The data we used, in this case, is the Simpsons, a popular U.S. TV Series. It includes 24 roles whose ages range from 0 to 90. Figure 6.9 (a) shows FuzzyCut implemented with age data where the x-axis represents age values. The y-axis shows the membership degree for every category. Table (b) in Figure 6.9 shows the categories and their properties (*e.g.,* membership values and the range of categories), which corresponds to the membership function. We can see that the membership degrees in the `Teenager` and `Less old` category are equal to 0.5. However, all the other categories' membership degrees equal 1.

We then implemented FuzzyCut with temperature data, as shown in (c) and (d) of Figure 6.9. As such temperature value does not have an exact maximum and minimum value, the data is unbounded and continuous (as shown in Table 6.1). It consists of temperatures from 357 divisions in the USA for 12 months. The temperature values range from -12.98 to 68.56. In (c), FuzzyCut categorizes the temperature values into three categories, shown with the x-axis. The y-axis refers to membership degrees. Category names are *Cold*, *Warm*, and *Hot*. Among them, a category named *Warm* is derived from *Cold* and *Hot*, with a membership degree of 0.5, as shown in (d) in Figure 6.9.

Engineering dataset. In the vibration control field, control strategy greatly influences performance. Compared with traditional control strategies, the fuzzy control strategy does not need explicit mathematical modeling, making it more applicable in complex modeling, in which assumptions and approximations are often made. An interactive visual technique to observe the changes in vibration status is highly necessary. In this case, domain experts can have an insight into fuzzy control strategy and reveal the influence of parameters, which is significantly meaningful for vibration control engineering. The dataset includes $1,024$ items with attributes *acc* (acceleration), *vel* (velocity), and *dis* (displacement). In this case, we use FuzzyCut to categorize attribute acceleration values. Compared to taxi speed values, acceleration



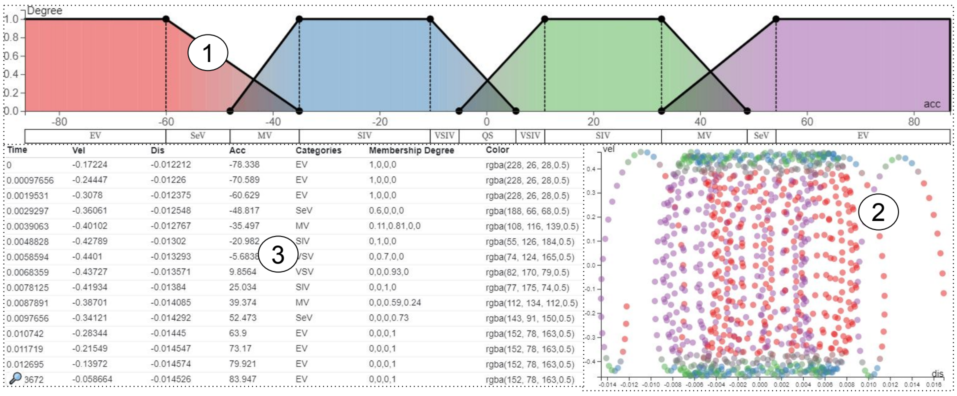
Min	Max	Membership Degree	Id	Name
0.990	17.747	1.00	0	Children
17.747	40.084	0.50	2	Teenager
40.084	60.054	1.00	4	Adult
60.054	70.113	0.50	6	Less old
70.113	90.900	1.00	8	Old

(b)

Min	Max	Membership Degree	Id	Name
-14.140	-0.052	1.00	0	Cold
-0.052	30.114	0.50	1	Warm
30.114	69.579	1.00	4	Hot

(d)

Figure 6.9: The expanding applications of FuzzyCut with age and temperature data. Age data are from 0 to 90. There are three main categories named *Children*, *Adult*, and *Old*. Meanwhile, two fuzzy categories were created, and their names are *Teenager* and *Less old*. In temperature data, there are two main categories named *Cold* and *Hot*. There is an intermediate category named *Warm* between them.

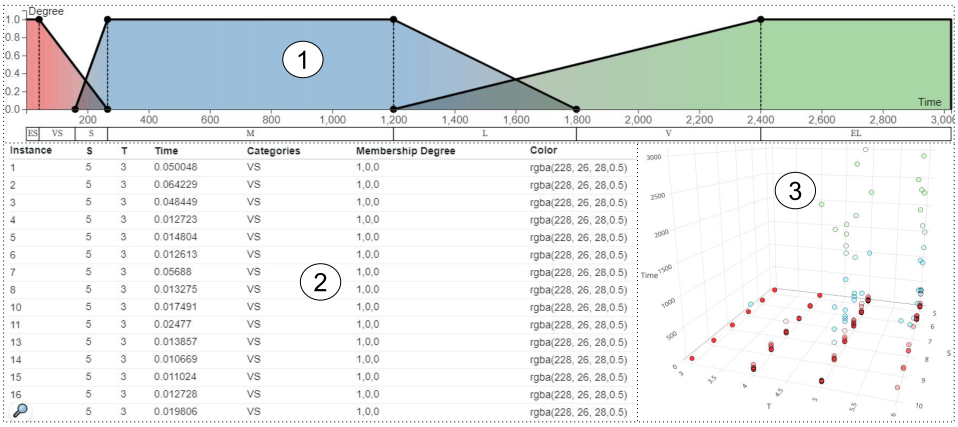


Time	Vel	Dis	Acc	Categories	Membership Degree	Color
0	-0.17224	-0.012212	-78.338	EV	1,0,0,0	rgba(228, 26, 28,0.5)
0.00097656	-0.24447	-0.01226	-70.589	EV	1,0,0,0	rgba(228, 26, 28,0.5)
0.0019531	-0.3078	-0.012375	-60.629	EV	1,0,0,0	rgba(228, 26, 28,0.5)
0.0029297	-0.36061	-0.012548	-48.817	SeV	0.6,0,0,0	rgba(188, 66, 68,0.5)
0.0039063	-0.40102	-0.012767	-35.497	MV	0.11,0.81,0,0	rgba(108, 116, 139,0.5)
0.0048828	-0.42789	-0.01302	-20.982	SIV	0,1,0,0	rgba(55, 126, 184,0.5)
0.0058594	-0.4401	-0.013293	-5.6838	VSV	0,0.7,0,0	rgba(74, 124, 165,0.5)
0.0068359	-0.43727	-0.013571	9.8564	VSV	0,0,0.93,0	rgba(82, 170, 79,0.5)
0.0078125	-0.41934	-0.01384	25.034	SIV	0,0,1,0	rgba(77, 175, 74,0.5)
0.0087891	-0.38701	-0.014085	39.374	MV	0,0,0.59,0.24	rgba(112, 134, 112,0.5)
0.0097656	-0.34121	-0.014292	52.473	SeV	0,0,0,0.73	rgba(143, 91, 150,0.5)
0.010742	-0.28344	-0.01445	63.9	EV	0,0,0,1	rgba(152, 78, 163,0.5)
0.011719	-0.21549	-0.014547	73.17	EV	0,0,0,1	rgba(152, 78, 163,0.5)
0.012695	-0.13972	-0.014574	79.921	EV	0,0,0,1	rgba(152, 78, 163,0.5)
3672	-0.058664	-0.014526	83.947	EV	0,0,0,1	rgba(152, 78, 163,0.5)

Figure 6.10: Categorize acceleration values. FuzzyCut creates 11 categories as shown in ①, where the *x*-axis refers to acceleration and the *y*-axis refers to membership degree. A scatterplot combines with FuzzyCut by acceleration values in ②, where the *x*-axis refers to displacement, the *y*-axis refers to velocity, and the colors refer to acceleration values. FuzzyCut derives new attributes such as **Categories**, **Membership Degree**, and **Color** in ③.

values have unbounded intervals instead of half-bounded intervals (introduced in Table 6.1).

In order to help domain experts observe vibration status, we designed a prototype by combining a scatterplot (Figure 6.10(②)) with FuzzyCut (Figure 6.10(①)). As shown in ①, we created 4 trapezoids (*Full Category*) named **ExV (Extremely Vibration)**, **SlV (Slight Vibration)**, **SlV (Slight Vibration)**, **ExV (Extremely Vibration)**. Besides, there are also three *Overlap Category* generated, named **MV (Moderate Vibration)**,**QS (Quasi Static)**, and **MV (Moderate Vibration)**. The x-axis in the scatterplot (②) represents displacement and the y-axis refers to velocity, which is able to reflect the vibration status. The derived new attributes include *Categories*, *Membership Degree*, and *Color* shown in 174. By using FuzzyCut, it is



Instance	S	T	Time	Categories	Membership Degree	Color
1	5	3	0.050048	VS	1,0,0	rgba(228, 26, 28,0.5)
2	5	3	0.064229	VS	1,0,0	rgba(228, 26, 28,0.5)
3	5	3	0.048449	VS	1,0,0	rgba(228, 26, 28,0.5)
4	5	3	0.012723	VS	1,0,0	rgba(228, 26, 28,0.5)
5	5	3	0.014804	VS	1,0,0	rgba(228, 26, 28,0.5)
6	5	3	0.012613	VS	1,0,0	rgba(228, 26, 28,0.5)
7	5	3	0.05688	VS	1,0,0	rgba(228, 26, 28,0.5)
8	5	3	0.013275	VS	1,0,0	rgba(228, 26, 28,0.5)
10	5	3	0.017491	VS	1,0,0	rgba(228, 26, 28,0.5)
11	5	3	0.02477	VS	1,0,0	rgba(228, 26, 28,0.5)
13	5	3	0.013857	VS	1,0,0	rgba(228, 26, 28,0.5)
14	5	3	0.010669	VS	1,0,0	rgba(228, 26, 28,0.5)
15	5	3	0.011024	VS	1,0,0	rgba(228, 26, 28,0.5)
16	5	3	0.012728	VS	1,0,0	rgba(228, 26, 28,0.5)
	5	3	0.019806	VS	1,0,0	rgba(228, 26, 28,0.5)

Figure 6.11: Categorize computational time values. FuzzyCut creates seven categories, as shown in ①, where the *x*-axis represents the time and the *y*-axis represents the membership degree. A 3D scatterplot combines to FuzzyCut by the computational time values, where the *x*-axis, the *y*-axis, and the *z*-axis are *T* (the number of tiers of a bay), *S* (the number of stacks of a bay), and computational time respectively. The colors in the scatterplot correspond to the colors in ①, which refer to computational time. FuzzyCut derives new attributes shown in ②.

easy to categorize acceleration and monitor which parts are unstable. Also, membership degrees connect to the generated categories to help users understand the confidence, which is also useful information for users to re-categorize acceleration values.

Computational time dataset. Computational time plays a very important role in the efficiency of an algorithm for solving combinatorial optimization problems. This case study categorizes the computational time for solving the Stochastic Container Relocation Problem (SCRP), which aims to retrieve all containers from a bay with the minimum number of relocations, a typical combinatorial optimization problem arising in container port operations. The data is collected from [232], an optimization problem calculation for minimizing the number of relocation of containers, which includes the computational times for solving the SCRP by an exact algorithm on 288 instances over 24 problem sets. A problem set is characterized by two parameters/attributes: *S* and *T*. *S* represents the number of stacks of a bay, and *T* represents the number of tiers of a bay. The computational time increases dramatically with the scale of the problem, and as a result, some problems cannot be solved to optimal within one hour. However, some problems can be solved in several milliseconds. Thus, evaluating the samples based on categorized computational times (introduced in Table 6.1) allows researchers to choose the correct way to solve this problem quickly.

As shown in Figure 6.11, ① illustrates that we create three main categories, which correspond to small-scale problems, medium-scale problems, and large-scale problems, respectively. Usually, the problems that can be solved optimally within five minutes are considered small-scale problems, while those that cannot be solved optimally within one hour are regarded as large-scale problems. However, for the SCRP, we cannot simply differentiate the problem scales by the computational time because it is observed that for the instances in the same problem set, their computational times can vary significantly. Therefore, to better understand the computational complexities of these problem sets, we would like to make a more detailed classification of the instance scales. In this case study, we classify the instances into seven categories with membership degrees based on their computational times. Making such detailed classifications is because the researcher is also concerned with the extent to which the instances can be solved fastly or slowly.

This technique is beneficial in several ways. Firstly, the researcher has a clear picture of the distribution of the computational times of each problem set, as shown in Figure 6.11(③), which enables her to have a more accurate understanding of the computational complexities of different problem sets. Secondly, the researcher can easily identify the problem sets that are very difficult to be solved and thus can turn to heuristic algorithms to obtain a near-optimal solution in a reasonable time. Lastly, the 3D scatterplot in Figure 6.11(③) enables one to gain a visual impression of the impacts of the width (S) and the height (T) of a bay on the computational efficiency of the SCRP, which is useful for the port operators to plan the dimension of the bay in the storage yard of container ports.

The open-source code enables further use of the technique by simply uploading quantitative data in our web app or by including the module in a Notebook using `import {fuzzyCut} from '0820d2ad9cfa734d'` in a JavaScript-based environment.

6.2.2 Expanding the Applications of Gridify

This section discusses how Gridify can be applied to other application domains. In Chapter 4, we have introduced the application of Gridify with two traffic-relevant data: taxi datasets and public transport accessibility datasets (introduced in Section 2.1.5). However, Gridify is designed to analyze any geo-coded dataset, such as trade data among countries and soccer players moving on the playground (listed in Table 6.1).

Trade dataset. In economics, there is a wealth of geo-related datasets, *e.g.,* countries bilateral flows of goods and services (we further refer to as *products*, introduced in Table 6.1). We explore a trade flows dataset using Gridify for 512 products (*e.g.,* fuel, cars) traded between 250 countries, from 1965 to 2015. Such dataset shows heterogeneous data types, as well as some derived and inconsistent values (*e.g.,* some values are missing even though this dataset has already been curated by the U.N. and economists [233]). Our process is to answer similar questions available from [233] following the *What, where, and when countries export?* scheme. Figure 6.12 (0) displays the overview of the dataset (A_s ⁙ $\emptyset$), which is highly cluttered (we introduced ⁙ and other notations in Section 4.3.2). Separation by country as an OD map using grid on Figure 6.12 (1) (A_s ⁙ BIN) $\bowtie$ (A_s ▦) preserves geo-connections by countries, but clutter is persistent: the reason is that countries have multiple connections over time (50 years), for each product (512), so potentially more than 2000 lines overlapping. So the next transformation will be to aggregated those mark, *e.g.,* using the `COUNT` to spot top exporters (Figure 6.12 (2a, 2b), grouped by sub-region).

Where does country A export? Answering this question requires selecting a particular country (*e.g.,* France) by clicking on it, for instance, starting from the previous visualization Figure 6.12 (2b, yellow square). One can filter by value (*e.g.,* filter out non-selected countries) using the query panel Figure 6.12 (3). Then the grid view focuses on the selected country Figure 6.12 (4), which is again cluttered as we break down an aggregation. So one starts over the abstraction process, similar to the first step, but using another strategy by using geo-grid

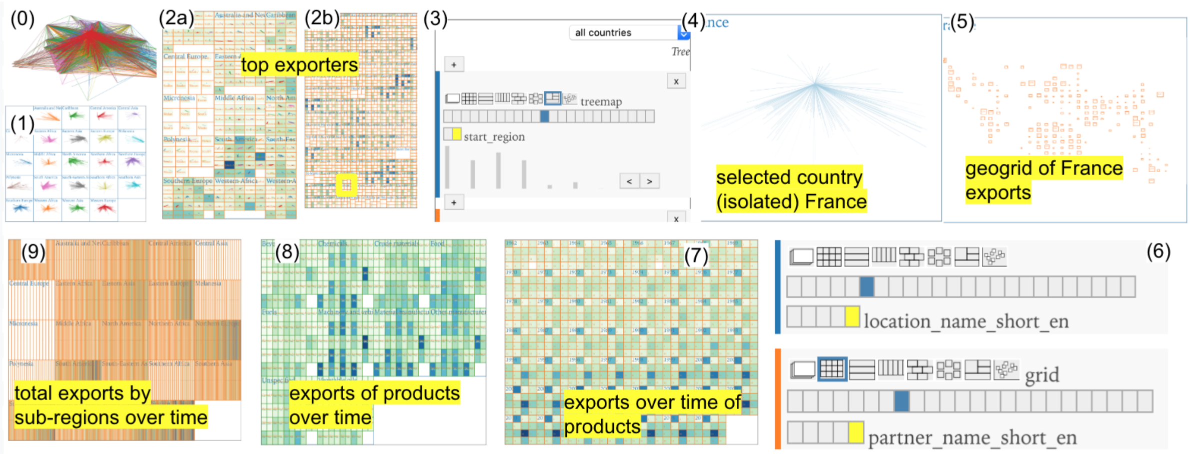


Figure 6.12: Trade dataset of 250 countries: (0) overview of global trade flow, (1) grid by countries while preserving the global map of exports (OD Map [31]), (2a) top exporters by sub-regions and (2b) by countries, (3) selected country available as a dimension, (4) selected country (France) dimension used to filter the dataset, (5) France trade partners as a geo-grid, (6) a trade partner (Germany is selected) and (7,8,9) represent bilateral trade flows between France and Germany over times by products using different visual strategies.

coordinates ($A_{geogrid}$ ⊞ $\emptyset$) to encode partner countries as cells which size reflects exports volume Figure 6.12 (5). Geo grids do not have clutter, so the user selects another country (*e.g.,* Germany) to explore bilateral flows to this country over time. Figure 6.12 (6) shows that the current query maintains two `SELECTED` values for different views on the dataset: one globally for the exporter country (France) and one at a level below for the importer (Germany). Following steps explore bilateral trade flows over time Figure 6.12 (7) and by products Figure 6.12 (8). The use of ▥ would be better suited to convey temporal change, but placeholders may be too small to convey aggregated values Figure 6.12 (9). The same exploration process could apply to the series of questions *Which country exports products C?*, *Where does country A export products C? Which products are top exports?*

Soccer dataset. Sports generate a growing source of spatio-temporal connections [234]. In particular, a tool called SoccerStories [235] uses geo-coded data to visually present sequences (*e.g.,* consecutive series of passes for the same team) as a clustered graph on the soccer field. SoccerStories already implements the multi-faceted visualization approach by dividing entities and their explicit connection using a *phase* attribute. We use the same underlying dataset provided by Opta–a sports data tracking company–where each pass has 12 values (shot, pass, ..). Each value may have up to 50 *qualifiers*, which are non-structured attributes, and specific to each type of pass. We pre-processed the dataset to the most common passes and focused on a La ligua game between FV Barcelona and Real Madrid (2-2) on Oct. 7th, 2012, for a total of $1,622$ events. The coordinate plot of positions Figure 6.13 (0) first shows some data quality issues: entities have missing positions (*e.g.,* no destination entity for a pass, so it is set to $(0,0)$). However, as we are interested in connections, we will keep all of them and build aggregates, *e.g.,* by the number of events occurrence grouped by `team` ⋈ `role` ⋈ `player` ⋈ `time period` Figure 6.13 (1). This very simple construction provides a wealth of information not only

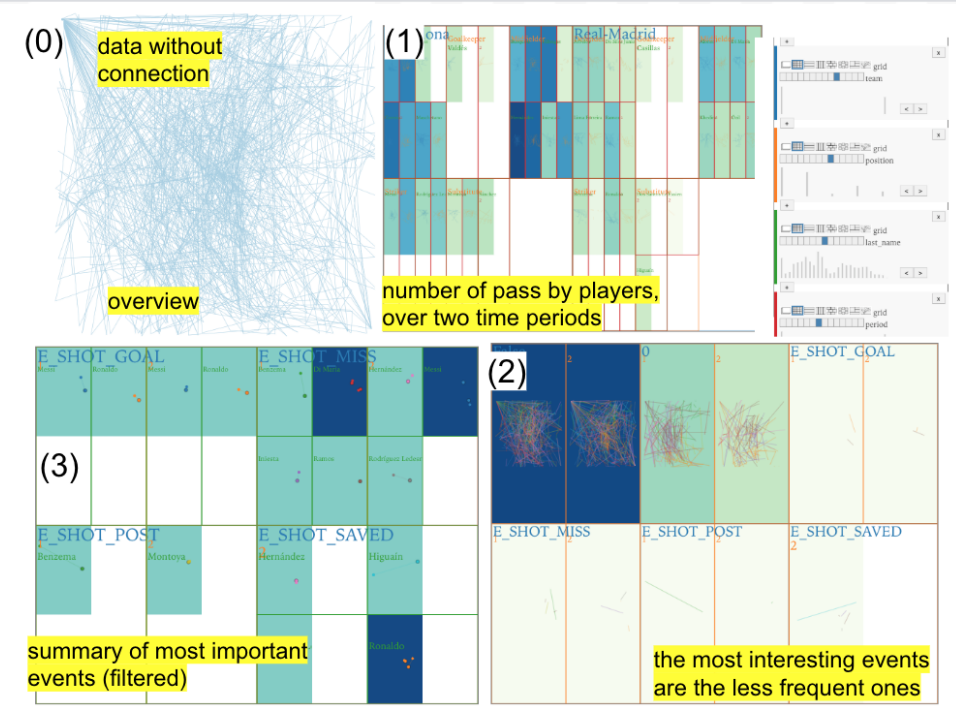


Figure 6.13: Passes during a soccer game between two teams (total 22 players) reveal teams strategy (Barcelona has a pass-oriented game) as well as players diversity of passes and their importance.

specific to the game (*e.g.,* Alves only played the first half of the game for Barcelona), but also the style of the teams (Barcelona ball possession is key, and they generate many passes), and individual players (midfielders in Barcelona are heavily involved, contrary to Real Madrid). Figure 6.13 (2) shows however a limit of using Gridify: the most interesting events are not the frequent ones (*e.g.,* goal shots). There only are a few of them compared to the other types of passes. This is how SoccerStories built sequences: starting from those interesting events, which usually are a handful (a dozen by games). However, by filtering out those interesting events, Figure 6.13 (3) provides a game summary showing the 4 goals of the game (Ronaldo and Messi) and key players who attempted to shoot.

To summarize, Gridify is designed for all geo-coded data, including geographic coordinate system data (*e.g.,* taxi data, trade data, and transit accessibility data) and Cartesian coordinate system data (*e.g.,* soccer player data), as shown in Table 6.1. We have implemented these geo-coded datasets, and the code has been released in Observable notebook (Link), so one can check the implementation of datasets and also upload their own datasets.

6.2.3 Expanding the Applications of GroupSet

This section discusses how GroupSet works in other domains except for the traffic density changes over time. In Chapter 5, we have introduced how GroupSet helps analyze and identify the traffic density changing patterns over time, focusing on the time-varying data. However, GroupSet is designed not only for analyzing traffic density changes. In this section, we explain how the GroupSet can be used in the soccer rankings datasets and MNIST classifi-

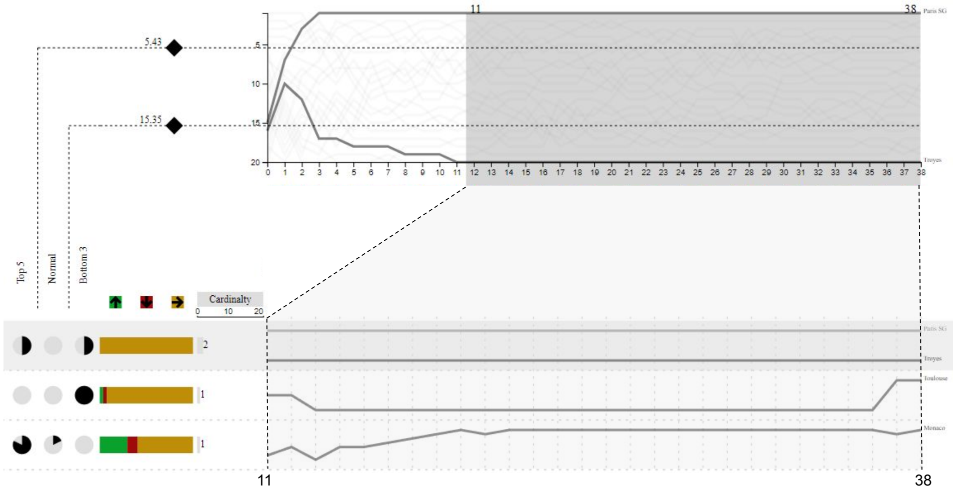


Figure 6.14: Discarded time intervals in soccer data using the brush feature on the global line chart (to keep time points between day 11 and day 38). As a result, compared to the global line chart on top, the below group line charts do not include the time interval from day 0 to day 10. Groups are then ranked by teams with fewer changes of sets (resulting in both the best and worst teams).

cation datasets (A handwritten digits dataset for training imaging process system and algorithm [223]). Table 6.1 contains all the detailed information of these datasets implemented in GroupSet. One special thing is that, compared with other datasets, the MNIST classification datasets are category-varying data instead of time-varying data.

Applications of GroupSet in soccer rankings dataset. Soccer data analysis is becoming increasingly important as data is now available to data scientists. We have already used a soccer dataset in Gridify (introduced in Section 6.2.3), but over spatial attributes for a specific game: we now explore temporal attributes over multiple games. In particular, game outcomes are indicators of teams' performance that can be aggregated and represented over a season. The standard representation is a ranking of usually around 20 teams (for European leagues) over time 19 × 2 games, *i.e.* 38 games. We create three sets {TOP5, NORMAL, BOTTOM3} which rank corresponds to the teams qualified for European competitions, the non-qualified teams, and teams relegated to the inferior division (**T1**). The NORMAL group usually is not the aim of the top-performing teams, but it is for less-performing teams (which usually never get to the TOP5 group).

Discarding early ranking variability. An essential property of permutations in soccer championships is that at the beginning, teams may change rank with a higher probability than later in the championship (when there are high points differences). So it is essential to discard the first days as shown in Figure 6.14 by selecting the time interval from *e.g.,* day 11 to the end as an analyzed period (**T6**). Once the beginning period is discarded, we sort the soccer teams by trend to find the most stable soccer teams. We notice that three sets are being more stable than other sets, which contain four teams (Figure 6.14): Paris, Troyes, Toulouse, and Monaco (**T2**). The most important thing is that these four teams are either stable at the top or bottom levels. Especially for Paris and Troyes, they did not change their rank from 11 to 38, and they are ranked first and last (**T5**).

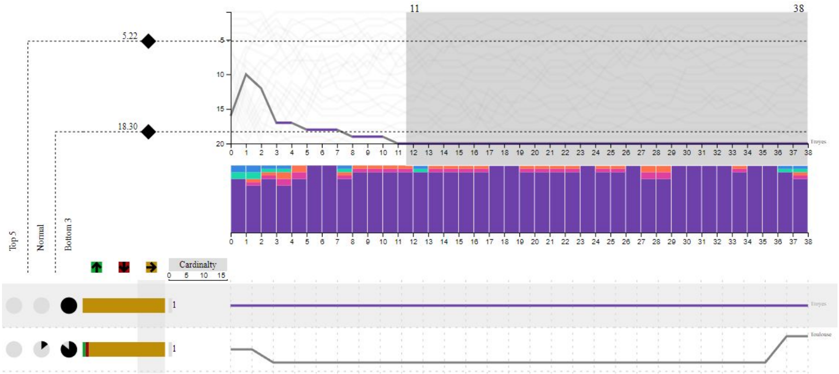


Figure 6.15: Understanding the less performing teams. They are the ones that had the worst rank for most of the season, except for Toulouse whose rank increases during the last games.

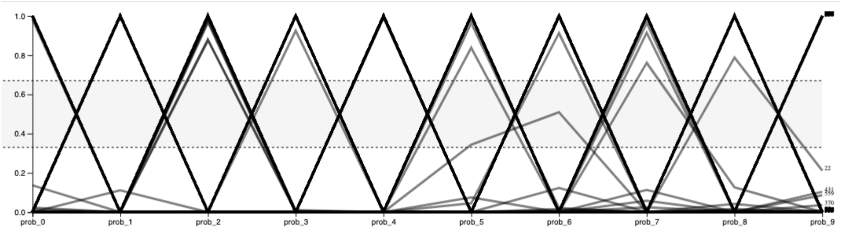


Figure 6.16: Classification results of $1,000$ handwritten images. The x-axis refers to predicted classes, and the y-axis refers to the predicted possibility. A line refers to the classification probability of a single image among these predicted classes.

Identifying best and worst performing teams. The next step is to answer basic performance questions such as which teams are the best or worst, which can immediately be seen on the championship's last day. However, we may dig into the analysis to check if this has been the case for the whole season. The rank by set membership degree shows teams that do not change groups, so they have always performed very well or very poorly (Figure 6.15). We aggregate the soccer teams by *Degree* and then sort them by *Degree-Bottom 3* so as to get the soccer teams with the highest membership to the `Bottom 3` on top (and the ones with partial membership to this group below) (**T5**). The team Troyes is the only one always being in the `Bottom 3` from 11 to 38, even though its rank raises at the very beginning of the season (but are discarded with our initial temporal selection).

Applications of GroupSet to category-varying data. This section discusses the application of how GroupSet analyzes the categories-varying data instead of time-varying data. We try to use GroupSet to analyze the classification results of MNIST datasets, a popular machine learning dataset of handwritten digits.

Training models to classify handwritten images of digits is one of the classic machine learning tasks. A frequent task is to check the result of the classification process, which is the grouping of an input dataset of $|E| = 1000$ elements which are images into $|T| = 10$ classes (digits ranging from 0 to 9) as in this case the instances are categories-varying data (and not time-varying data). While they are independent, we order them by ascending order. The sets here are $X = \{LOW, MEDIUM, HIGH\}$ the thresholds we set to assess if a classification

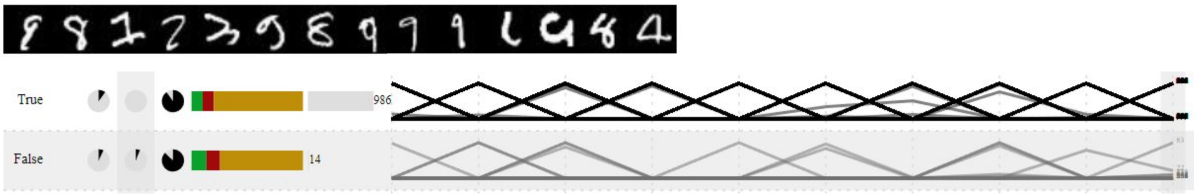


Figure 6.17: Overview of misclassification. Two aggregated groups are `True` set referring to correct classification and `False` set referring to misclassification. There are 14 handwritten images in the wrong classification, and 6 of them are digit 9, which is the most one.

Dataset	$\lvert E\rvert$	$\lvert T\rvert$	$[min(Q), max(Q)]$	S names
	elements	instances		
Road traffic	1,334	24	[0, 265]	{FREE FLOW,NON-FREE FLOW,BUSY}
Soccer rankings	20	38	[1, 38]	{TOP5, MIDDLE, BOTTOM3}
MNIST Image classification	1,000	10	[0, 81]	{LOW,MIDDLE,HIGH}
Student grades	23	17	[0, 14]	{GOOD,VERY GOOD, EXCELLENT}
University rankings	100	3	[0, 15]	{TOP10,TOP50,TOP100,OTHER}

Table 6.2: List of datasets (and their properties) implemented in GroupSet. We do not introduce the *Student grades* datasets and *University ranks* datasets in this manuscript, but their implementations are in GroupSet and one can check with the link.

probability assigns the image to the class. In this scenario, we uniformly split the interval to create categories (*e.g.,* split into three categories).

Frequent misclassification. Investigating misclassification and the reason is an important topic in machine learning research. GroupSet can support machine learning experts in observing and testing the classification results of models interactively. Most classifiers predict the results represented with possibility, such as the line chart shown in Figure 6.16 where the x-axis refers to the classes, and the y-axis refers to the possibility. The line chart shows that most possible values are around 1 or 0.

Using GroupSet, we can aggregate these samples based on specific attributes. As shown in Figure 6.17, we aggregate samples by *eval*, which refers to the `FALSE` or `TRUE` results. By hovering the `FALSE` group, we have an overview of these images of digits. There are 14 samples with misclassification, and 6 of them are digit 9. Thus, analyzing digit 9 is an interesting point that can help machine learning experts refine the training model.

Now we analyze the misclassification of digit 9, as shown in Figure 6.18. First, we create

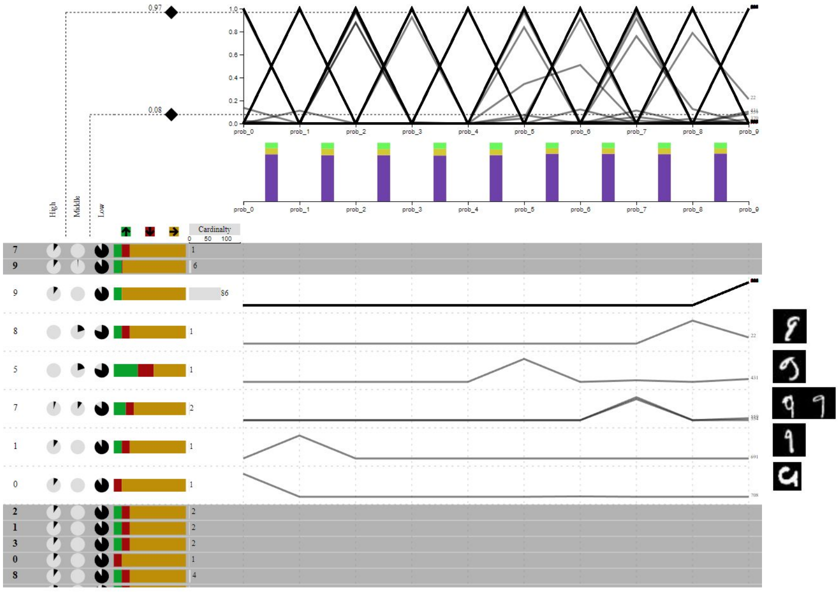


Figure 6.18: The misclassification of handwritten digit 9. There are six samples of digit 9 with misclassification displayed on the right side of each row. The rows (expanded first-level group of digit 9) represent prediction results (*e.g.,* the first row represents 9 is predicted to be 9 and the second row represents 9 is predicted to be 8). The last row shows a digit 9 is predicted to be 0. It might be because the 'o' of digit 9 is too big.

three categories named `High`, `Middle`, and `Low`. We then aggregate the samples with *label*, which refers to the real values of digits, so we have the ten first-level groups. However, it cannot display classified situations (*e.g.,* true or false) with only one aggregation. Thus, we aggregate the samples again with *pred*, which refers to classified results. Finally, we collapse all the first-level groups except digit 9. There are several misclassification rows of digit 9, as shown in Figure 6.18. Combined with the image visualization shown on the right side of Figure 6.18, we can observe why the misclassification happened. For example, digit 9 is classified as digit 0 because the 'circle' of digit 9 is too large.

Expanding the application of GroupSet in the category-varying data is possible. Especially for the classification data introduced in this section, users can highlight the groups they want through aggregation in the combination matrix, as shown in Figure 6.18. Besides, we have also implemented other datasets with GroupSet (Link). Table 6.2 lists all these datasets and corresponding information. In the future, exploring GroupSet's possibilities in other domains and other data types would be exciting and challenging.

6.3 Open Challenges

To conclude this manuscript, we now list a series of open challenges that remain to be addressed.

Visualizations onboarding and interoperability. We have introduced several techniques to support visual analysis for road traffic data and other application domains. Each technique has a learning curve to decode the visualization to interpret and interact with the data. Such a gap requires time and perseverance, which experts may not need, especially in critical, time-sensitive contexts. A solution is to design visualization as building blocks [236] with basic features shared across techniques. Despite we implemented such an approach (*e.g.,* with the FuzzyCut intervals creation feature included in GroupSet), most visual elements are specific to each design. Another path for *interoperability* is to let users use similar preferences across visualizations (*e.g.,* default dataset, grouping methods, etc.) in a programming language agnostic way. We already used such an approach (*i.e.* for views specification in the Control Center Dashboard) now popular in the visualization community using Domain Specific Language (DSL) [237], which is the mini-language tailored for a specific domain that allows programs to be implemented at the level of abstraction. Vega [173] specification is such a DSL, and in the future, we will align our DSL with Vega's for further visualization interoperability.

Data quality assessment and fixing. Real-world datasets usually suffer from data quality, as some information may either be missing or erroneous (*e.g.,* road sensor or malfunctioning), which may lead analysts to the wrong conclusion that a road may seem empty while it can be congested without reliable data. Thus there is a need for a visualization technique to convey such data quality issues in a visual way. In this manuscript, we showed it could be used with Gridify when detecting missing coordinates. For temporal techniques, however, it is challenging, especially if values are temporarily distorted due to an external factor (*e.g.,* humidity due to weather) that are not captured in the dataset. Visualizations should be able to provide information for analysts either to understand the level of reliability, for example, by informing on the data source reliability and contextual factors (*e.g.,* weather). Mechanisms to fix such data based on corrections (*e.g.,* temporal smoothing or distortion corrections) may be also be implemented as parts of the interaction features to be used without any technical knowledge.

Scalability in number of items and dimensions. Finally, our techniques have been used to explore either univariate data or geo-coded attributes with up to N dimensions (with N that cannot exceed the dozens of attributes; otherwise, the grid would be too small). Also, our techniques have been used for thousands of data elements. However, real-world datasets could be used with a huge number of dimensions and elements. In terms of representation, a solution would be to use larger screens such as multi-screen displays to have more pixels to encode information. Another solution is to provide multi-level aggregation methods that group data by data distributions and progressively load data items for finer grain analysis. This has implications on both the design of the technique to convey that data have been partially loaded, as well as a back-end (server) infrastructure to implement progressive data structures [238].

Bibliography

[1] "Google Maps," Road layer, Lyon, France, 2022. vii, 2

[2] Caroline Girardon, "Lyon est plus embouteillée qu'avant le Covid-19 à des niveaux extrêmes ," 20 minutes, Nov. 2021. vii, 2

[3] Stefan Thiesen, "Flir one infrared Street Views," YouTube, Feb. 2017. vii, 8

[4] F. Pozzebom/ABr, "Português: Brasília - Polícia Rodoviária Federal usa novo radar com microdigicam para capturar imagens na fiscalização das rodovias durante a terça-feira de Carnaval de 2007 — Wikipedia, the free encyclopedia," Feb. 2007. vii, 8

[5] Bidgee, "English: Traffic camera on a traffic light pole at the Lake Albert Road and Sturt Highway intersection in Wagga Wagga — Wikipedia, the free encyclopedia," Jul. 2008. vii, 8

[6] "File:Traffic lights — Wikipedia, the free encyclopedia." vii, 8

[7] N. Dilmen, "English: Ultrasonic sensor — Wikipedia, the free encyclopedia," Jan. 2014. vii, 8

[8] Hustvedt, "English: Pneumatic tubes at a bank — Wikipedia, the free encyclopedia." vii, 8

[9] Omegatron, "Schematic symbol and electronic model for a piezoelectric sensor — Wikipedia, the free encyclopedia," Dec. 2007. vii, 8

[10] NASA, "English: The magnetometer, mounted on a long boom to keep it far away from the RTGs measures magnetic fields in space and in the vicinity of Jupiter. The electronics at the left of the magnetometer are mounted in the spacecraft — Wikipedia, the free encyclopedia." vii, 8

[11] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, "Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2149–2158, Dec. 2013. viii, 19, 23, 24, 25, 26, 27

[12] Matthew W. Chwastyk, "The World's Congested Human Migration Routes in 5 Maps," National Geographic, Sep. 2015. viii, 25

[13] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko, "Stacking-Based Visualization of Trajectory Attribute Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2565–2574, Dec. 2012. viii, 19, 22, 24, 25, 31

[14] A. Anwar, T. Nagel, and C. Ratti, "Traffic Origins: A Simple Visualization Technique to Support Traffic Incident Analysis," in *2014 IEEE Pacific Visualization Symposium*. IEEE, Mar. 2014, pp. 316–319. viii, 24, 25, 32, 83

[15] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan, "TripVista: Triple Perspective Visual Trajectory Analytics and its application on microscopic traffic data at a road intersection," in *2011 IEEE Pacific Visualization Symposium*. Hong Kong, China: IEEE, Mar. 2011, pp. 163–170. viii, 24, 25, 26, 29

[16] M. Barry and Brian Card, "Visualizing MBTA Data," Github, Jun. 2014. viii, 24, 26, 27, 28

[17] J. Zhao, P. Forer, and A. S. Harvey, "Activities, ringmaps and geovisualization of large human movement fields," *Information Visualization*, vol. 7, no. 3-4, pp. 198–209, Sep. 2008. viii, 24, 26, 29

[18] R. Scheepens, N. Willems, H. van de Wetering, G. Andrienko, N. Andrienko, and J. J. van Wijk, "Composite Density Maps for Multivariate Trajectories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2518–2527, Dec. 2011. viii, 19, 22, 27

[19] A. Slingsby, J. Wood, and J. Dykes, "Treemap Cartography for showing Spatial and Temporal Traffic Patterns," *Journal of Maps*, vol. 6, no. 1, pp. 135–146, Jan. 2010. viii, 24, 27

[20] F. Wang, W. Chen, F. Wu, Y. Zhao, H. Hong, T. Gu, L. Wang, R. Liang, and H. Bao, "A visual reasoning approach for data-driven transport assessment on urban roads," in *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct. 2014, pp. 103–112. viii, 19, 20, 24, 28

[21] H. Liu, Y. Gao, L. Lu, S. Liu, H. Qu, and L. M. Ni, "Visual analysis of route diversity," in *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct. 2011, pp. 171–180. viii, 24, 28

[22] D. Guo, "Flow Mapping and Multivariate Visualization of Large Spatial Interaction Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1041–1048, Nov. 2009. viii, 24, 28, 29

[23] N. Cao, C. Lin, Q. Zhu, Y.-R. Lin, X. Teng, and X. Wen, "Voila: Visual Anomaly Detection and Monitoring with Streaming Spatiotemporal Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 23–33, Jan. 2018. viii, 19, 20, 24, 30

[24] W. Wu, Y. Zheng, N. Cao, H. Zeng, B. Ni, H. Qu, and L. M. Ni, "MobiSeg: Interactive region segmentation using heterogeneous mobility data," in *2017 IEEE Pacific Visualization Symposium (PacificVis)*, Apr. 2017, pp. 91–100. viii, 24, 30

[25] D. Seebacher, M. Miller, T. Polk, J. Fuchs, and D. A. Keim, “Visual Analytics of Volunteered Geographic Information: Detection and Investigation of Urban Heat Islands,” *IEEE Computer Graphics and Applications*, vol. 39, no. 5, pp. 83–95, Sep. 2019. viii, 24, 27, 29, 31, 32

[26] Z. Deng, D. Weng, X. Xie, J. Bao, Y. Zheng, M. Xu, W. Chen, and Y. Wu, “Compass: Towards Better Causal Analysis of Urban Time Series,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 1051–1061, Jan. 2022. viii, 24, 31, 32

[27] D. Liu, P. Xu, and L. Ren, “TPFlow: Progressive Partition and Multidimensional Pattern Extraction for Large-Scale Spatio-Temporal Data Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1–11, Jan. 2019. viii, 24, 31, 32

[28] G. J. Klir and B. Yuan, *Fuzzy sets and fuzzy logic: theory and applications.* Upper Saddle River, N.J: Prentice Hall PTR, 1995. viii, 36

[29] J. Wood, J. Dykes, and A. Slingsby, “Visualisation of Origins, Destinations and Flows with OD Maps,” *CARTOGR J*, vol. 47, pp. 117–129, May 2010. ix, 50

[30] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister, “UpSet: Visualization of Intersecting Sets,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1983–1992, Dec. 2014. xi, 68, 70, 74, 76, 79

[31] J. Wood, A. Slingsby, and J. Dykes, “Visualizing the Dynamics of London’s Bicycle-Hire Scheme,” *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 46, no. 4, pp. 239–251, Dec. 2011. xiii, 53, 64, 96

[32] M. Sweet, “Does Traffic Congestion Slow the Economy?” *Journal of Planning Literature*, vol. 26, no. 4, pp. 391–404, Nov. 2011. 1

[33] “INRIX — Wikipedia, the free encyclopedia,” Apr. 2022. 1

[34] H. Khreis, “Traffic, air pollution, and health,” in *Advances in Transportation and Health.* Elsevier, 2020, pp. 59–104. 1

[35] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann, Jan. 1999. 2

[36] J. W. Tukey, *Exploratory data analysis.* Reading, Mass., 1977, vol. 2. 2, 51, 68

[37] A. Kerren, J. Stasko, J.-D. Fekete, and C. North, *Information Visualization: Human-Centered Issues and Perspectives.* Springer Science & Business Media, Jul. 2008. 2

[38] C. Huberty and J. Morris, “Multivariate Analysis Versus Multiple Univariate Analyses,” *Psychological Bulletin*, vol. 105, pp. 302–308, Mar. 1989. 3

[39] C. D. Lloyd, *Exploring Spatial Scale in Geography.* John Wiley & Sons, May 2014. 3

[40] L. A. Zadeh, G. J. Klir, and B. Yuan, *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers.* World Scientific, 1996. 5, 40

[41] D. L. Gerlough and M. J. Huber, “Traffic flow theory: a monograph,” Transportation Research Board, National Research Council, Washington, Tech. Rep., 1975. 7

[42] L. Po, F. Rollo, C. Bachechi, and A. Corni, “From Sensors Data to Urban Traffic Flow Analysis,” in *2019 IEEE International Smart Cities Conference (ISC2)*, Oct. 2019, pp. 478–485. 8

[43] P. Beyer, “Non-Intrusive Detection, the Way Forward,” in *Proceedings of the 34th Southern African Transport Conference (SATC 2015)*, 2015, p. 10. 9, 10

[44] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. Anastasiu, and J.-N. Hwang, “CityFlow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-Identification,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* Long Beach, CA, USA: IEEE, Jun. 2019, pp. 8789–8798. 9

[45] S. Y. Cheung, S. C. Ergen, and P. Varaiya, “Traffic Surveillance with Wireless Magnetic Sensors,” in *Proceedings of the 12th ITS world congress*, vol. 1917, Oct. 2005, p. 14. 10

[46] “Global Positioning System — Wikipedia, the free encyclopedia,” Dec. 2021. 10

[47] Y. Ma, T. Lin, Z. Cao, C. Li, F. Wang, and W. Chen, “Mobility Viewer: An Eulerian Approach for Studying Urban Crowd Flow,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2627–2636, Sep. 2016. 10

[48] “Tesla Model S — Wikipedia, the free encyclopedia,” Jan. 2022. 11

[49] M. Schneider, “Automotive Radar – Status and Trends,” in *German microwave conference*, 2005, pp. 144–147. 11

[50] S. P. Hoogendoorn and P. H. L. Bovy, “State-of-the-art of vehicular traffic flow modelling,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 215, no. 4, pp. 283–303, Jun. 2001. 11

[51] H. Lenz, C. Wagner, and R. Sollacher, “Multi-anticipative car-following model,” *The European Physical Journal B*, vol. 7, no. 2, pp. 331–335, Jan. 1999. 12

[52] P. Liao, T.-Q. Tang, T. Wang, and J. Zhang, “A car-following model accounting for the driving habits,” *Physica A: Statistical Mechanics and its Applications*, vol. 525, pp. 108–118, Jul. 2019. 12

[53] M. Lárraga, J. d. Río, and L. Alvarez-lcaza, "Cellular automata for one-lane traffic flow modeling," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 1, pp. 63–74, Feb. 2005. 12

[54] "Monte Carlo method — Wikipedia, the free encyclopedia," Jan. 2022. 12

[55] S. Jeon and B. Hong, "Monte Carlo simulation-based traffic speed forecasting using historical big data," *Future Generation Computer Systems*, vol. 65, pp. 182–195, Dec. 2016. 12

[56] "CarSim — Wikipedia, the free encyclopedia," Nov. 2021. 12

[57] "PTV VISSIM — Wikipedia, the free encyclopedia," Dec. 2021. 12

[58] F. Xia, J. Wang, X. Kong, Z. Wang, J. Li, and C. Liu, "Exploring Human Mobility Patterns in Urban Scenarios: A Trajectory Data Perspective," *IEEE Communications Magazine*, vol. 56, no. 3, Mar. 2018. 13

[59] "Webcam — Wikipedia, the free encyclopedia," Jul. 2022. 13

[60] J. Han, M. Kamber, and J. Pei, "1 - Introduction," in *Data Mining (Third Edition)*, ser. The Morgan Kaufmann Series in Data Management Systems, J. Han, M. Kamber, and J. Pei, Eds. Boston: Morgan Kaufmann, Jan. 2012, pp. 1–38. 14

[61] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *, IEEE Symposium on Visual Languages, 1996. Proceedings*, Sep. 1996, pp. 336–343. 14, 25, 85

[62] D. Chu, D. A. Sheets, Y. Zhao, Y. Wu, J. Yang, M. Zheng, and G. Chen, "Visualizing Hidden Themes of Taxi Movement with Semantic Transformation," in *2014 IEEE Pacific Visualization Symposium*, Mar. 2014, pp. 137–144. 18, 19, 23

[63] G. N. Oliveira, J. L. Sotomayor, R. P. Torchelsen, C. T. Silva, and J. L. Comba, "Visual analysis of bike-sharing systems," *Computers & Graphics*, vol. 60, pp. 119–129, Nov. 2016. 18

[64] F. Miranda, H. Doraiswamy, M. Lage, K. Zhao, B. Goncalves, L. Wilson, M. Hsieh, and C. T. Silva, "Urban Pulse: Capturing the Rhythm of Cities," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 791–800, Jan. 2017. 18

[65] R. Wang, C.-Y. Chow, Y. Lyu, V. C. S. Lee, S. Kwong, Y. Li, and J. Zeng, "TaxiRec: Recommending Road Clusters to Taxi Drivers Using Ranking-Based Extreme Learning Machines," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 585–598, Mar. 2018. 18

[66] M. Lu, C. Lai, T. Ye, J. Liang, and X. Yuan, "Visual analysis of route choice behaviour based on GPS trajectories," in *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct. 2015, pp. 203–204. 18

[67] Q. Shen, W. Zeng, Y. Ye, S. M. Arisona, S. Schubiger, R. Burkhard, and H. Qu, "StreetVizor: Visual Exploration of Human-Scale Urban Forms Based on Street Views," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 1004–1013, Jan. 2018. 18

[68] N. Ferreira, M. Lage, H. Doraiswamy, H. Vo, L. Wilson, H. Werner, M. Park, and C. Silva, "Urbane: A 3D framework to support data driven decision making in urban development," in *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct. 2015, pp. 97–104. 18

[69] D. Liu, D. Weng, Y. Li, J. Bao, Y. Zheng, H. Qu, and Y. Wu, "SmartAdP: Visual Analytics of Large-scale Taxi Trajectories for Selecting Billboard Locations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 1–10, Jan. 2017. 18, 19, 23

[70] "Individual mobility — Wikipedia, the free encyclopedia," Oct. 2021. 19, 21

[71] W. Chen, F. Guo, and F.-Y. Wang, "A Survey of Traffic Data Visualization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2970–2984, Dec. 2015. 19, 24, 76

[72] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. Van De Wetering, "Visual Traffic Jam Analysis Based on Trajectory Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2159–2168, Dec. 2013. 19, 20

[73] B. N. Hilton, T. A. Horan, R. Burkhard, and B. Schooley, "SafeRoadMaps: Communication of Location and Density of Traffic Fatalities through Spatial Visualization and Heat Map Analysis," *Information Visualization*, vol. 10, no. 1, pp. 82–96, Jan. 2011. 19, 20

[74] Y. Gu, M.-J. Kraak, and Y. Engelhardt, "Revisiting flow maps: a classification and a 3D alternative to visual clutter," in *Proceedings of the International Cartographic Association (ICA)*, Washington, USA, 2017, p. 51. 19

[75] G. Andrienko and N. Andrienko, "Spatio-temporal aggregation for visual analysis of movements," in *2008 IEEE Symposium on Visual Analytics Science and Technology*, Oct. 2008, pp. 51–58. 19, 20

[76] R. Scheepens, C. Hurter, H. Van De Wetering, and J. J. Van Wijk, "Visualization, Selection, and Analysis of Traffic Flows," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 379–388, Jan. 2016. 19, 20, 22, 24, 28, 53

[77] O. Lock, T. Bednarz, and C. Pettit, "The visual analytics of big, open public transport data – a framework and pipeline for monitoring system performance in Greater Sydney," *Big Earth Data*, vol. 5, no. 1, pp. 134–159, Jan. 2021. 19, 20

[78] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang, "Hierarchical and Networked Vehicle Surveillance in ITS: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 557–580, Apr. 2015. 19, 21

[79] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37–47, Mar. 2002. 19, 21

[80] T. Gao, Z.-g. Liu, W.-c. Gao, and J. Zhang, "A Robust Technique for Background Subtraction in Traffic Video," in *Advances in Neuro-Information Processing*, M. Köppen, N. Kasabov, and G. Coghill, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5507, pp. 736–744. 19, 21

[81] S. Messelodi, C. M. Modena, and M. Zanin, "A computer vision system for the detection and classification of vehicles at urban road intersections," *Pattern Analysis and Applications*, vol. 8, no. 1-2, pp. 17–31, Sep. 2005. 19, 21

[82] B. Morris and M. Trivedi, "Learning, Modeling, and Classification of Vehicle Track Patterns from Live Video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 425–437, Sep. 2008. 19, 21

[83] Y. Wang, K. Qin, Y. Chen, and P. Zhao, "Detecting Anomalous Trajectories and Behavior Patterns Using Hierarchical Clustering from Taxi GPS Data," *ISPRS International Journal of Geo-Information*, vol. 7, no. 1, p. 25, Jan. 2018. 19, 21

[84] W. Kuang, S. An, and H. Jiang, "Detecting Traffic Anomalies in Urban Areas Using Taxi GPS Data," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–13, 2015. 19, 21

[85] Z. Wang, T. Ye, M. Lu, X. Yuan, H. Qu, J. Yuan, and Q. Wu, "Visual Exploration of Sparse Traffic Trajectory Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1813–1822, Dec. 2014. 19

[86] W. Wu, J. Xu, H. Zeng, Y. Zheng, H. Qu, B. Ni, M. Yuan, and L. M. Ni, "TelCoVis: Visual Exploration of Co-occurrence in Urban Human Mobility Based on Telco Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 935–944, Jan. 2016. 19, 22

[87] N. Andrienko, G. Andrienko, L. Barrett, M. Dostie, and P. Henzi, "Space Transformation for Understanding Group Movement," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2169–2178, Dec. 2013. 19, 22, 24, 29

[88] I. Kalamaras, A. Zamichos, A. Salamanis, A. Drosou, D. D. Kehagias, G. Margaritis, S. Papadopoulos, and D. Tzovaras, "An Interactive Visual Analytics Platform for Smart Intelligent Transportation Systems Management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 487–496, Feb. 2018. 19, 22

[89] T. von Landesberger, F. Brodkorb, P. Roskosch, N. Andrienko, G. Andrienko, and A. Kerren, “MobilityGraphs: Visual Analysis of Mass Mobility Dynamics via Spatio-Temporal Graphs and Clustering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 11–20, Jan. 2016. 19, 22

[90] X. Yao, D. Zhu, Y. Gao, L. Wu, P. Zhang, and Y. Liu, “A Stepwise Spatio-Temporal Flow Clustering Method for Discovering Mobility Trends,” *IEEE Access*, vol. 6, pp. 44 666–44 675, 2018. 19, 22

[91] X. Luo, Y. Yuan, Z. Li, M. Zhu, Y. Xu, L. Chang, X. Sun, and Z. Ding, “FBVA: A Flow-Based Visual Analytics Approach for Citywide Crowd Mobility,” *IEEE Transactions on Computational Social Systems*, vol. 6, no. 2, pp. 277–288, Apr. 2019. 19, 22

[92] H. Otten, L. Hildebrand, T. Nagel, M. Dork, and B. Muller, “Shifted Maps: Revealing spatio-temporal topologies in movement data,” in *2018 IEEE VIS Arts Program (VISAP)*. Berlin, Germany: IEEE, Oct. 2018, pp. 1–10. 19, 22

[93] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*. Routledge, Feb. 2018. 19, 22

[94] R. Scheepens, N. Willems, H. van de Wetering, and J. J. van Wijk, “Interactive visualization of multivariate trajectory data with density maps,” in *2011 IEEE Pacific Visualization Symposium*. Hong Kong, China: IEEE, Mar. 2011, pp. 147–154. 19, 22

[95] V. Cristie, M. Berger, P. Buš, A. Kumar, and B. Klein, “CityHeat: visualizing cellular automata-based traffic heat in Unity3D,” in *SIGGRAPH Asia 2015 Visualization in High Performance Computing*, Nov. 2015, pp. 1–4. 19, 22

[96] C. Kang, S. Gao, X. Lin, Y. Xiao, Y. Yuan, Y. Liu, and X. Ma, “Analyzing and geo-visualizing individual human mobility patterns using mobile call records,” in *2010 18th International Conference on Geoinformatics*. Beijing, China: IEEE, Jun. 2010, pp. 1–7. 19, 24, 31

[97] W. Zeng, C.-W. Fu, S. Muller Arisona, S. Schubiger, R. Burkhard, and K.-L. Ma, “Visualizing the Relationship Between Human Mobility and Points of Interest,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2271–2284, Aug. 2017. 19, 23

[98] S. Al-Dohuki, Y. Wu, F. Kamw, J. Yang, X. Li, Y. Zhao, X. Ye, W. Chen, C. Ma, and F. Wang, “SemanticTraj: A New Approach to Interacting with Massive Taxi Trajectories,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 11–20, Jan. 2017. 19, 23

[99] X. Zhao, Y. Zhang, Y. Hu, S. Wang, Y. Li, S. Qian, and B. Yin, “Interactive Visual Exploration of Human Mobility Correlation Based on Smart Card Data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4825–4837, Aug. 2021. 19, 23

[100] G. Sagl, M. Loidl, and E. Beinat, “A Visual Analytics Approach for Extracting Spatio-Temporal Urban Mobility Information from Mobile Network Traffic,” *ISPRS International Journal of Geo-Information*, vol. 1, no. 3, pp. 256–271, Nov. 2012. 19, 23

[101] C. Neustaedter, S. Greenberg, and M. Boyle, “Blur filtration fails to preserve privacy for home-based video conferencing,” *ACM Transactions on Computer-Human Interaction*, vol. 13, no. 1, pp. 1–36, Mar. 2006. 21

[102] A. Clarinval and B. Dumas, “Intra-City Traffic Data Visualization: A Systematic Literature Review,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–18, 2021. 24

[103] D. Xu and Y. Wei, “Study on Visual Techniques of Potential Pattern Discovery for Time Series Data,” *MATEC Web of Conferences*, vol. 232, p. 02049, 2018. 24, 26

[104] Z. Wang and X. Yuan, “Urban trajectory timeline visualization,” in *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*. Bangkok, Thailand: IEEE, Jan. 2014, pp. 13–18. 24, 25

[105] W. Zeng, C. Fu, S. M. Arisona, A. Erath, and H. Qu, “Visualizing Mobility of Public Transportation System,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1833–1842, Dec. 2014. 24, 25

[106] Y. Tanahashi and K.-L. Ma, “Design Considerations for Optimizing Storyline Visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2679–2688, Dec. 2012. 24, 26

[107] H. Doraiswamy, N. Ferreira, T. Damoulas, J. Freire, and C. T. Silva, “Using Topological Analysis to Support Event-Guided Exploration in Urban Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2634–2643, Dec. 2014. 24, 26, 31

[108] C. Palomo, Z. Guo, C. T. Silva, and J. Freire, “Visually Exploring Transportation Schedules,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 170–179, Jan. 2016. 24, 26

[109] “Calendar — Wikipedia, the free encyclopedia,” Jan. 2022. 24, 26

[110] J. Pu, S. Liu, H. Qu, and L. Ni, “Visual Fingerprinting: A New Visual Mining Approach for Large-Scale Spatio-temporal Evolving Data,” in *Advanced Data Mining and Applications*, ser. Lecture Notes in Computer Science, S. Zhou, S. Zhang, and G. Karypis, Eds. Berlin, Heidelberg: Springer, 2012, pp. 502–515. 24, 27

[111] J. Pu, S. Liu, Y. Ding, H. Qu, and L. Ni, “T-Watcher: A New Visual Analytic System for Effective Traffic Surveillance,” in *2013 IEEE 14th International Conference on Mobile Data Management*, vol. 1, Jun. 2013, pp. 127–136. 24, 27

[112] S. Liu, J. Pu, Q. Luo, H. Qu, L. M. Ni, and R. Krishnan, "VAIT: A Visual Analytics System for Metropolitan Transportation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1586–1596, Dec. 2013. 24, 28

[113] J. Xu, Y. Tao, Y. Yan, and H. Lin, "VAUT: a visual analytics system of spatiotemporal urban topics in reviews," *Journal of Visualization*, vol. 21, no. 3, pp. 471–484, Jun. 2018. 24, 29

[114] A. Slingsby, J. Dykes, and J. Wood, "Using treemaps for variable selection in spatio-temporal visualisation," *Information Visualization*, vol. 7, no. 3-4, pp. 210–224, Sep. 2008. 24, 30

[115] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale, "A Descriptive Framework for Temporal Data Visualizations Based on Generalized Space-Time Cubes," *Computer Graphics Forum*, vol. 36, no. 6, pp. 36–61, 2017. 24, 31

[116] I. Kveladze, M.-J. Kraak, and C. Elzakker, "A Methodological Framework for Researching the Usability of the Space-Time Cube," *The Cartographic Journal*, vol. 50, pp. 201–210, Aug. 2013. 24

[117] Y. Tang, F. Sheng, H. Zhang, C. Shi, X. Qin, and J. Fan, "Visual analysis of traffic data based on topic modeling (ChinaVis 2017)," *Journal of Visualization*, vol. 21, Mar. 2018. 24, 31

[118] J. Dykes, J. Wood, and A. Slingsby, "Rethinking Map Legends with Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 890–899, Nov. 2010. 24, 32

[119] E. R. Tufte, "The Visual Display of Quantitative Information," *The Journal for Healthcare Quality (JHQ)*, vol. 7, no. 3, p. 15, Jul. 1985. 25, 51, 53, 57, 69

[120] J. C. Roberts, P. W. S. Butcher, and P. D. Ritsos, "One View Is Not Enough: Review of and Encouragement for Multiple and Alternative Representations in 3D and Immersive Visualisation," *Computers*, vol. 11, no. 2, p. 20, Feb. 2022. 32, 33, 89

[121] J. C. Roberts, "State of the Art: Coordinated & Multiple Views in Exploratory Visualization," in *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*. Zurich, Switzerland: IEEE, Jul. 2007, pp. 61–71. 32, 33, 84

[122] J. C. Roberts, H. Al-maneea, P. W. S. Butcher, R. Lew, G. Rees, N. Sharma, and A. Frankenberg-Garcia, "Multiple Views: different meanings and collocated words," *Computer Graphics Forum*, vol. 38, no. 3, pp. 79–93, 2019. 32

[123] X. Chen, W. Zeng, Y. Lin, H. M. AI-maneea, J. Roberts, and R. Chang, "Composition and Configuration Patterns in Multiple-View Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1514–1524, Feb. 2021. 33, 89

[124] L. Shao, Z. Chu, X. Chen, Y. Lin, and W. Zeng, "Modeling layout design for multiple-view visualization via Bayesian inference," *Journal of Visualization*, vol. 24, no. 6, pp. 1237–1252, Dec. 2021. 33

[125] R. Langner, U. Kister, and R. Dachselt, "Multiple Coordinated Views at Large Displays for Multiple Users: Empirical Findings on User Behavior, Movements, and Distances," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 608–618, Jan. 2019. 33

[126] H. M. Al-maneea and J. C. Roberts, "Study of Multiple View Layout Strategies in Visualisation," in *IEEE Conference on Visualization: InfoVis*, 2018, p. 3. 33, 34

[127] S. LYi, J. Jo, and J. Seo, "Comparative Layouts Revisited: Design Space, Guidelines, and Future Directions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1525–1535, Feb. 2021. 33

[128] J. C. Roberts, C. Headleand, and P. D. Ritsos, "Sketching Designs Using the Five Design-Sheet Methodology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 419–428, Jan. 2016. 33

[129] I. Cruz and Y. F. Huang, "A Layered Architecture for the Exploration of Heterogeneous Information Using Coordinated Views," in *2004 IEEE Symposium on Visual Languages - Human Centric Computing*, Sep. 2004, pp. 11–18. 33

[130] A. Wu, Y. Wang, M. Zhou, X. He, H. Zhang, H. Qu, and D. Zhang, "MultiVision: Designing Analytical Dashboards with Deep Learning Based Recommendation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 162–172, Jan. 2022. 33

[131] P. Xu, H. Fu, T. Igarashi, and C.-L. Tai, "Global beautification of layouts with interactive ambiguity resolution," in *Proceedings of the 27th annual ACM symposium on User interface software and technology*. Honolulu Hawaii USA: ACM, Oct. 2014, pp. 243–252. 33

[132] N. Boukhelifa, P. Rodgers, and J. Roberts, "A coordination model for explor atory multi-view visualization," in *International Conference on Coordinated and Multiple Views in Exploratory Visualization*, ser. International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2003). NA, France: IEEE Computer Society, 2003, p. np. 34

[133] C. Eichner, H. Schumann, and C. Tominski, "Multi-display Visual Analysis: Model, Interface, and Layout Computation," *arXiv:1912.08558 [cs]*, Dec. 2019. 34

[134] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965. 36, 71

[135] W. Pedrycz and F. Gomide, *Fuzzy Systems Engineering: Toward Human-Centric Computing*. John Wiley & Sons, Oct. 2007. 36

[136] M. Bostock, V. Ogievetsky, and J. Heer, “D³ Data-Driven Documents,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011. 37, 41, 51, 86

[137] C. Seifert and E. Lex, “A Novel Visualization Approach for Data-Mining-Related Classification,” in *2009 13th International Conference Information Visualisation*, Jul. 2009, pp. 490–495. 37

[138] P. Rheingans and M. DesJardins, “Visualizing high-dimensional predictive model quality,” in *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*, Oct. 2000, pp. 493–496. 37

[139] B. Alsallakh, A. Hanbury, H. Hauser, S. Miksch, and A. Rauber, “Visual Methods for Analyzing Probabilistic Classification Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1703–1712, Dec. 2014. 37

[140] N. Cao, Y.-R. Lin, and D. Gotz, “UnTangle Map: Visual Analysis of Probabilistic Multi-Label Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 2, pp. 1149–1163, Feb. 2016. 37

[141] Y.-R. Lin, N. Cao, D. Gotz, and L. Lu, “UnTangle: Visual Mining for Data with Uncertain Multi-labels via Triangle Map,” in *2014 IEEE International Conference on Data Mining*, Dec. 2014, pp. 340–349. 37

[142] Y. Park and J. Park, “Disk Diagram: An Interactive Visualization Technique of Fuzzy Set Operations for the Analysis of Fuzzy Data,” *Information Visualization*, vol. 9, no. 3, pp. 220–232, Sep. 2010. 37, 71

[143] L. Zhu, W. Xia, J. Liu, and A. Song, “Visualizing fuzzy sets using opacity-varying freeform diagrams,” *Information Visualization*, vol. 17, no. 2, pp. 146–160, 2018. 37

[144] F. Zhou, B. Bai, Y. Wu, M. Chen, Z. Zhong, R. Zhu, Y. Chen, and Y. Zhao, “FuzzyRadar: visualization for understanding fuzzy clusters,” *Journal of Visualization*, vol. 22, no. 5, pp. 913–926, Oct. 2019. 37

[145] F. Zhou, M. Chen, Z. Wang, F. Luo, X. Luo, W. Huang, Y. Chen, and Y. Zhao, “A radviz-based visualization for understanding fuzzy clustering results,” in *Proceedings of the 10th International Symposium on Visual Information Communication and Interaction*, ser. VINCI ’17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 9–15. 37

[146] J. Sharko and G. Grinstein, “Visualizing Fuzzy Clusters Using RadViz,” in *2009 13th International Conference Information Visualisation*, Jul. 2009, pp. 307–316. 38

[147] A. R. Buck and J. M. Keller, “Visualizing uncertainty with fuzzy rose diagrams,” in *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, Dec. 2014, pp. 30–36. 38

[148] L. Hall and M. Berthold, "Fuzzy parallel coordinates," in *PeachFuzz 2000. 19th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.00TH8500)*, Jul. 2000, pp. 74–78. 38

[149] B. Pham and R. Brown, "Visualisation of fuzzy systems: requirements, techniques and framework," *Future Generation Computer Systems*, vol. 21, no. 7, pp. 1199–1212, 2005. 38

[150] M. Berthold and L. Hall, "Visualizing fuzzy points in parallel coordinates," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 3, pp. 369–374, Jun. 2003. 38

[151] J. Caha and A. Vondrakova, "Fuzzy Surface Visualization using HSL Colour Model," *Scientific Visualization*, vol. 9, pp. 26–42, 2017. 38

[152] K. Brodlie, R. Allendes Osorio, and A. Lopes, "A Review of Uncertainty in Data Visualization," in *Expanding the Frontiers of Visual Analytics and Visualization*, J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. C. Wong, Eds. London: Springer, 2012, pp. 81–109. 38

[153] M. Skeels, B. Lee, G. Smith, and G. G. Robertson, "Revealing Uncertainty for Information Visualization," *Information Visualization*, vol. 9, no. 1, pp. 70–81, Jan. 2010. 38

[154] X. Dong and C. C. Hayes, "Uncertainty Visualizations: Helping Decision Makers Become More Aware of Uncertainty and Its Implications," *Journal of Cognitive Engineering and Decision Making*, vol. 6, no. 1, pp. 30–56, Mar. 2012. 38

[155] E. Massad, N. R. S. Ortega, L. C. d. Barros, and C. J. Struchiner, *Fuzzy Logic in Action: Applications in Epidemiology and Beyond.* Springer Science & Business Media, Feb. 2009. 39

[156] J. Zhao, C. Collins, F. Chevalier, and R. Balakrishnan, "Interactive Exploration of Implicit and Explicit Relations in Faceted Datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2080–2089, Dec. 2013. 50, 52

[157] R. Vuillemot and J. Boy, "Structuring Visualization Mock-Ups at the Graphical Level by Dividing the Display Space," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 424–434, Jan. 2018. 51, 52, 54, 55, 56, 63, 64, 65

[158] M. A. Munizaga and C. Palma, "Estimation of a disaggregate multimodal public transport Origin–Destination matrix from passive smartcard data from Santiago, Chile," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 9–18, Oct. 2012. 51

[159] D. Li, Y. Lin, X. Zhao, H. Song, and N. Zou, "Estimating a Transit Passenger Trip Origin-Destination Matrix Using Automatic Fare Collection System," in *Database Systems for Adanced Applications*, ser. Lecture Notes in Computer Science, J. Xu, G. Yu, S. Zhou, and R. Unland, Eds. Springer Berlin Heidelberg, 2011, pp. 502–513. 51

[160] O. Ersoy, C. Hurter, F. Paulovich, G. Cantareiro, and A. Telea, “Skeleton-Based Edge Bundling for Graph Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2364–2373, Dec. 2011. 51

[161] M. S. Iqbal, C. F. Choudhury, P. Wang, and M. C. González, “Development of origin–destination matrices using mobile phone call data,” *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 63–74, Mar. 2014. 51

[162] T. Munzner, *Visualization Analysis and Design*. CRC Press, Dec. 2014. 51

[163] R. Beecham, C. Rooney, S. Meier, J. Dykes, A. Slingsby, C. Turkay, J. Wood, and B. L. W. Wong, “Faceted Views of Varying Emphasis (FaVVEs): a framework for visualising multi-perspective small multiples,” *Computer Graphics Forum*, vol. 35, no. 3, pp. 241–249, Jun. 2016. 51

[164] S. v. d. Elzen and J. J. v. Wijk, “Small Multiples, Large Singles: A New Approach for Visual Data Exploration,” *Computer Graphics Forum*, vol. 32, no. 3pt2, pp. 191–200, 2013. 52, 64

[165] M. Ghoniem, J. Fekete, and P. Castagliola, “A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations,” in *IEEE Symposium on Information Visualization*, Oct. 2004, pp. 17–24. 52

[166] M. Wattenberg, “Visual Exploration of Multivariate Graphs,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’06. New York, NY, USA: ACM, 2006, pp. 811–819. 52

[167] W. S. Cleveland and R. McGill, “Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods,” *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 531–554, 1984. 53

[168] D. Park, S. M. Drucker, R. Fernandez, and N. Elmqvist, “ATOM: A Grammar for Unit Visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2017. 53

[169] A. Satyanarayan, K. Wongsuphasawat, and J. Heer, “Declarative Interaction Design for Data Visualization,” in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’14. New York, NY, USA: ACM, 2014, pp. 669–678. 53

[170] L. Wilkinson, *The Grammar of Graphics (Statistics and Computing)*. Berlin, Heidelberg: Springer-Verlag, 2005. 53

[171] C. Stolte, D. Tang, and P. Hanrahan, “Polaris: A System for Query, Analysis, and Visualization of Multidimensional Databases,” *Commun. ACM*, vol. 51, no. 11, pp. 75–84, Nov. 2008. 53

[172] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 649–658, Jan. 2016. 53, 57

[173] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A Grammar of Interactive Graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 341–350, Jan. 2017. 53, 56, 63, 102

[174] D. Holten and J. J. Van Wijk, "Force-Directed Edge Bundling for Graph Visualization," *Computer Graphics Forum*, vol. 28, no. 3, pp. 983–990, 2009. 53

[175] W. R. Tobler, "Experiments in migration mapping by computer," *The American Cartographer*, pp. 155–163, 1987. 53

[176] X. Zhu, D. Guo, C. Koylu, and C. Chen, "Density-based multi-scale flow mapping and generalization," *Computers, Environment and Urban Systems*, vol. 77, p. 101359, Sep. 2019. 53

[177] A. Slingsby, J. Dykes, and J. Wood, "Configuring Hierarchical Layouts to Address Research Questions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 977–984, Nov. 2009. 54, 56, 57

[178] K. Furmanova, S. Gratzl, H. Stitz, T. Zichner, M. Jaresova, M. Ennemoser, A. Lex, and M. Streit, "Taggle: Scalable Visualization of Tabular Data through Aggregation," *arXiv:1712.05944 [cs]*, Dec. 2017. 55

[179] R. Amar, J. Eagan, and J. Stasko, "Low-Level Components of Analytic Activity in Information Visualization," in *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, ser. INFOVIS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 15–. 55

[180] C. Stolte, D. Tang, and P. Hanrahan, "Multiscale visualization using data cubes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 2, pp. 176–187, Apr. 2003. 55

[181] J. Fuchs, P. Isenberg, A. Bezerianos, and D. Keim, "A Systematic Review of Experimental Studies on Data Glyphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1863–1879, Jul. 2017. 56

[182] N. H. Riche, B. Lee, and C. Plaisant, "Understanding Interactive Legends: a Comparative Evaluation with Standard Widgets," *Computer Graphics Forum*, vol. 29, no. 3, pp. 1193–1202, 2010. 59

[183] Y. Liu and J. Heer, "Somewhere Over the Rainbow: An Empirical Assessment of Quantitative Colormaps," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. Montreal QC, Canada: ACM Press, 2018, pp. 1–12. 59

[184] X. Yao, L. Wu, D. Zhu, Y. Gao, and Y. Liu, “Visualizing spatial interaction characteristics with direction-based pattern maps,” *Journal of Visualization*, Jan. 2019. 63

[185] C. Perin, T. Wun, R. Pusch, and S. Carpendale, “Assessing the Graphical Perception of Time and Speed on 2d+Time Trajectories,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 698–708, Jan. 2018. 63

[186] B. Bach, C. Perin, Q. Ren, and P. Dragicevic, “Ways of Visualizing Data on Curves,” Apr. 2018, pp. 1–14. 63

[187] J. Matejka, F. Anderson, and G. Fitzmaurice, “Dynamic Opacity Optimization for Scatter Plots,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI ’15. New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 2707–2710. 63, 69

[188] H.-w. Chang, Y.-c. Tai, H.-w. Chen, and J. Y.-j. Hsu, “iTaxi: Context-Aware Taxi Demand Hotspots Prediction Using Ontology and Data Mining Approaches,” p. 8. 63

[189] W. Cancino, N. Boukhelifa, and E. Lutton, “EvoGraphDice: Interactive evolution for visual analytics,” in *2012 IEEE Congress on Evolutionary Computation*, Jun. 2012, pp. 1–8. 64

[190] S. Zhao, M. McGuffin, and M. Chignell, “Elastic hierarchies: combining treemaps and node-link diagrams,” in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, Oct. 2005, pp. 57–64. 64

[191] A. Lex, H. Schulz, M. Streit, C. Partl, and D. Schmalstieg, “VisBricks: Multiform Visualization of Large, Inhomogeneous Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2291–2300, Dec. 2011. 64

[192] D. Liu, P. Xu, and L. Ren, “TPFlow: Progressive Partition and Multidimensional Pattern Extraction for Large-Scale Spatio-Temporal Data Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1–11, Jan. 2019. 64

[193] S. v. d. Elzen and J. J. v. Wijk, “BaobabView: Interactive construction and analysis of decision trees,” in *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct. 2011, pp. 151–160. 64

[194] Y. Yang, T. Dwyer, B. Jenny, K. Marriott, M. Cordeil, and H. Chen, “Origin-destination flow maps in immersive environments,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 693–703, 2018. 64

[195] D. Guilmaine, C. Viau, and M. J. McGuffin, “Hierarchically Animated Transitions in Visualizations of Tree Structures,” in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI ’12. New York, NY, USA: ACM, 2012, pp. 514–521. 64

[196] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, *Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges*. The Eurographics Association, 2014. 68, 70, 71, 74

[197] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, *Visualization of Time-Oriented Data*, ser. Human-Computer Interaction Series. London: Springer, 2011. 69

[198] J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg, “Evaluation of alternative glyph designs for time series data in a small multiple setting,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’13*. Paris, France: ACM Press, 2013, p. 3237. 69

[199] J. Zhao, F. Chevalier, and R. Balakrishnan, “KronoMiner: using multi-foci navigation for the visual exploration of time-series data,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’11. New York, NY, USA: Association for Computing Machinery, May 2011, pp. 1737–1746. 69

[200] W. Javed and N. Elmqvist, “Stack zooming for multi-focus interaction in time-series data visualization,” in *2010 IEEE Pacific Visualization Symposium (PacificVis)*, Mar. 2010, pp. 33–40. 69

[201] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan, “Exploratory Analysis of Time-Series with ChronoLenses,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2422–2431, Dec. 2011. 69

[202] H. Reijner and P. Software, *The Development of the Horizon Graph*, 2008. 69

[203] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu, “StoryFlow: Tracking the Evolution of Stories,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2436–2445, Dec. 2013. 69

[204] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic, “Time Curves: Folding Time to Visualize Patterns of Temporal Evolution in Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 559–568, Jan. 2016. 69

[205] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, “LineUp: Visual Analysis of Multi-Attribute Rankings,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2277–2286, Dec. 2013. 70

[206] C. Shi, W. Cui, S. Liu, P. Xu, W. Chen, and H. Qu, “RankExplorer: Visualization of Ranking Changes in Large Time Series Data,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, pp. 2669–2678, Dec. 2012. 70, 73

[207] H. Lei, J. Xia, F. Guo, Y. Zou, W. Chen, and Z. Liu, “Visual exploration of latent ranking evolutions in time series,” *Journal of Visualization*, vol. 19, no. 4, pp. 783–795, Nov. 2016. 70

[208] S. Silva and T. Catarci, “Visualization of linear time-oriented data: a survey,” in *Proceedings of the First International Conference on Web Information Systems Engineering*, vol. 1, Jun. 2000, pp. 310–319 vol.1. 70

[209] R. Vuillemot and C. Perin, “Investigating the Direct Manipulation of Ranking Tables for Time Navigation,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI ’15. New York, NY, USA: ACM, 2015, pp. 2703–2706. 70

[210] C. Bryan, K.-L. Ma, and J. Woodring, “Temporal Summary Images: An Approach to Narrative Visualization via Interactive Annotation Generation and Placement,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 511–520, Jan. 2017. 70

[211] A. Inselberg, “The plane with parallel coordinates,” *The Visual Computer*, vol. 1, no. 2, pp. 69–91, Aug. 1985. 70

[212] R. Kosara, F. Bendix, and H. Hauser, “Parallel Sets: interactive exploration and visual analysis of categorical data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 558–568, Jul. 2006. 70

[213] T. von Landesberger, S. Bremm, N. Andrienko, G. Andrienko, and M. Tekušová, “Visual analytics methods for categoric spatio-temporal data,” in *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct. 2012, pp. 183–192. 70, 73, 76

[214] B. Alsallakh and L. Ren, “PowerSet: A Comprehensive Visualization of Set Intersections,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 361–370, Jan. 2017. 70

[215] B. Alsallakh, W. Aigner, S. Miksch, and H. Hauser, “Radial Sets: Interactive Visual Analysis of Large Overlapping Sets,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2496–2505, Dec. 2013. 70

[216] M. A. Yalcin, N. Elmqvist, and B. B. Bederson, “AggreSet: Rich and Scalable Set Exploration using Visualizations of Element Aggregations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 688–697, Jan. 2016. 70, 76, 79

[217] P. H. Nguyen, K. Xu, R. Walker, and B. W. Wong, “TimeSets: Timeline visualization with set relations,” *Information Visualization*, vol. 15, no. 3, pp. 253–269, Jul. 2016. 70

[218] S. Agarwal and F. Beck, “Set Streams: Visual Exploration of Dynamic Overlapping Sets,” *Computer Graphics Forum*, vol. 39, no. 3, pp. 383–391, 2020. 70

[219] S. Agarwal, G. Tkachev, M. Wermelinger, and F. Beck, “Visualizing Sets and Changes in Membership Using Layered Set Intersection Graphs,” in *VMV: Vision, Modeling, and Visualization*, vol. VMV2020. The Eurographics Association, 2020, pp. 69–78. 70

[220] W. Freiler, K. Matković, and H. Hauser, “Interactive Visual Analysis of Set-Typed Data,” *IEEE transactions on visualization and computer graphics*, vol. 14, pp. 1340–7, Nov. 2008. 70

[221] L. Liu and R. Vuillemot, “Categorizing Quantities using an Interactive Fuzzy Membership Function,” in *The 12th International Conference on Information Visualisation Theory and Applications*, On-line, France, Feb. 2021. 71

[222] H. Wang and M. Song, “Ckmeans.1d.dp: Optimal k-means Clustering in One Dimension by Dynamic Programming,” *The R journal*, vol. 3, no. 2, pp. 29–33, Dec. 2011. 75

[223] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. 80, 98

[224] C. Nowakowski, “Human Factors in Traffic Management Centers: A Literature Review,” University of Michigan, Tech. Rep., 1999. 82

[225] A. Prouzeau, “Collaboration around wall displays in command and control contexts - Chapter 3,” PhD Thesis, 2017. 82

[226] M. Zeilstra, M. Wilms, F. Blommers, and D. de Bruijn, “Development of future scenarios by prediction of mental workload in a traffic management control room,” *Advances in Human Aspects of Transportation: Part III*, vol. 9, p. 125, 2014. 83

[227] S. D. Starke, C. Baber, N. J. Cooke, and A. Howes, “Workflows and individual differences during visually guided routine tasks in a road traffic management control room,” *Applied Ergonomics*, vol. 61, pp. 79 – 89, 2017. 83

[228] D. J. Simons and C. F. Chabris, “Gorillas in Our Midst: Sustained Inattentional Blindness for Dynamic Events,” *Perception*, vol. 28, no. 9, pp. 1059–1074, 1999. 83

[229] C. Baber, N. S. Morar, and F. McCabe, “Ecological Interface Design, the Proximity Compatibility Principle, and Automation Reliability in Road Traffic Management,” *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 241–249, 2019. 83

[230] A. Prouzeau, A. Bezerianos, and O. Chapuis, “Towards Road Traffic Management with Forecasting on Wall Displays,” in *Proceedings of the 2016 International Conference on Interactive Surfaces and Spaces*, ser. ISS ’16. Niagara Falls, Canada: ACM, Nov. 2016, pp. 119–128. 83

[231] T. Schwarz, S. Butscher, J. Mueller, and H. Reiterer, “Content-Aware Navigation for Large Displays in Context of Traffic Control Rooms,” in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI ’12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 249–252. 83

[232] V. Galle, V. H. Manshadi, S. B. Boroujeni, C. Barnhart, and P. Jaillet, “The Stochastic Container Relocation Problem,” *Transportation Science*, vol. 52, no. 5, pp. 1035–1058, Oct. 2018. 94

[233] R. Hausmann and C. A. Hidalgo, *The atlas of economic complexity: Mapping paths to prosperity*. MIT Press, 2014. 95

[234] C. Perin, R. Vuillemot, C. D. Stolper, J. T. Stasko, J. Wood, and S. Carpendale, “State of the Art of Sports Data Visualization,” *Computer Graphics Forum*, vol. 37, no. 3, pp. 663–686, 2018. 96

[235] C. Perin, R. Vuillemot, and J.-D. Fekete, “SoccerStories: A Kick-off for Visual Soccer Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2506–2515, Dec. 2013. 96

[236] C. Stoiber, F. Grassinger, M. Pohl, H. Stitz, M. Streit, and W. Aigner, “Visualization Onboarding: Learning How to Read and Use Visualizations,” Open Science Framework, preprint, Aug. 2019. 102

[237] N. Vasudevan and L. Tratt, “Comparative study of DSL tools,” *Electronic Notes in Theoretical Computer Science*, vol. 264, Jul. 2011. 102

[238] C. Turkay, N. Pezzotti, C. Binnig, H. Strobelt, B. Hammer, D. A. Keim, J.-D. Fekete, T. Palpanas, Y. Wang, and F. Rusu, “Progressive Data Science: Potential and Challenges,” Sep. 2019. 102