UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: VISUALIZATION OF TWITTER CONVERSATIONS OVER TIME

Trabajo de integración curricular presentado como requisito para la obtención del título de Ingeniero en Tecnologías de la Información

Autor:

Llumiquinga Molina José Luis

Tutor:

Ph.D. - Cuenca Erick

Urcuquí, Diciembre 2022

Autoría

Yo, **José Luis Llumiquinga Molina**, con cédula de identidad **1725590895**, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el autor del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Diciembre del 2022.

José Luis Llumiquinga Molina
CI: 1725590895

Autorización de publicación

Yo, **José Luis Llumiquinga Molina** , con cédula de identidad **1725590895**, cedo a la Universidad de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Urcuquí, Diciembre del 2022.

Firmado electrónicamente por:
JOSE LUIS
LLUMIQUINGA MOLINA

José Luis Llumiquinga Molina
CI: 1725590895

Dedication

"This work is dedicated to my family, to my father, although he is no longer here, to my mother, my brother, and the rest of my relatives who have always been with me and supported me. "

Acknowledgments

My special thanks to my graduate project advisor for his patience, continuous support, and enthusiasm. Thanks to my family for all their support forever. Thanks to my friends in T2-2, who inspired me to study continuously and overcome the limits one set for oneself. Finally, thanks to Arianna, who has supported me and given me great unconditional advice, making me grow.

Abstract

Information visualization is a field of visual data analysis that works with abstract information. It also integrates interactions that facilitate users to assimilate the displayed information and understand it quickly. This work focuses on developing a visualization tool that follows all the fundamentals and theories involved in developing visualization for analyzing Twitter data. It analyzes an aspect often neglected in the literature despite its importance, specifically with the visualization tool Twitter conversations that have great importance and repercussions in social and political aspects are analyzed and explored. To obtain Twitter conversations, the official Twitter API is used. This basic information is treated to obtain a file with nodes and links that is very generic. Then the visualization tool is developed that receives the file of nodes and links and allows visualization of Twitter conversations. It also allows users to see how the conversations evolve. The development is validated, and finally, the visualization tool is tested on different datasets where interesting discoveries can lead to research related to bots on Twitter. Thanks to the visualization tool, the findings are adapted to a series of problems and datasets. Therefore the developed tool gives way to its use in other applications, allowing the discovery and analysis of network structures.

Keywords: InfoVis, Twitter, Conversation, Graph, Web Application.

Resumen

La visualización de la información es un campo de análisis visual de datos que trabaja con información abstracta. También integra interacciones que facilitan a los usuarios asimilar la información mostrada y comprenderla rápidamente. Este trabajo se centra en el desarrollo de una herramienta de visualización que sigue todos los fundamentos y teorías implicadas en el desarrollo de la visualización para el análisis de datos de Twitter. Se analiza un aspecto muchas veces descuidado en la literatura a pesar de su importancia, concretamente con la herramienta de visualización se analizan y exploran las conversaciones de Twitter que tienen gran importancia y repercusión en aspectos sociales y políticos. Para obtener las conversaciones de Twitter se utiliza la API oficial de Twitter. Esta información básica se trata para obtener un archivo con nodos y enlaces muy genérico. A continuación se desarrolla la herramienta de visualización que recibe el fichero de nodos y enlaces y permite visualizar las conversaciones de Twitter. También permite ver la evolución de las conversaciones. Se valida el desarrollo y, por último, se prueba la herramienta de visualización en diferentes conjuntos de datos en los que se producen descubrimientos interesantes que pueden dar lugar a investigaciones relacionadas con los bots en Twitter. Gracias a la herramienta de visualización, los hallazgos se adaptan a una serie de problemas y conjuntos de datos. Por lo tanto, la herramienta desarrollada da paso a su uso en otras aplicaciones, permitiendo el descubrimiento y análisis de estructuras de red.

Palabras Clave: InfoVis, Twitter, Conversaciones, Grafo, Aplicación web.

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem statement

The current era in which we live is characterized by the continuous use of a very diverse range of mobile technological devices (e.g., cell phones, GPS, smartwatch), and coupled with the internet help; they generate a constant and massive amount of information (Schlosser et al. [34]). This massive production of large amounts of data may contain information about the movement of objects (e.g., vehicles, people, social media posts) over time. An example of these data that move over space and time are Twitter[1] posts.

Twitter is one of the essential social networks of recent times. This social network has 650 million registered users and is the third most important social network after Instagram and Facebook. Twitter has 330 million monthly active users, 152 million daily active users, and approximately 500 million tweets posted daily (Antonakaki et al. [35]). According to Marketingcharts [36], it is estimated that the average daily time that people in the United States use this social network is more than 3 hours.

Twitter allows people to share whatever information type, like thoughts, news, links, and more. "The basic building block of Twitter is the Tweet" (Twitter Developer Platform [37]) which contains what the user wants to share. This tweet includes much helpful information to understand the information propagation along space and time; It contains information about when the tweet was posted and where it was posted. Based on the Tweet, various interactions could have originated as Retweets, replies, and quotes, containing the information necessary to track the data along space and time.

Twitter plays a vital role in the dissemination of news. Twitter has millions of interactions and is one of the most important sources for studying user interactions and information dissemination. On Twitter, there are accounts representing private and public institutions, agencies, public figures, political parties, celebrities, and other collectives of many natures. Given the great variety of Twitter accounts, it can be the subject of research in many fields, such as computer science and social or geopolitical. Knowing how information is moved and the interaction of users within Twitter can have many utilities. For example, In Celik [38] is addressed the problem of finding Socio-spatiotemporal important locations (SSTILs), which are places frequently visited by social media users in their

[1] `https://twitter.com/`

social media history. Celik [38] highlights the importance of finding these places for several application domains, such as recommender systems, urban planning, advertising applications, and more, so it is crucial to have a tool that facilitates these tasks and finding user patterns.

According to Antonakaki et al. [35], one of the major focuses of Twitter research is the structure of this social network through graphs. Several papers study how to construct and analyze graphs taking nodes such as hashtags, retweets, mentions, and replies (see Ferrara et al. [39], Bakshy et al. [40], Conover et al. [41], Bliss et al. [32] and Nishi et al. [42]). Some papers focus on studying the properties of the nodes. In others, the important thing is the structure of the network itself. Some find diffusion patterns in the graph structure. Papers focused on the graph structure can find user similarities or detect highly influential users. They can detect bots, and they can even measure the success of a Tweet.

What has not been addressed at present are conversations on Twitter. Twitter conversations are analogous to real conversations and debates. A Twitter conversation comprises a series of replies and quotes to a tweet. Therefore, you can perform various studies such as those mentioned above. Using graphs, you can analyze the dissemination of information, hate speech, fake news, and bots.

Most work that deals with graphs in the Twitter social network uses visualization tools. Visualizing the structure is one of the easiest ways to explore and discover what is going on in a network. Visualizations are responsible for delivering representations of datasets designed to help people perform their tasks more effectively. According to Spence [43], visualization is a tool to help humans form mental models of complex phenomena that would otherwise be difficult to understand. According to Munzner [8], visualizations are appropriate when human capabilities need to be augmented. Not when it is necessary to replace people with computational decision methods.

According to Fadloun [44], it is challenging to communicate the Information Visualization value, but this will be explored in the following. Information visualization is used in exploration tasks where much information is explored. Generally, a person using an information visualization tool will not have a specific objective and questions in mind. A person using an information visualization tool can examine, make discoveries or learn.

Information visualization is an exploratory process. The user may have questions and tasks about the dataset being viewed while performing an exploration. On the other hand, if a user has a precise question, he can convert this question to a database query. With the specific query, a quick and accurate answer can be provided. Visualization allows us to quickly see phenomena that were not known before and answer questions that were not known beforehand.

Visualizations can help to understand the data better. According to Fekete et al. [45], images augment human memory to provide a larger working set for reflection and analysis. Therefore images can be cognitive aids. Also, vision is the sense with the highest bandwidth: 100 MB/s (Mirel [46]). Therefore, vision is one of the best channels for delivering information to the brain. When data is displayed correctly, the human eye can perceive many properties effortlessly, regardless of the amount of data.

Finally, the visualizations worked on in many state-of-the-art papers do not take aspects of Information Visualization, and neither do they work with conversations. The presented work combines these two fields to help explore conversations on Twitter.

1.2 Objectives

1.2.1 General Objective

Propose a visualization tool based on fundamental, theoretical, and practical aspects of information visualization design to analyze multidimensional data from Twitter.

1.2.2 Specific Objectives

- Develop a visualization tool that satisfies the task of analyzing the trajectories and relationships of tweets within conversations using the most appropriate techniques for the available data and that additionally can be adapted not only to Twitter conversations.
- Present a visualization tool that pays attention to the temporal dimension based on the fundamental aspects for developing visualizations and complies with certain principles and design choices suitable for analyzing Twitter conversations.
- Develop a visualization tool that can be easily shared and integrates interactions through its development in HTML, CSS, and JavaScript for use from a web browser.

1.3 Contributions

This work proposes a visualization tool[2] that addresses a practically unexplored topic in the literature: the conversations within Twitter. Using Information Visualization techniques, a web tool addresses twitter conversations and uses different well-studied visual coding techniques. The tool allows one to explore and analyze Twitter conversations with an interactive approach and can be used in other fields of study with other datasets. Additionally, with a poster that is part of this work, it won first place in the Open House "Todos Nos Sumamos" of the School of Mathematics and Computer Science in the category of Degree II work. The Open House was held at Yachay Tech University on September 14, 2022, where students from Yachay Tech University presented their works in three categories: Degree I and II and research papers.

1.4 Document Organization

This work is made up of six main parts. These are summarized below.

- Chapter 1: Introduction. This chapter presents a brief introduction to the subject, then details the problem statement that addresses the development of this work and its contribution. The objectives and the organization of this document are presented.
- Chapter 2: Theoretical Framework. This section establishes the theoretical foundations on which this work is based and presents definitions to understand this work.

[2] `https://joseluis96001.github.io/`

- Chapter 3: State of the Art. This chapter explores a series of works that have had important repercussions within Information Visualization. Their techniques, their methods, and also the problems addressed are analyzed.
- Chapter 4: Methodology. This section presents the methodology used in this work. The process involved in developing the visualization tool and the reasons for the choices made to encode the information are presented.
- Chapter 5: Results and Discussion. This part presents the results obtained from this work and discusses various aspects involved in the development and use of the visualization tool.
- Chapter 6: Conclusions and Future Work. This section discusses the achievements of this work and provides an outlook of future research that builds on this work.

Chapter 2

Theoretical Framework

This section explores the theoretical foundations surrounding the field of Information Visualization. It begins with an introduction to Information Visualization and the three main fields of visualization: Information Visualization, Geographic Visualization, and Scientific Visualization. A general pipeline for visualization development is discussed. Then a brief differentiation between Information Visualization and Visual Analytics is explained. Then, is explained the process of creating visualizations. Essential aspects of visual data encoding are discussed, as well as interactions and problems. Also covered is how to validate visualizations with a nested model proposed by Munzner [8]. Then, we explore one of the fundamental aspects of visualization, the data. Later, Spatial-temporal Data is explored. Next, examples of datasets are given, and data in motion is analyzed. Finally, a summary is presented.

2.1 Information Visualization

According to Wijk [47], Visualization (VIS) can be ambiguous. This term can refer to a particular technology, a research discipline, a technique, or a visible result. In any of mentioned contexts, visualizations allow the audience to appropriate knowledge in a very efficient and functional manner. Visualizations enable people to understand a specific situation and use the visual capabilities of human beings to identify relationships and characteristics and find patterns in an agile way.

The beginning of the VIS domain as an independent field of research dates back to 1987. The term visualization in scientific computing was introduced at Washington's National Science Foundation conference in February 1987. In this conference, McCormick et al. [48] defined visualization as follows: "Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. It enriches the process of scientific discovery and fosters profound and unexpected insights".

The VIS domain can also be seen as a communication and teaching tool. For instance, DeFanti et al. [49] discussed visualization as a way of communication that goes beyond the limits of application and technology. At that time, interactions were practically unthinkable

due to the hardware used. Still, many advances in graphics hardware have been made, and visualizations have evolved a lot since then.

Moorhead et al. [50] talk about the value of visualization. They state that visualization, whether static or interactive, helps explain and understand data through the use of software. They also explain that designers can take advantage of the visual capabilities of human beings to improve and speed up the understanding of information.

Munzner [7] mentions that visualizations provide representations of datasets designed to help people perform tasks efficiently. It is explained that visualizations become essential when there is a need to augment human capabilities and that consideration should be given to how to create and interact with them. Additionally, it is suggested that VIS designers must consider the limitations of computers, humans, and displays.

Since the beginning of visualization in scientific computing, computing power has been significant for handling large amounts of data and generating graphics. With today's large amounts of data generation, having adequate hardware becomes even more critical. Thanks to advances in technological development, the creation of visualizations has become more accessible. Today's laptops have or exceed the power of the graphics workstations of decades ago and can handle large amounts of data.

The data handled contains information that can be useful for generating visualizations. Nowadays, it is common to see many interconnected devices, such as cell phones, smartwatches, and more. People are constantly interacting with these devices generating data (Paggi et al. [51]). The data can be presented in a variety of ways. Still, with it, interpretable information can be developed that can be used to analyze an event. According to Capurro and Hjørland [52], information has a digital nature and is multidisciplinary. Information can be seen as a thing, an object, for example, as a set of bits, data, or knowledge.

An example quite often quoted that gives a general idea of what InfoVis is the famous map made by Charles Minard in Fig. 2.1 on Napoleon's 1812 campaign into Russia. The map shows the losses of Napoleon's army during the Russian campaign by the width of the principal bands, and this is the prevailing idea that stands out more than the links between data and geography. It is seen how the size of Napoleon's army declines from his departure to his return. Fig. 2.1 does not consider the cartographic projections, and the scales were adapted according to the need. The gray line shows the advance of Napoleon's army toward Moscow, while the black line shows his return.

One might think that the two main flows in Fig. 2.1 took different routes, but in reality, the return of the armada follows approximately the same route as the outbound flow to Moscow. These two areas are drawn separately for clarity. The map also shows locations and generalized troop movements in general flows. Next to the retreat route is a temperature diagram, and names indicate battles and significant geographical features.

2.1.1 Visualization Types

The whole field of visualization can be subdivided into three subfields, and several papers deal with these topics. *Information Visualization* (see Lamping and Ramma [53], Andrews [54], Andrews et al. [55]), *Geographic Visualization* (see Müller [56], Joshi et al. [57], Rinner [58]) and *Scientific Visualization* (see Ki and Klasky [59], Wang and Han [60], Li and Chen [61], Coltekin et al. [62]). All these subfields are detailed below.

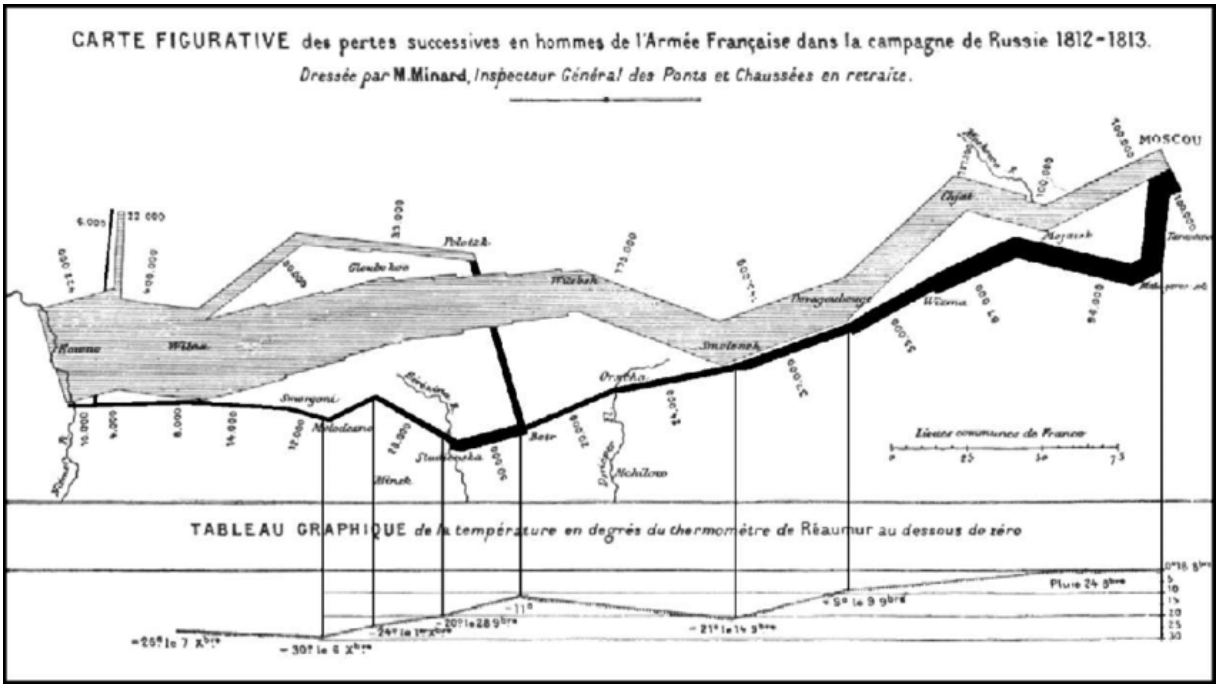


Figure 2.1: Napolean's March on Moscow. Taken from Cotrim and Campos [1].

Information Visualization (InfoVis)

Card et al. [63] gives the best-known definition of information visualization. They define InfoVis as "the use of computer supported, interactive, visual representations of abstract data to amplify cognition". Visual representations help understand and generate insight from data by aiding the visualization's interactions. This definition encompasses many terms that other authors also mention when defining InfoVis (see Lallé and Conati [64], Judelman [65], Yi et al. [66], Pousman et al. [67]). Visual representations present information to users using visual coding, colors, orientation, shapes, sizes, areas, volumes, and more. Abstract data can be digital data that does not have a real-world version, such as vectors, data migrating from one database to another, music being downloaded from the internet, and other examples. Cognition is acquiring information through mental functions such as repetition, memorization, reasoning, perception, attention, and many more (Buck [68]). Finally, as can be seen, InfoVis, according to Card et al. [63], links all these concepts to present a complete definition that has served as a starting point for definitions of InfoVis by other authors.

Sorapure [69] also presents a complete de finition of InfoVis: "InfoVis combines visual features (e.g., color, size, position), textual elements (e.g., titles, labels, instructions), and interactive options (e.g., search, zoom, filter) to produce different views of data to leverage human powers of perception in finding meaningful patterns and thus drawing information and insight out of data". The author explains that InfoVis has traditionally been developed to find answers to specific questions, i.e., they are clear, practical, and objective. It is said that these types of visualizations are utilitarian, where clear communication is essential

According to Sorapure [69], researchers in InfoVis of a "secondary" stream have developed techniques that focus on non-expert audiences. Therefore, InfoVis have a broader

spectrum of users working with data with more personal meanings, social, and intuitive knowledge than analytical data (see Pousman et al. [70]). Developers need to consider the motivations and practices of everyday consumers.

InfoVis usually deals with abstract data, i.e., with no physical referent. There may also be abstract information structures, such as hierarchies, networks, or multidimensional spaces. For example, Chang et al. [2] investigate airline user satisfaction through sentiment analysis and visual analytics. They develop visualizations to observe the relationships between aspect ratings and customer satisfaction before and during the COVID-19 pandemic. Fig. 2.2 compares scores for eight different aspects of TripAdvisor reviews before and during the pandemic. The colors represent a different aspect, where there is no natural or inherent relationship between the aspects and the colors. However, there is no natural relationship between the aspects and the colors. The visualization designers emphasized making the visualization's different marks meaningful. The shape of the markers (airplane-shaped), their colors, and the numbers on the side of the markers help to show relationships or draw attention and allow users to interpret the visualization in a meaningful way. In addition, interactions are used to enhance the insights generated. Sentiment analysis and data visualization found that airline ratings and customer satisfaction scores decreased after the COVID-19 outbreak.

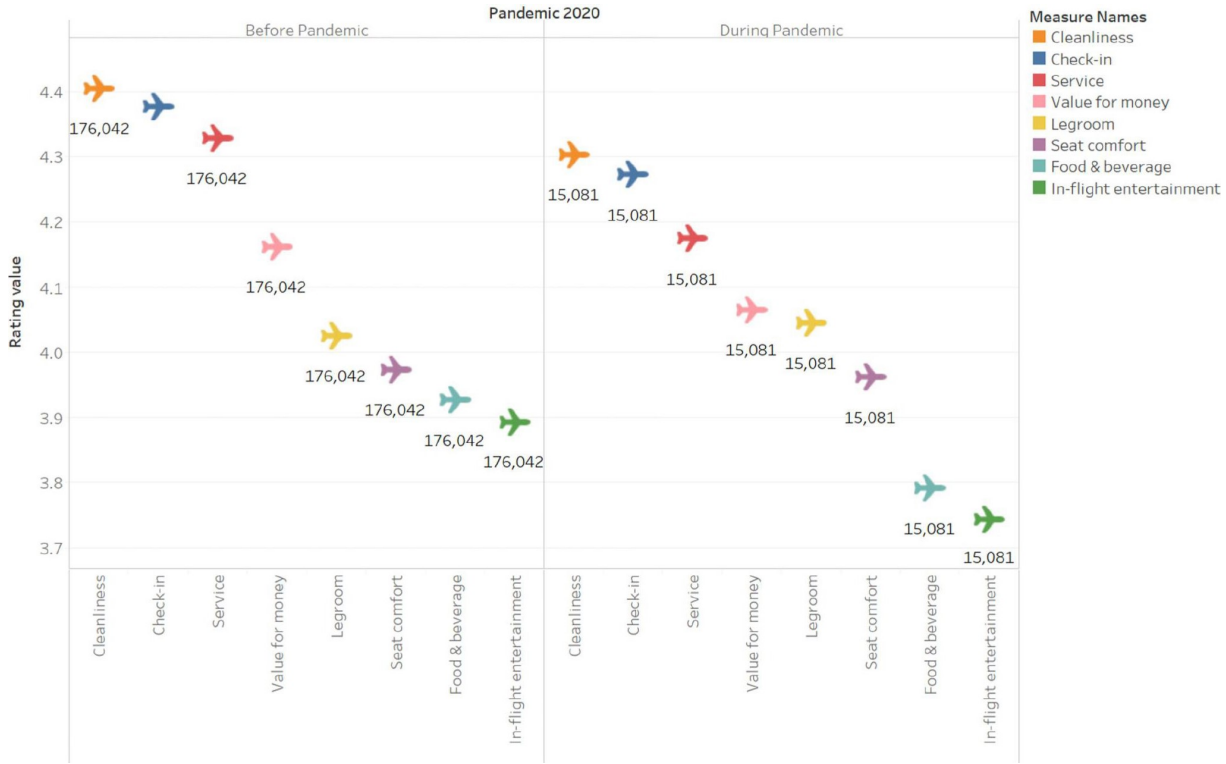


Figure 2.2: Aspect ratings for several airlines—before and during COVID-19. Taken from Chang et al. [2].

Geographic Visualization (GeoVis)

GeoVis aims to display geographic information so that the user can easily infer knowledge, generate hypotheses, and find solutions. According to Reichenbacher and Swienty [71], the primary task in designing geovisualizations is to avoid high complexity and too many visible elements that produce very dense visualizations. All these aspects must be considered when designing a geovisualization to allow users to quickly locate and decode the relevant geographic information to make inferences. Kraak [72] also mentions that maps are used to stimulate understanding of geospatial patterns, relationships that may exist, and trends in geospatial data itself.

For example, Jung [3] performs a quantitative analysis from a dataset composed of tweets. The dataset contains the following information: user ID, user description, geographic coordinates, created date and time, place type, and tweet text. The author takes a 5% sample of 14,848 tweets from the Dolly (Data on Local Life and You) project[1] at the University of Kentucky. The tweets are mapped with their geographic coordinates and are grouped by time. It is finally visualized that most of the tweets are clustered in the western part of King Country, belonging to the state of Washington and that more than half of the tweets were created between late afternoon and midnight.

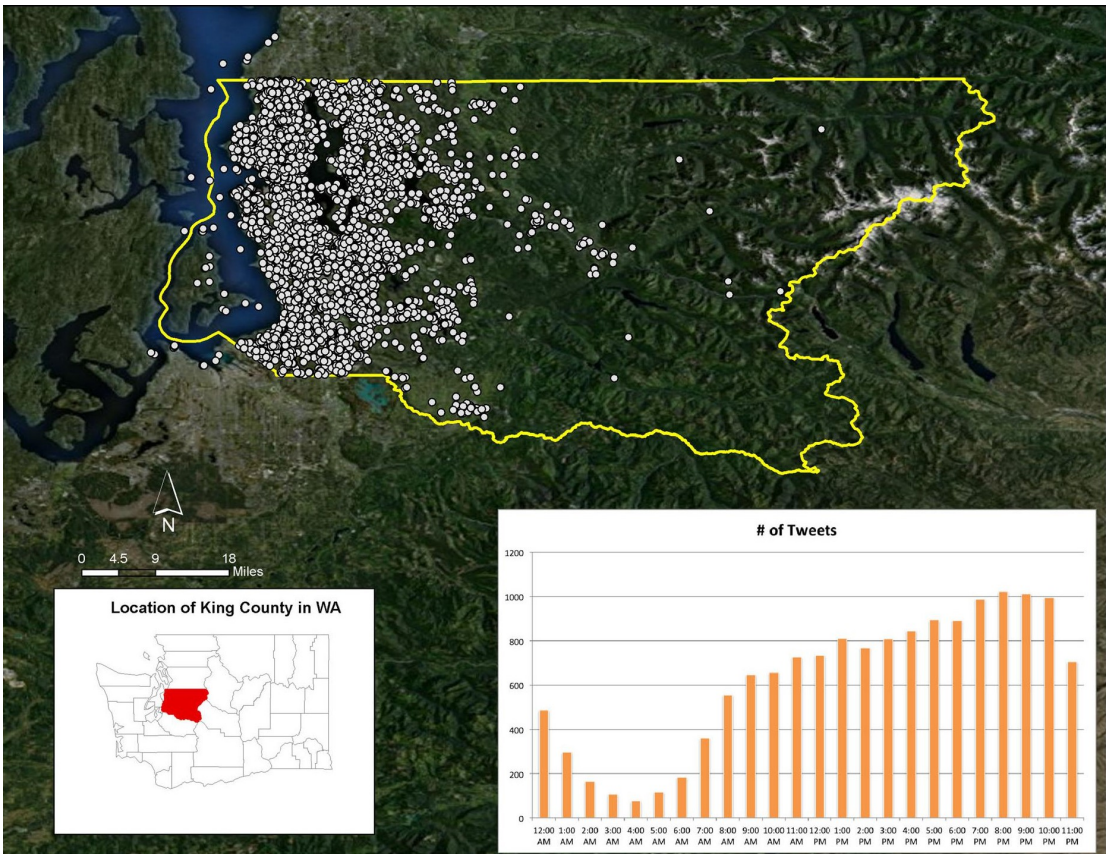


Figure 2.3: Spatial and temporal distribution of all tweets. Taken from Jung [3].

[1] `http://www.floatingsheep.org`

Another example of GeoVis is presented by Musto et al. [4]. They research hatred using a heat map showing areas of higher risk of hatred towards women in Italy (Fig. 2.4). The heat map was produced by performing sentiment analysis and content classification of tweets with geolocation. These tweets were obtained over a time span of 10 months, and in total, there were more than 1,600,000 tweets showing intolerance. This work is a potential sample of how geovisualization techniques can be used to implement monitoring strategies for social objectives.

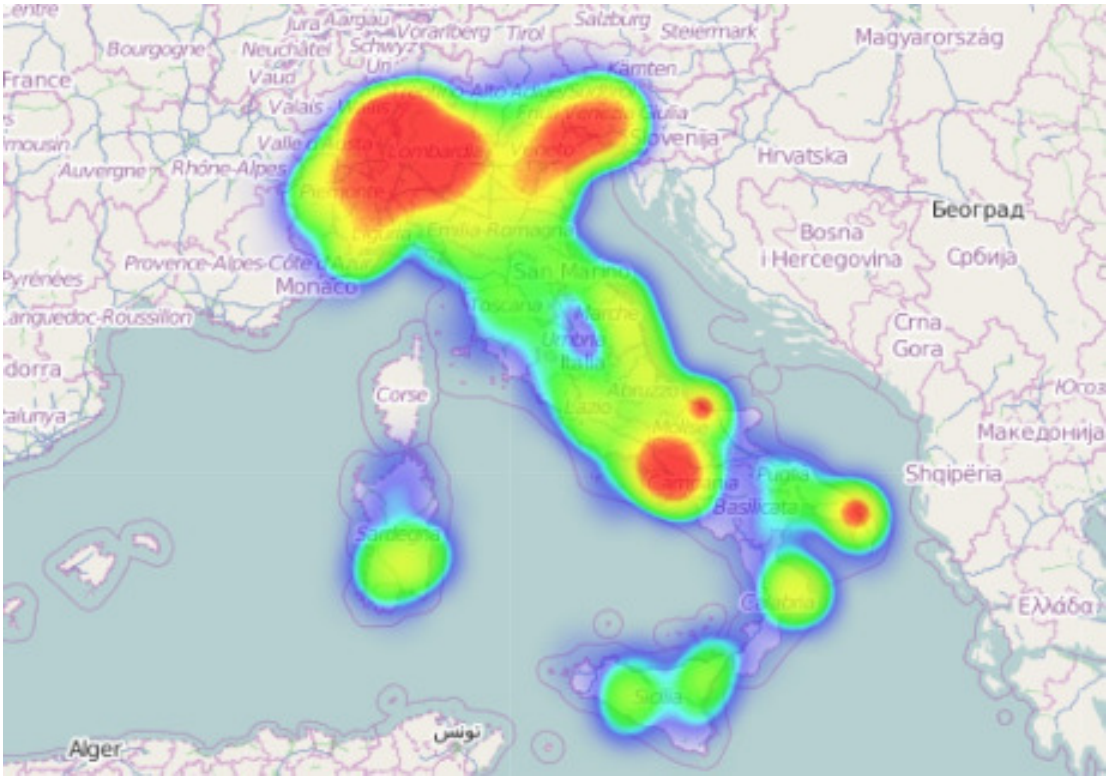


Figure 2.4: Map of hatred towards women in Italy. Taken from Musto et al. [4].

Scientific Visualization (SciVis)

SciVis aims to provide a high understanding of the data being investigated. According to Brodie et al. [73], SciVis uses techniques, computer graphics methods, user-interface methodology, image processing, system design, and signal processing to provide understanding and insights into the data. SciVis also benefits from the processing capabilities of today's computers and workstations with large memory capacities and graphics power to analyze and visualize large volumes of data, time-varying and multidimensional data (Brodie et al. [73]).

Scientific visualization (SciVis) generally deals with 3D objects and usually represents flows, volumes, and surfaces in a three-dimensional space. SciVis can be applied in a wide variety of fields such as molecular modeling, medical imaging, brain structure, function mathematics, simulations, geosciences, space exploration, astrophysics, computational fluid dynamics, and finite element analysis (McCormick et al. [74]). For example, Chavent et al.

[5] developed a GPU ray-emission algorithm that uses hyperboloids as atomic bonds. This technique allows rendering molecules ranging from a few atoms to molecular assemblies with more than 500,000 particles. Fig. 2.5 shows an example of the rendering produced in this work.

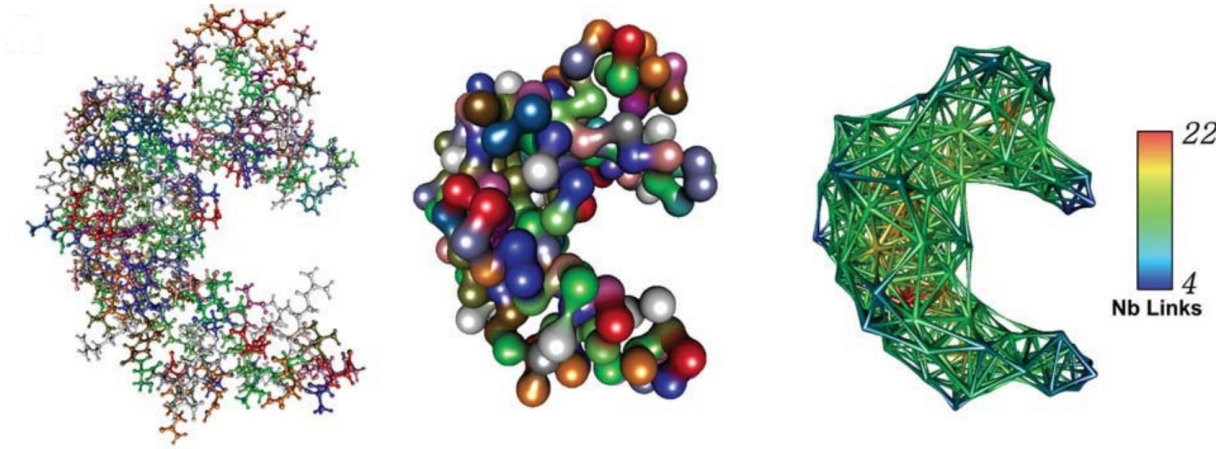


Figure 2.5: Use of the HyperBalls representation to depict coarse-grained models, Guanylate kinase enzyme. Taken from Chavent et al. [5].

Another example of SciVis can be found in the work of Ferstl et al. [6] where authors deal with uncertain flow visualization. Fig. 2.6 shows the trends in flow trajectories called Streamline Variability Plots. The images in Fig. 2.6 show trends in wind field fluxes from numerical weather prediction data. The streamlines show a stable forecast at a time later than the numerical data used.

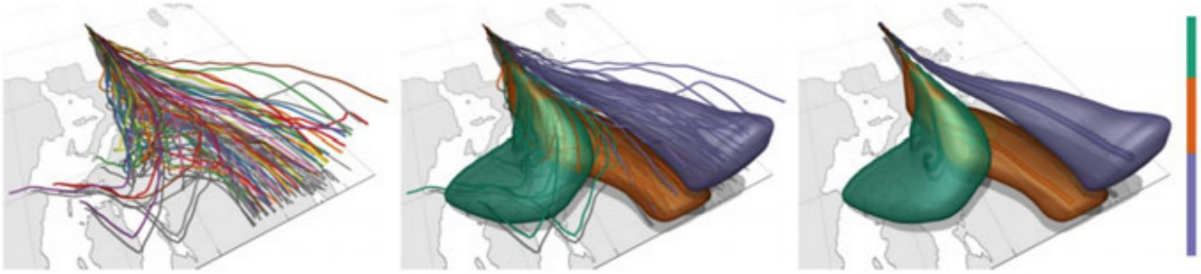


Figure 2.6: Streamline Variability Plots. Ferstl et al. [6].

Manovich [75] gives us an idea to differentiate InfoVis from SciVis/GeoVis, "Infovis uses arbitrary spatial arrangements of elements to represent relationships between data objects. SciVis and GeoVis usually work with a priory fixed spatial arrangement of the actual physical objects" and also informs us that these ideas and the two InfoVis principles above are not strict for all examples of visualizations. Sorapure [69] states that InfoVis also differs from SciVis in that the latter presents visual representations that refer to physical or material data, not abstract data, as InfoVis can do.

2.2 Information Visualization Pipeline

The role of the information visualization pipeline is to transform information into graphical data (Hansen and Johnson [76]). There are several ways to create a visualization of information (see McCormick et al. [74], Hansen and Johnson [76], Kim et al. [77], Haber and McNabb [78], Schumann and Müller [79], Chi [80], Chi and Riedl [81]). Chi [80] discusses several visualization techniques based on the data state reference model and presents their taxonomy, where they emphasized the importance of a visualization pipeline and similarities and differences between the different visualization techniques.

Santos and Brodlie [82] present a pipeline for information visualization, which describes the process of creating visual representations of data. Fig. 2.7 shows a pipeline proposed by the authors consisting of four main parts, each of which is described below according to Tominski [83]:

- Data Analysis: The raw data is prepared for display at this stage by applying a smoothing filter, interpolating missing values, or correcting erroneous measurements. This stage is computer-centered as there is little or no user interaction.
- Filtering: At this stage, the prepared data is taken, and only the data to be displayed is selected.
- Mapping: At this stage, the focus data is mapped into primitive geometrics, such as points or lines, and its attributes are mapped as colors, positions, sizes, or shapes. Mapping is essential because this is where the expressiveness and effectiveness of the visualization are achieved.
- Rendering: In the last stage, the geometric data is transformed into image data. In this final stage, the appropriate rendering technique must be applied to achieve either an image or an animation.

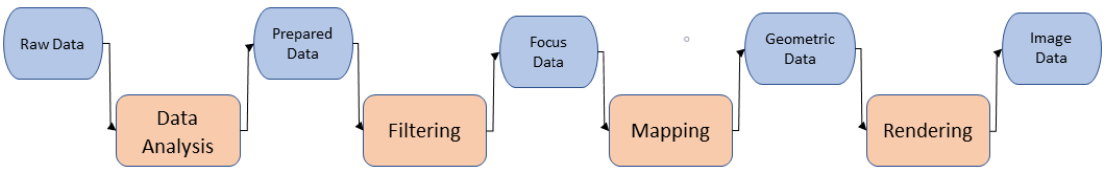


Figure 2.7: The visualization pipeline.

2.3 Visual Analytics Pipeline

A topic that is important to clarify is that of Visual Analytics. It is easy to confuse InfoVis and Visual Analytics, and this is because there are elements they have in common. Many times InfoVis works are related and intertwined with Visual Analytics works. However, InfoVis works do not always use advanced data analysis algorithms and are not always concerned with analytical tasks.

The Kleim et al. [84] paper explains that Visual Analytics is more than visualizations. Visual Analytics combines visualization and interactions with data analysis algorithms for decision making. The data analysis involved in Visual Analytics is done in an automated way so that the human factor intervenes when it is impossible to solve a problem in an automated way. Visual Analytics is closely related to Data mining, Data management, Human-computer interaction, and Perception and cognition.

InfoVis has focused on producing views, creating and using interaction techniques suitable for visualizing a specific dataset and topic. Nevertheless, it has not focused on how data from user interactions can be turned into intelligence for implicit analytical processes. For Visual Analytics, it is more of a priority to analyze data from the beginning and during the creation of the complete visualization. In this way, Visual Analytics focuses more on bringing intelligence to the analytical process by learning from the user and using visualization (Fadloun [44]). An example of Visual Analytics can be the following, using data mining, a dashboard can be created in Tableau to estimate the risk and schedule of COVID-19.

Visual analytics aims to create a transparent and analytical way of processing data and information (Kohlhammer et al. [85]). It examines processes rather than results to improve knowledge and decisions.

There are some definitions for Visual Analytics that are related to each other. For example, Wong and Thomas [86] defines it as "the science of analytical reasoning facilitated by interactive human-machine interfaces". Keim et al. [87] states, "Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning, and decision making based on huge and complex datasets". These definitions show that Visual Analytics can be applied to various disciplines with many processes and techniques that enable understanding and decision-making based on data. They propose a Visual Analytics Pipeline. Fig. 2.8 shows the interaction between data, visualizations, models, and users to obtain knowledge.

The Visual Analytics Pipeline shows the different steps represented by ovals and the transitions between them represented by arrows. According to Keim et al. [87] the data to be analyzed and visualized is usually heterogeneous, so the first step to be applied is the preprocessing and transformation of the data. This step includes several activities, such as data cleaning, normalization, and grouping. Once certain transformations have been applied to the data, the pipeline indicates that one of two paths can be followed first: visualize or apply some automatic analysis method. On the one hand, the data mining path can be applied to create a model that can be refined and evaluated. Visualizations allow interaction so parameters can be modified, or other analysis models can be selected. "Alternating between visual and automatic methods is characteristic for the visual analytics process and leads to a continuous refinement and verification of preliminary results" (Keim et al. [87]). On the other hand, the visual exploration path allows the user to interact with information to direct construct the model. If a visual exploration of the data is performed first, the user must confirm the hypotheses generated by an automated analysis. User interaction with the visualization is necessary to reveal useful information, for example, by zooming into different data areas. The results of the visualizations can be used to guide model building in the automated analysis. In summary, visual analytics allows:

- Gain insight from visualizations, automatic analysis, and previous interactions between visualizations, models, and human analysts.

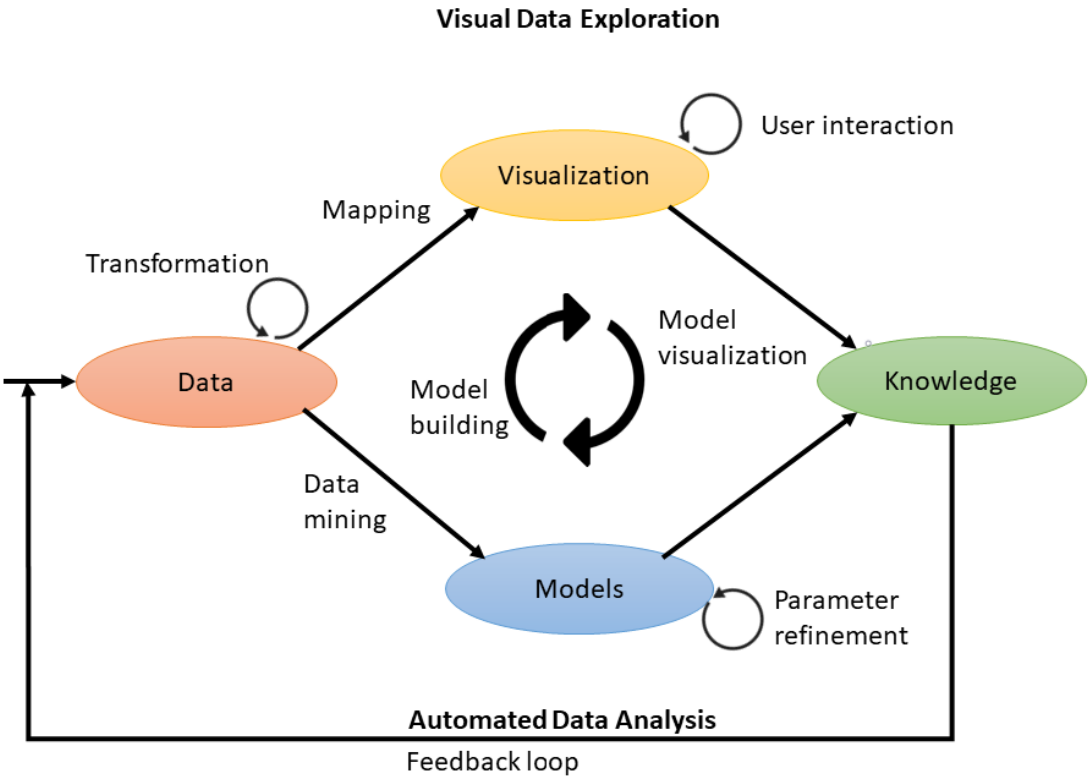


Figure 2.8: Visual Analytics pipeline.

- Synthesize information and derive insight from massive, dynamic, ambiguous, often conflicting data.
- Detect the expected and discover the unexpected.
- Provide timely, defensible, and understandable assessments.
- Communicate these assessments effectively for action.

2.4 Visualization Creation Process

In the section 2.2, you can see a pipeline for creating visualizations. However, an analysis framework summarizes this process, serves as a guide, and shows the key elements to consider when creating visualizations. Munzner [7] summarizes the process of generating visualizations with three questions and these are shown in Fig. 2.9.

Fig. 2.9 shows a guide for creating Visualizations, but there may be variations and other ways to create visualizations. The What? question refers to what data the user sees. The Why? refers to why the user is using the visualization tool. Finally, the How? refers to how the interaction and visual coding are constructed.

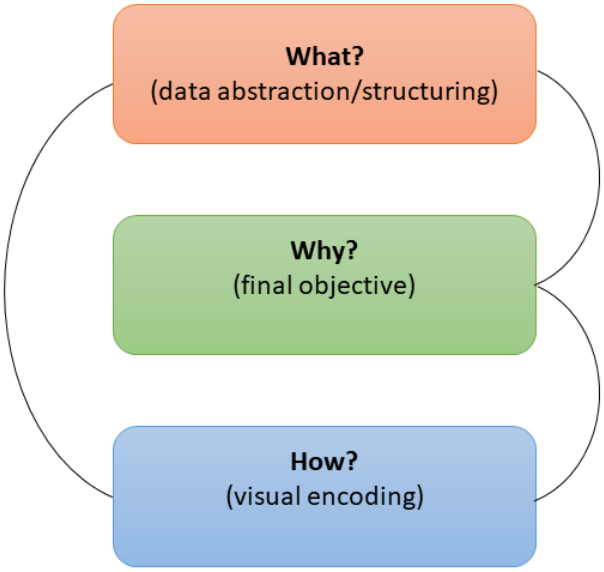


Figure 2.9: High level framework for visualization generation.

2.4.1 What?

Fig. 2.10 summarizes the what for creating Visualizations. The question addressed here involves identifying the abstraction and structure of the data that will be displayed, modified, and explored in the Visualization. The basis of Visualization is the data, which is why it is important to identify them. Fig. 2.10 shows the five main types of abstract input data that can be visualized. The main types of datasets are also shown. These dataset types are formed from combinations of the five main data types. For these dataset types, the complete data set may be available immediately in the form of a static file, or it may be available in the form of dynamic data that is processed incrementally. The types of data attributes can be categorical or ordered (ordinal or quantitative). Finally, the sorting direction of the attributes can be sequential, divergent, or cyclic (such as months of each year). In summary, this way of organizing the data allows us to know the structure of the data, its organization (type of dataset), and its attributes (variables of the observations).

2.4.2 Why?

Fig. 2.11 summarizes the why for creating Visualizations. The figure focuses on why you would use a visualization tool and shows it in actions and objectives. This section specifies the tasks to be solved when developing a visualization. Tasks are referred to in a general way, i.e., tasks that can be performed in different disciplines. Each discipline has its domain language, but certain operations are common.

The high-level actions for using a visualization tool are to consume or produce information. Consuming has three cases: discovery, presentation, and enjoyment, where discovering may involve generating or verifying a hypothesis.

At the medium level, the action is search. Searching can be classified according to whether the identity and location of the target are known or unknown: if the target is known, but its location is not, the search type is locating. The location is known, but the

Figure 2.10: What can be visualized: data, datasets, and attributes. Taken from Munzner [7].

target is not. The search type is searching; finally, neither the target nor the location is known. The search type is exploration.

At the lowest level of actions, we have the queries, which can have three scopes: identify a target, compare some targets and summarize all targets. In the case of identifying a target, if a search returns known objects, then their characteristics are returned. For example, a map's proportion of some value can be shown by saturating a color. More sophisticated language expressions may be required for the comparison case, and an overview of everything can be provided for the summary scope.

The actions mentioned above are performed on targets, and the classification of these targets is explained below. The targets for all data types are to find trends, characteristics, and outliers. The target can be an attribute's value (such as minimum or maximum). Also, for one attribute, the target can be to find the distribution of all values. With several attributes, the objective may be to find dependencies, correlations, or similarities. With networked data, the objective may be the network's general topology and routes, among others. Finally, with spatial data, the objective can be the shape.

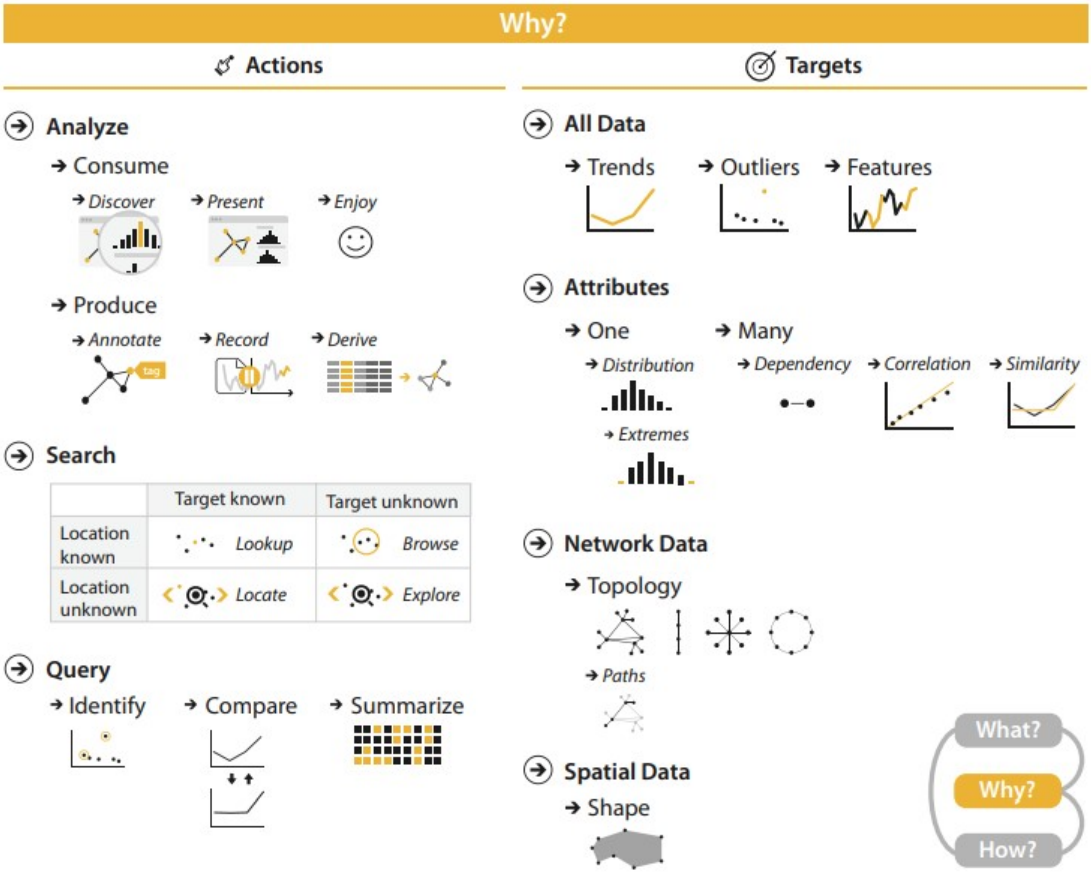


Figure 2.11: Why use visualization in terms of actions and targets. Taken from Munzner [7].

2.4.3 How?

Fig. 2.12 summarizes how a visual language can be constructed from design options. In other words, how deals with the design and implementation of visualizations. It is seen at a high level in four main classes. Data encoding has several options on how to organize data spatially. These are: expressing values, separating, sorting, aligning regions, and using given spatial data. For Map, it includes mapping data into non-spatial visual channels, such as color, size, angle, shape, motion, and more. (see Section 2.5 for more information)

The manipulation group shows that you can change any aspect of the view, select elements, and navigate between elements to change the viewpoint. The facets group shows options to juxtapose and coordinate views and how to split data into views (partition), and at last, superimpose layers of views. Finally, the reduction group shows that you can filter data and options for aggregating data and embedding information in a view.

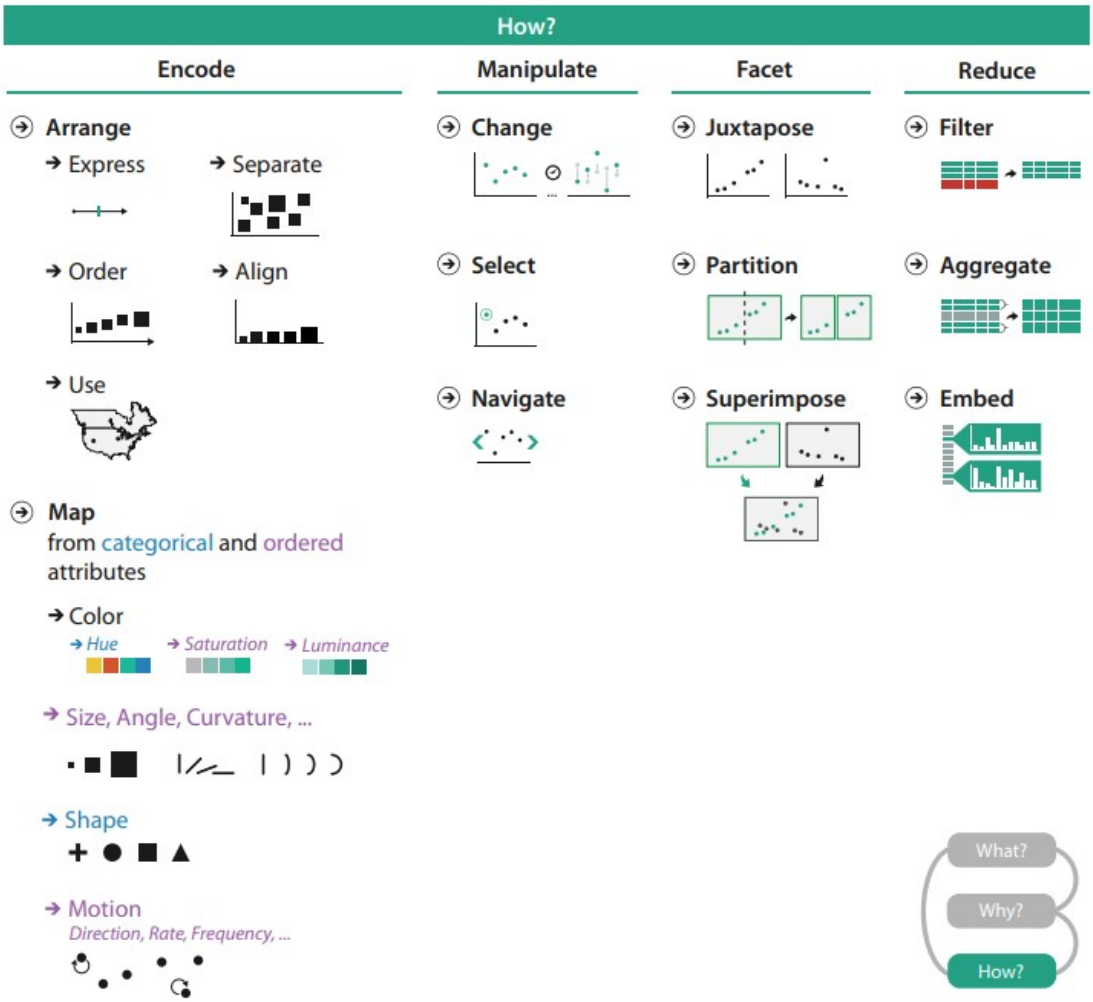


Figure 2.12: How to design visualization idioms: encode, manipulate, facet, and reduce. Taken from Munzner [7].

It is no coincidence that Munzner [7], in his work, has placed coding, manipulation, faceting, and reduction within the how. All of these aspects are essential to the design and implementation of a visualization. In the work of Dimara and Perin [88], the categorizations

for the allowed actions of a person within a data interface are mentioned. Within the perceptualization actions (operating on perceptual formatting of data or visual change) are mapping actions and presentation data actions. These two actions correspond exactly to the taxonomy elements presented in Fig. 2.12.

2.4.4 What-Why-How Example

The visualization analysis framework presented in this section can be useful for comparative analysis. The following examines two visualization tools with different answers for how the idiom is designed and have the same context in the why and what. Fig. 2.13 shows tools for the visualization of trees. Fig. 2.14 shows that the two tools take the same input data, a large tree composed of nodes and links. The why is also the same since both tools are used for the same purpose: to locate the paths between nodes and identify them. In the case of idioms, some are the same. Both tools allow the user to navigate and select a path. The two tools differ in how they manipulate and organize the visualization elements. SpaceTree links the selection to a change in what is displayed, automatically aggregating and filtering the selected elements. Whereas, TreeJuxtaposer allows the user to arrange areas of the tree, so that good visibility of the areas of interest is maintained.

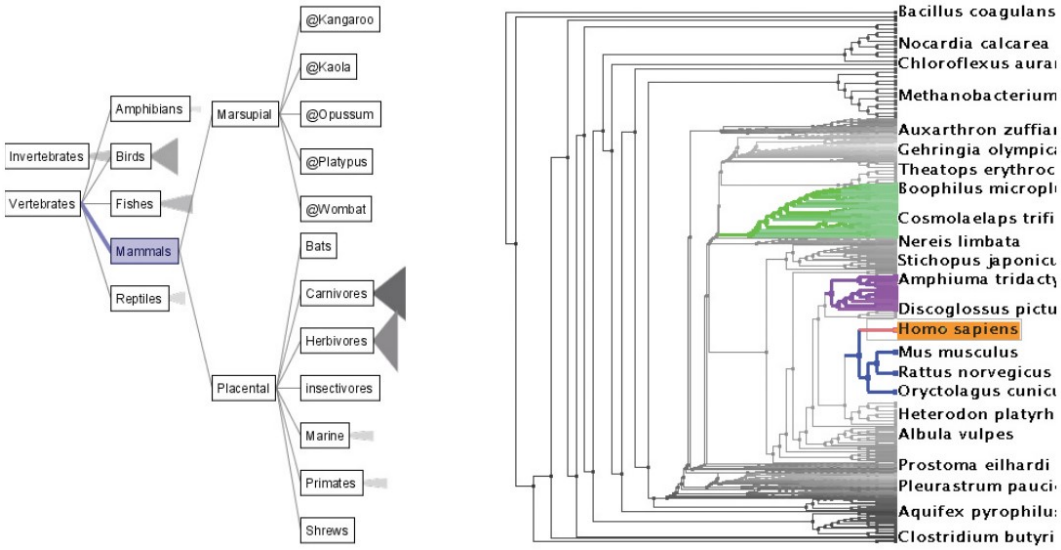


Figure 2.13: SpaceTree (Left) and TreeJuxtaposer (right). Taken from Munzner [7].

2.5 Visual Encoding

An important issue is how data is encoded into elements that are in the visual domain in a visualization tool. The visual domain is a multidimensional space whose axes and domains are the elements that are perceived in the visualizations. As seen in the previous section, to encode data, it is necessary to consider the organization in the visual space (arrange) and to define how the attributes of the observations define the visual properties (map).

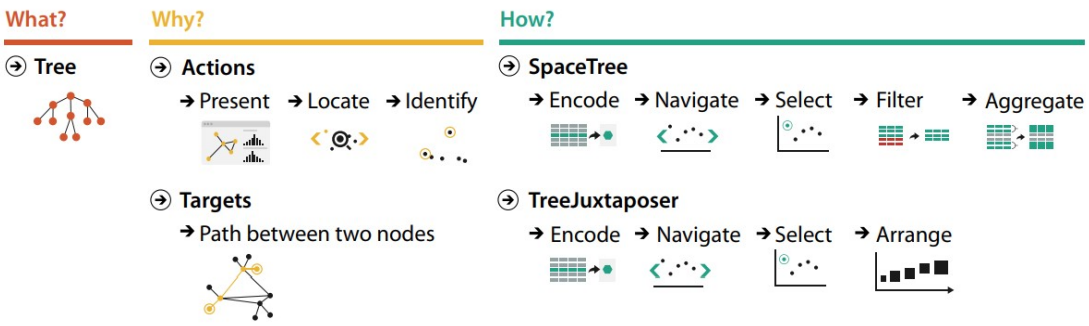


Figure 2.14: Visualization analysis framework to compare SpaceTree and TreeJuxtaposer. Taken from Munzner [7].

The decisions to be taken should be according to the attributes (quantitative, ordered, qualitative, categorical) and should allow the optimal development of the mapping.

Visual coding is the process of assigning graphical properties to data attributes. According to Munzner, this process is done through graphical elements (marks) and visual channels controlling the appearance. In Munzner's words, "The core of the design space of visual encodings can be described as an orthogonal combination of two aspects: graphical elements called marks, and visual channels to control their appearance ".

2.5.1 Markers

Markers are primitive geometrical objects in an image and are classified according to the required spatial dimensions. Examples of markers are shown in Fig. 2.15. A zero-dimensional marker is a point, a one-dimensional marker is a line, and a two-dimensional marker is an area. There are also 3D markers, but they are not very common. A marker can represent a node or link for networked data sets. Link markers represent relationships between elements. There are two types of link markers: containment and connection. Containment tags show hierarchical relationships using areas. Connection marks are lines that show pairwise relationships between two elements.

2.5.2 Channels

Channels allow to govern the appearance of the markers, and this is independent of the dimensionality of the marker. In Fig. 2.16, you can see a sample of the existing channels. They can be position, shape, size, color, and inclination. These channels can be combined, and their choice depends on the type of mark. For example, a dot has only one dimension. Also, there are motion-oriented channels. They can be oscillating circles with straight jumps, directions of movement, and speed. Curvature is also a visual channel, as well as the shape of an object.

The use of visual channels in encoding is governed by two principles, the principle of expressiveness and effectiveness. The expressiveness principle dictates that the visual encoding should express all and only the information in the data set's attributes. Within this principle, a basic expression is that ordered data should be displayed so that the

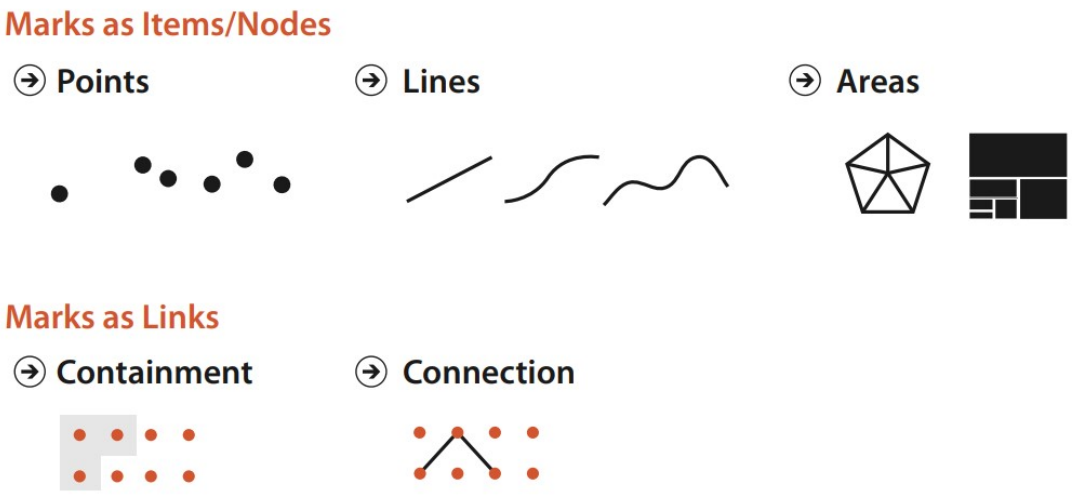


Figure 2.15: Markers as individual elements or as links between elements. Taken from Munzner [7].

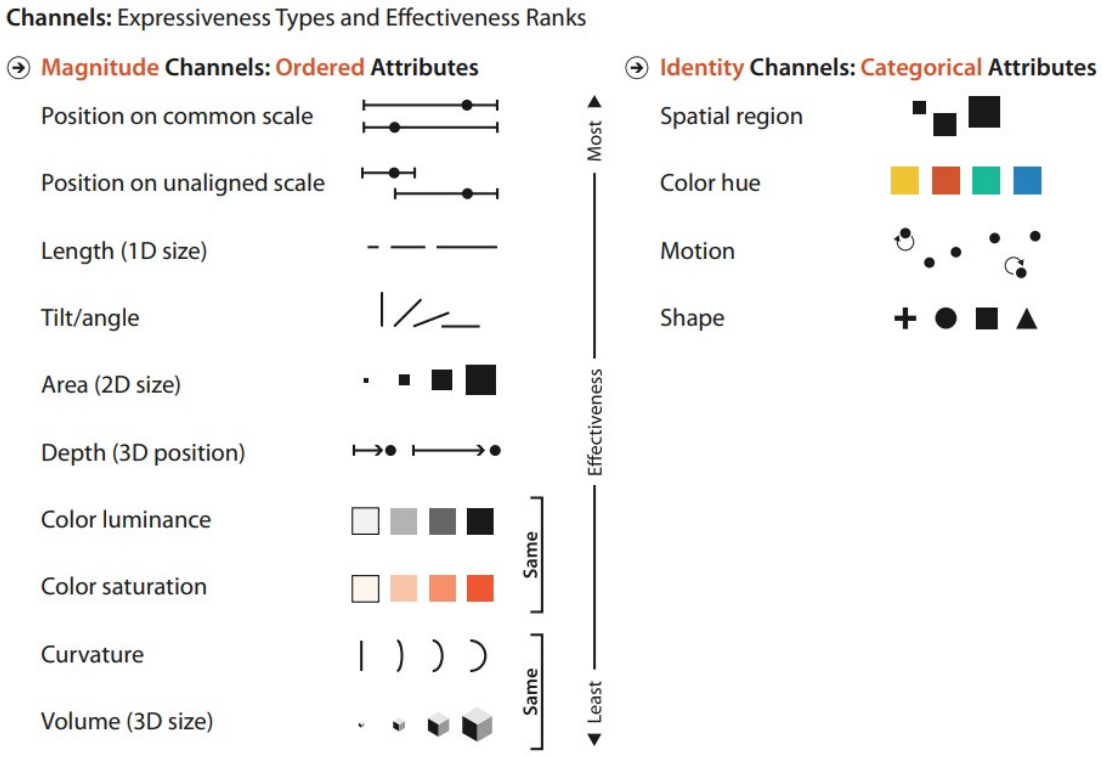


Figure 2.16: Effectiveness of channels and dependence of channel expressivity on matching attributes. Taken from Munzner[7].

perceptual system intrinsically senses it as ordered. Unordered data should not be displayed so that it is perceived as having an order that does not exist. The principle of effectiveness states that the attribute's importance should match the channel's perceptibility. The most important attributes should be coded with the most effective channels to be most noticeable. The same is true for less important attributes that should be coded with less effective channels.

In Fig. 2.16, there is a classification for the channels: Magnitude and Identity. Magnitude channels are best suited for ordered attributes, both ordinal and quantitative. Identity channels are best suited for categorical attributes that have no intrinsic order. Fundamentally, they are classified in this way, which is why the attributes are classified as categorical and ordered when discussing what can be visualized.

Munzner [7], in Fig. 2.16, presents a ranking of the effectiveness of the channels so that you can choose the most appropriate channels for different types of attributes you need to display. The best channels are at the top of the image and are the most effective. The image indicates that position on a common scale is the most effective channel for expressing magnitude. Another effective channel is length, but it is difficult to know how much one element is longer than another. The same is true for luminance or color intensity. It is possible to tell how dark or intense the color is, but not how much. If you need to determine the difference between magnitudes, luminance and saturation are not good channels, but if you need to determine an order, these channels are acceptable. In the Identity channels, the most effective is the position in space, followed by color hue. This last channel allows for separate categories, and using a rainbow color scale with quantitative values is a mistake. The motion channel is also good, especially for a single set of moving elements against the rest of the static elements.

Generally, the most commonly used channels can be the color hue for separating categories, color luminance for displaying sorted data, size for quantitative data, and texture (dashed, broken lines) for smooth coding.

2.6 Interactions

In the work of Dimara and Perin [88], a generalized definition of interaction is presented: "Interaction for visualization is the interplay between a person and a data interface involving a data-related intent, at least one action from the person and an interface reaction that is perceived as such". The interaction allows experimenting with what-if scenarios and provokes curiosity in the users. As seen in the definition, for an interaction to occur, a person or a user must participate, and depending on an action, a change in the visualization can be triggered. Also, a computer must intervene in the background by the processing power. Without processing power, seeing more than static visualizations would not be possible.

The visual representations that arise from the encoded data are essential for visualizations, but also important are the interactions that can be made with the visualization elements. The importance of interaction in visualizations can be reflected in the statement by Thomas and Cook [89]: "Visual representations alone cannot satisfy analytical needs. Interaction techniques are required to support the dialogue between the analyst and the data ".

In section 2.4.3, we have already mentioned that you can manipulate views, change how elements are displayed, and even have options to reduce the information of the visualization. All these actions help to identify key features, find hidden patterns, analyze, search, explore and understand some phenomena. Interactions also become essential when there is a large amount of data in the visualization and is complex. Interactions can amplify the granularity of detail in visualization and can also serve to create different ways of representing and summarizing data. With interaction, you can change the view and focus on specific elements. The latter is impossible in a static view since it focuses on simple tasks and working with simple data sets. At last, it can be challenging for interactions to handle large and complex data. The challenge has to do with limitations in memory, computing power, high-resolution displays, and more, but these issues are beyond the scope of this paper.

The most common interactions are explored below, specifically in actions that operate on the specific presentation of data, such as: marking something as interesting, navigating (e.g., pan and zoom), stylizing, highlighting, and decorating.

2.6.1 Zoom

This interaction allows viewing the information at different levels of detail (Tominski [90]). The user can freely view the information space at different scales. According to Furnas [91], the display can serve as a window into the information space. Different parts of the display can be accessed by moving a window, which can be resized to adjust the scale of what is seen. According to Bederson [92], there are several ways to change the scale, such as using the mouse wheel, the mouse drag on laptops, and touch screens with a two-finger gesture. There are two types of zooming:

- Geometric zoom: In this type of zoom, the size of objects is changed by scaling the view. When zooming in on the image, the view is closer to the plane, fewer elements will be seen, and they will be larger. When zooming out, you will see fewer elements, which will be smaller.
- Semantic zoom: This type of zoom is more complex than geometric zoom. What characterizes this zoom is that the displayed elements can look completely different at different scales. In this case the element represented is adapted to the number of pixels available in the region of the image space.

2.6.2 Selection

This type of interaction allows users to focus on specific visualization elements. This interaction aims to identify elements of interest for a more focused exploration and analysis. Usually, this type of interaction is implemented by clicking with a mouse on the element to be selected. Alternatively, selection can be implemented by using a lasso selection tool.

2.6.3 Highlighting

Highlighting is an interaction closely related to selection, but they are not the same. In this interaction, certain selected elements or desired elements change their visual appearance to stand out from the other elements. The selection should cause highlighting with immediate visual feedback. A common way to highlight objects is by changing their color. Another option is to highlight the outline of objects. These options can work very well for large elements, but if the elements are very small, you can increase the size of an element or the width of a link to highlight the objects.

2.6.4 Constrained Navigation

There are visualizations with free navigation. This type of navigation allows unrestricted camera movement anywhere. The disadvantage of this type of navigation is that there may be difficulties in trying to find a suitable viewpoint. The camera may be pointing to an empty or complex location where it would be difficult to return to an appropriate view state. One way to overcome this may be with a reset view button.

Constrained navigation tries to limit the behavior of the camera. One of the most common view restriction approaches is limiting the zoom range. With the zoom range limitation, the camera cannot zoom beyond where all drawn elements are visible. Nor can it zoom in closer than the smallest object.

2.7 Common Problems in Visualizations

In the development of visualizations, problems may make it difficult to understand or manage the information presented. Some of the most known problems are discussed below.

2.7.1 Misleading colors

Colors in visualizations can be important to differentiate elements or find relationships in them. A common mistake is to use too many colors. Using too many colors can cause confusion and visual overload to the user. Another common mistake is to use colors that do not have good contrast. This will make it impossible for the user to differentiate between colors. Another problem is using high degrees of color contrast when the disparities in the values are not as great as the contrast shows. Finally, another mistake related to color is making an unsafe color choice for color-blind users.

2.7.2 Clutter

This problem can occur when the information to be displayed is too abundant. Humans are not good at calculating the meaning of multiple abstract values visually. Too much information can be overwhelming and confusing to users. An image with numerous complex data collections can be difficult to analyze, and details cannot be easily perceived.

2.7.3 Occlusion

Occlusion occurs when some objects cannot be seen because they are hidden behind others. This problem can lead to omitting certain objects that may be important for the visualization. This problem is often solved by camera movements or by readjusting the objects in the display.

2.8 Visualization Validation

Munzner [8] proposes a visualization model to design and validate visualization systems without neglecting how these systems are evaluated and when the model should be chosen. Validation of a proposed visualization system is important to determine if the Vis tool meets the design objectives.

This nested model consisting of four layers: i) characterize the tasks and data in the vocabulary of the problem domain, ii) abstract into operations and data types, iii) design visual encoding and interaction techniques, and iv) create algorithms to execute these techniques efficiently. Each of these layers has its way of being validated and provides a model for analyzing existing visualization systems or guiding the design of proprietary systems.

Fig. 2.17 shows the model proposed by Munzner [8]. The first two levels address the why and what questions seen in section 2.4. The second and third levels address the how question. The levels are nested, so the upper level's output is the lower level's input. The nested levels represent that if an error occurs in a higher level, this error is transmitted to the lower levels. For instance, if an error is made in the abstraction stage, even if the next two stages are perfect, a visualization system will not be created that solves the supposed problem.

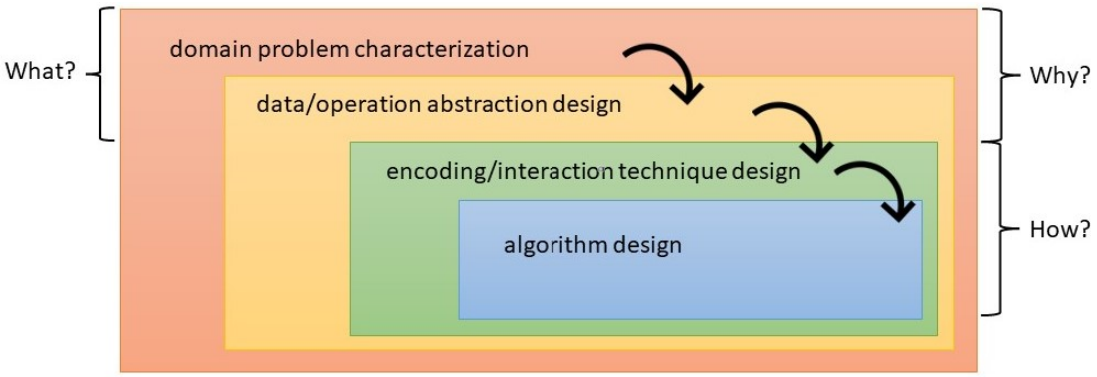


Figure 2.17: Nested visualization model with its four layers.

The steps of this visualization model are detailed below:

1. Domain problem and data characterization: The visualization designer must know the tasks and data the target users will handle in a specific domain. The designer should also know about the target audience's problems to present a human-centered design. This level's output should have highly detailed questions or actions made by the users to collect the data.

2. Operation and data type abstraction: In this stage, the problems and data are mapped from the domain-specific vocabulary to an information visualization vocabulary. The output of this step is descriptions of operations and generic data types. For example, determine domain parameters and hypotheses; compare, query, and correlate anomalies. In this step, the raw data is transformed into data types that visualization techniques can handle. To illustrate, as stated by Munzner [8] "Quantitative data can be binned into ordered or categorical data, tabular data can be transformed into relational data with thresholding, and so on". The visualization designer should not take the first attraction that comes to mind and move on to the next level as it may result in visualizations with additional visual loads. Additional visual loads often make it difficult for the end-user to understand.

3. Visual encoding and interaction design: The design of visual encoding has been widely treated and studied in the literature, unlike the interaction design theory for visualizations which has been less developed. In the proposed model, visual encoding and interaction are considered part of the same level since they are interdependent.

4. Algorithm design: This level deals with the design and creation of an algorithm. The algorithm is helpful for the realization of visual coding and interaction designs in an automatic way.

2.8.1 Threats and Validation

As mentioned above, each level has a way to validate it, and there are also several threats to validation. As seen in the nested model (Fig. 2.17), the higher levels influence the lower levels. Thus, when validating the levels, there will be immediate and subsequent validations. It should be emphasized that the validation of external levels is not immediate. The levels that have immediate validation provide partial evidence of success. They do not demonstrate in their entirety that the threat to a level was entirely resolved. Additionally, as mentioned by Munzner [8] "a poor showing of a validation test that appears to invalidate a choice at an outer level may in fact be due to a poor choice at one of the levels inside it".

A summary of the threats and validations of the visualization model is shown in Fig. 2.18. The length of red lines indicates the magnitude of the dependency between the threat and the subsequent validation. Validation of an outer level requires that an inner level be addressed.

The model for designing visualizations shown in Fig. 2.17 is presented as-is for simplicity. Although the four stages are rarely performed in sequence, separating them into four stages helps analyze whether each level has been addressed correctly. In a user-centered design, there is usually an interactive refinement process, where one layer will feed back and advance to refine others. The validation and implementation process is explained below:

1. Domain threats: The main threat in the first stage of the visualization model is that the problem to be solved or analyzed is mischaracterized, i.e., that the target users do not have the problems indicated. The immediate way to validate this is through observation and interviews with the target audience. The way to validate this level downstream is by observing the adoption rate of the visualization system. The rate tells what the target users do of their preference.

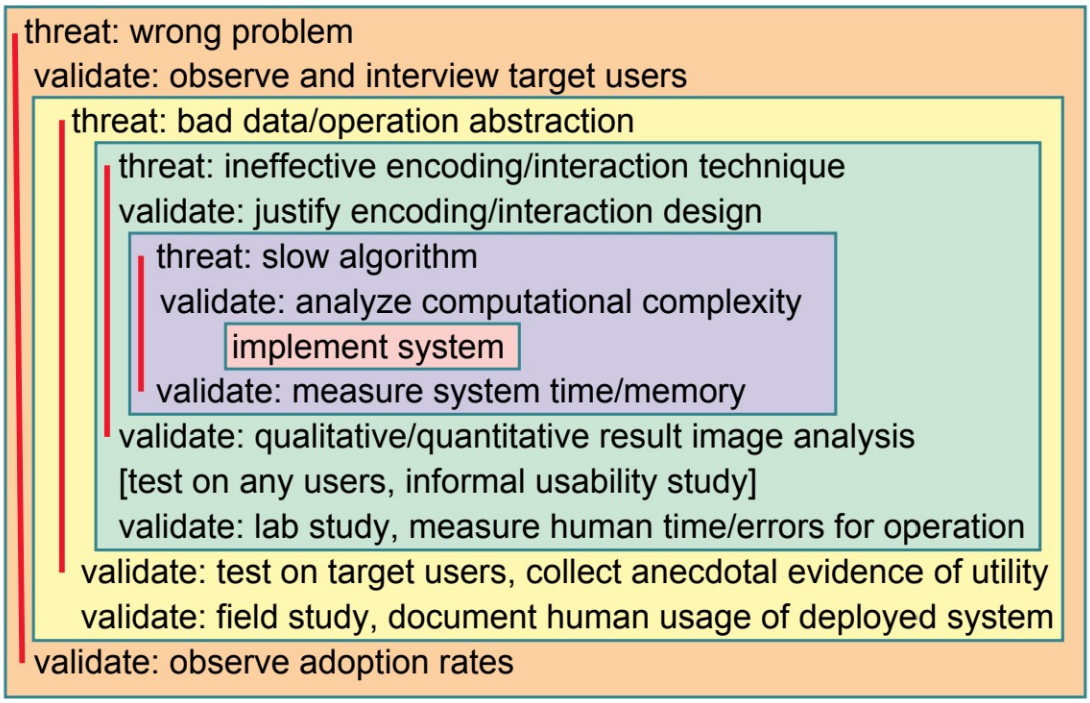


Figure 2.18: Threats and validation in the nested model. Taken from Munzner [8].

2. Abstraction threats: Threats at the abstraction level involve the operations and data types chosen, not solving the target audience's problem. One way to validate the threat is to test the system with the target users doing their job. Another way to validate the threat is to observe and document as a study how the target audience uses the system as part of their real-world workflow.

3. Encoding and interaction threats: The threat at this stage manifests itself if the chosen visual design does not effectively communicate the desired abstraction. One approach to validate this threat is to justify the design concerning known cognitive and perceptual principles (Tory and Moller [93], Zuk et al. [94]). Another way is to discuss the options in a design study document. One way to validate this threat is through a formal user study in the form of a laboratory experiment. For example, according to Munzner [8] "A group of people use the implemented system to carry out assigned tasks, usually with both quantitative measurements of the time spent and errors made and qualitative measurements such as preference". Another way to validate is the presentation of qualitative discussion of results in the form of images or videos. Finally, a third way to validate is quantitatively measuring the resulting images (Ware et al. [95]).

4. Algorithm threats: At this level, the threat is that the algorithm is not optimal in both time and memory performance. One way to validate this is by analyzing the algorithm's computational complexity. The way to validate at lower levels is to measure clock time and memory performance. The other threat at this level is that the algorithm is incorrect or does not comply with the visual coding specification

or the established design. One way to validate this threat is to check if the image created meets the correction objectives of the algorithm.

2.8.2 Angles of Attack

According to Munzner [7], there are two ways to approach the development of a visualization. The first is "Problem-driven", which aims at developing a problem-based visualization tool. In this case, you start working on some users' problems and try to design a solution or a tool that will improve their work or satisfy their needs. Munzner [7] mentions that this first approach is referred to as a design studio in the literature. "Problem-driven" usually tries to solve a problem using existing interactions and visual encodings. However, it does not mean that a new abstraction language that satisfies the problem cannot be proposed. Munzner's nested model can be useful in "Problem-driven" to avoid omitting some important steps in problem-based work. Finally, Munzner's nested model in problem-driven should be attacked from the *top down*.

The second approach to developing a visualization tool is the so-called "Technique-driven" approach. This work starts with a new idea for a new visual coding, interaction language, or a new algorithm. Munzner's nested model must be approached from the *bottom up*. Start at one of the two lower levels and focus on the design at that level.

2.9 Data

One of the most important parts of generating information visualizations is data. An increasing amount of data is constantly being produced and recorded by human-created information systems, and data literacy is an essential aspect of the constant amount of data created. According to Sorapure [69], most definitions of data literacy include the ability to access, understand and manipulate data of various types. With data literacy, it is possible to ask questions, make decisions based on data, create arguments, and use tools to process, manipulate, and represent data to communicate.

Bertini and Lallanne [96] report that one goal of information visualization is to extract knowledge from raw data. They present definitions of what is data, information, knowledge, and other terms. According to the authors, data has to do with collections of facts that are collected through observations, measurements, and experiments. The data can be numbers, words, or even images. It is also mentioned that the data in InfoVis are generally referred to as abstract data since they usually have no spatial structure to map them in some geometry.

2.9.1 Data Structure

According to Andrienko and Andrienko [9], data has two types of components: referrers and attributes, also known as independent and dependent variables. A dataset in an abstract way can be seen as a correspondence between references, i.e., values of the referrers, and characteristics (values of the attributes). Authors indicate that there are three major types of referrers: time (e.g., days), space (e.g., enumeration districts), population or groups of

objects (e.g., school subjects). Two examples can be mentioned to understand more about the two types of data components :

- In a dataset of daily stock prices. The referrer is the time, its values, i.e., its references are the moments as days. The attribute is the price of the stock, its values, i.e., it characteristics, are the price each day.
- Census dataset of a country. The referrer is the set of enumerated districts, where each district is the reference. The various counts, the total population, and the number of women are the attributes, and the counts themselves are the characteristics.

A dataset consists of elements with a similar structure called data record. For example, the grades of two student exams of a course in each record have three components: the student, the grade of the first and second exam.

As previously mentioned, data has two components: referrers and attributes. The referrer is the one that defines the context in which the data was obtained, and the attributes are the results obtained from the measurements, calculations, observations, and more in a context. Take the example of a country census. The district and the year define the context in which the population was measured. The references (values of the referrers) do not contain information about a phenomenon. They relate the characteristics (values of the attributes) to places and time. The characteristic itself is the one that is directly related to the phenomenon studied as "number of population". Finally, another way the references differ from the values of the attributes is that the former can be chosen arbitrarily, and the characteristics are determined by choice. For this reason, they can be called independent and dependent variables. For example, when conducting a census, the census takers can arbitrarily decide the date. They can change the census boundaries or decide whether to take larger or smaller territories, but the measured number will depend on the above.

2.9.2 Data Properties

Data and its components can be distinguished according to their properties. According to Klir [97], the most critical properties of the data are "the ordering of the set elements, the existence of distances between the elements, and continuity." Andrienko and Andrienko mention that the ordering can be unordered, partially ordered, or entirely (linearly) ordered. Distance is the difference between two numbers. Continuity refers to whether the data is discrete, such as when measuring various objects, or continuous, such as when measuring temperature. Another important property of data is completeness. Incompletely specified data have missing values, and these cases should be handled cautiously. For instance, time is linearly ordered continuously and has distances. Space is continuous with distances, but there is no natural order between the elements. However, there may be ways to order them, such as using a coordinate system to establish total ordering when existing a dimension. The population of a group of objects forms a discrete set with no order or distance.

Data and visualizations can be used in many fields such as communication, data mining exploration and analysis, biology, sociology, cartography, and many more (Fekete [98]).

2.10 Spatial-temporal Data

Spatio-temporal data has been generated and is widely available. This Spatio-temporal data is produced in various domains such as communication, environmental sciences, human mobility, and more. What characterizes this data from other data types is that they have both spatial and temporal properties. Zhang et al. [99] mention that thanks to new developments in location and wireless communication technologies, a large amount of spatio-temporal is available. They describe spatial and temporal properties:

- Spatial properties: These properties are based on geographical hierarchy and distance. Geographical hierarchy refers to the fact that there is information about the geographical location and that it has different granularities. Places at the top level of the hierarchy will have coarser granularities, while locations at lower levels of the hierarchy will have smaller granularities. There is also a certain geographical distance between locations that measure their correlation.

- Temporal properties: Thee properties are based on closeness, period, or time trend. Spatio-temporal data have timestamps that order the data chronologically, data sequences can be created, and periodic patterns can be determined with this property.

Analysis of spatio-temporal data is usually conditional (see Schabenberger and Gotway [100]), meaning that either the temporal dimension is analyzed first and then the spatial dimension or vice versa. Still, this way of analyzing spatio-temporal data is incomplete because space and time should not be separated. Examples of data with Spatio-temporal characteristics are satellite images of the planet Earth, temperature readings from different seasons, voting results, trajectories of people or animals, and volcano eruptions, to name a few examples (Pebesma [101]).

To analyze this data, Peuquet [102] proposes a triad framework that relates three elements that make up the triad, they are: the place of the view (where), the object of the view (what), and time (when) (see Fig.2.19).

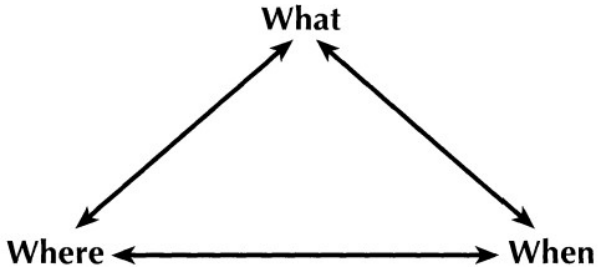


Figure 2.19: Components of the Triad framework by Peuquet.

The elements that are interrelated in the triad framework allow three types of questions to be asked:

- What: This question refers to the object or objects represented at a specific location or locations and at a specific time or set of times.

- Where: Refers to the location of a given object or set of objects at a given time or set of times.

- When: Refers to the time that an object or objects have a given location.

2.11 Movement Data

Movement data change their spatial location over time (Andrienko, G. and Andrienko, N. [103]). This kind of data has the following properties:

- Temporal properties: The size of the time intervals, whether the size of the time intervals is constant or variable, whether the measurements were performed over the entire time of the data or on a sample.
- Spatial properties: The slightest change exists in an object's position, the spatial precision, whether the captured positions are everywhere or how they are distributed.
- Properties of the moving sets: Whether there is a single moving object, a few, or many movers. Whether there is data on all the movers of interest or only a sample. Finally, whether the selection of movers is representative.
- Data collection properties: Such as the accuracy in the positions given by a sensor, the degree of error that may exist in the edits, positions that have been missed, and what those cut positions mean.

The properties described above depend on how the data are collected. The configuration and features of technological devices from which the data is collected will influence the properties of the data. For example, data in motion can be obtained by a GPS and will have a high data resolution if the device allows it.

Movement data are discrete. For two successive instants of time, the space of time between them is not defined. On the one hand, there are movement data that have a high-resolution, temporal and spatial, these data allow interpolations. In this way, a continuous path can be recreated. These high-resolution data are known as quasi-continuous. On the other hand, there are also movement data that do not allow the creation of interpolations, and these data are called episodic. Episodic data are produced by data collection methods where position measurements are not frequent. For example, devices used to track the movement of wild animals do not make regular position measurements. There are several methods for storing the position of an object (Andrienko et al. [104]):

- Time-based: The positions of moving objects are stored at regular time intervals.
- Change-based: Mover positions are captured when the position, velocity, or direction of movement changes concerning the previous position.
- Location-Based: Positions are captured when the moving object enters or is near a specific location.
- Event-based: Positions are stored when a specific event occurs or when certain activities are performed, such as posting a photo on Facebook or writing a Tweet.

- Combinations: There can be combinations of the primary approaches. For example, a GPS can capture locations at a given time and when the object's location changes significantly.

An examples of movement data is personal driving data. This data can be collected over time using a GPS device placed in a vehicle, resulting in a quasi-continuous dataset. Another example of movement data is datasets containing traffic information from ships and vessels using the Automatic Identification System (AIS)[2]. The AIS system allows vessels to communicate their position with other ships to avoid collisions. Movement data can also be obtained from a city's public transport. Even movement data can be obtained from people when they walk just by equipping them with GPS devices that track the places where they move. Another form of mobile data can be photo trajectories that store geographically referenced information, such as Flickr[3].

An example of a moving dataset is shown in Fig. 2.20. The dataset pertains to observing migratory movements of four white storks from August 20, 1998, to May 1, 1999. The data itself contains the locations of the birds on various dates. The referrers, in this case, are the time and the observed storks. The observed storks belong to the population referrer with four different values because there are four birds. In this case, when taking the referrers as time and storks, there is an attribute: the location in space. As can be seen, the location depends on time in this case. The location in space concerning time is continuous.

The structure of this dataset can be seen from another perspective as in Fig. 2.21. In this case, space and time are the referrers, and the attribute belongs to which stork is present in a given space and time.

2.11.1 Trajectory Data

As we have already seen in the previous section, there are several technologies to obtain data in motion, such as GPS. Also, it is possible to get the position of an object through technologies such as Radio Frequency Identification (RFID), smartphone sensors, and others. Therefore it is easier to obtain data sources that allow generating data trajectory.

Feng and Zhu [10] define a trajectory as a sequence of timestamped geographical locations. A trajectory is a sample of location points through which the moving object of interest has passed. Fig 2.22 shows an example of a continuous trace defined as $(p_0, t_0), (p_1, t_1), ..., (p_7, t_7)$ where p_i is a location and t_i is a timestamp.

As mentioned, movement is a change of position in space over time. Feng and Zhu [10] explain that a location can usually be expressed as a tuple of latitude and longitude. This type of change of position, using specific geographic points, generates a trajectory called a geographical trajectory or a raw trajectory.

There are also other types of trajectory, such as the Semantic Trajectory. In this type of trajectory, the locations are associated with semantic entities. The semantic entities can be meaningful places in the real world or descriptive texts that express the feeling and emotions of a person. It can be collected from various sources, basically from moving data. For instance, it can be from online data such as geotagged messages and content from

[2] https://www.atlantis-press.com/proceedings/jimet-15/25843780

[3] https://www.flickr.com/

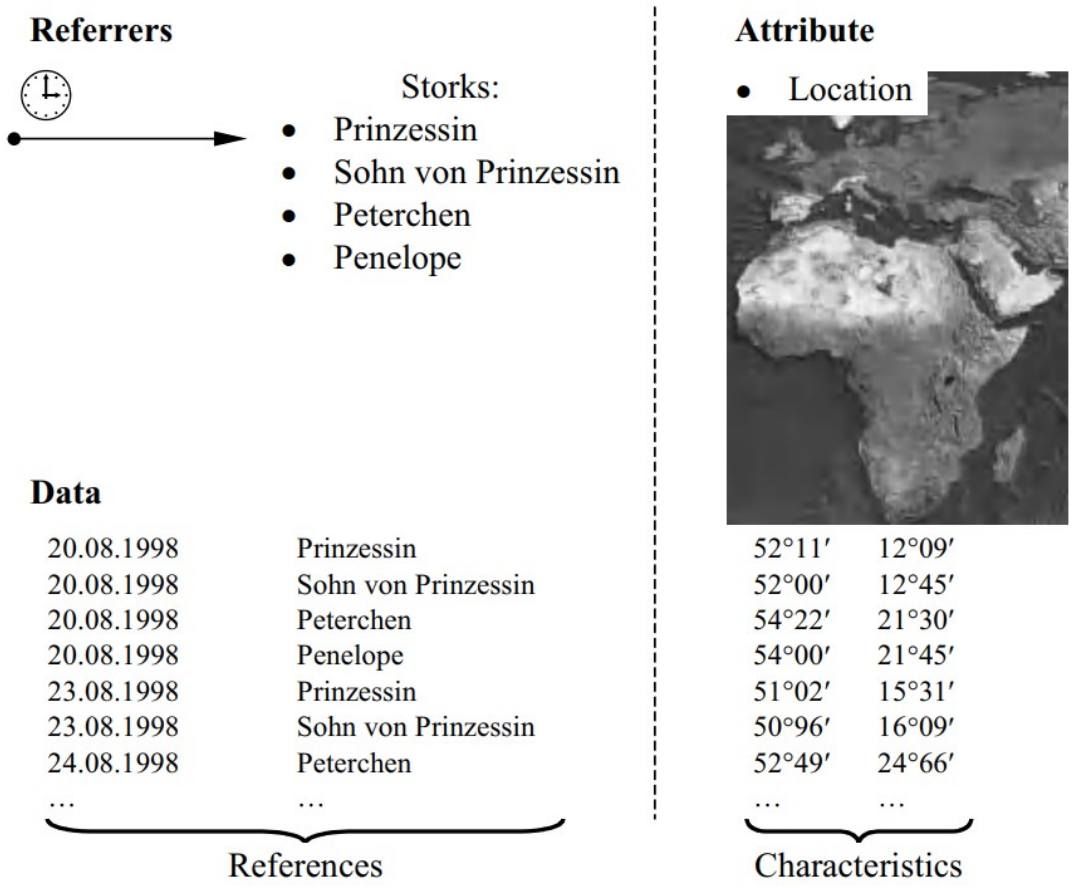


Figure 2.20: The structure of the dataset of white stork migration. Taken from Andrienko, N. and Andrienko, G. [9].

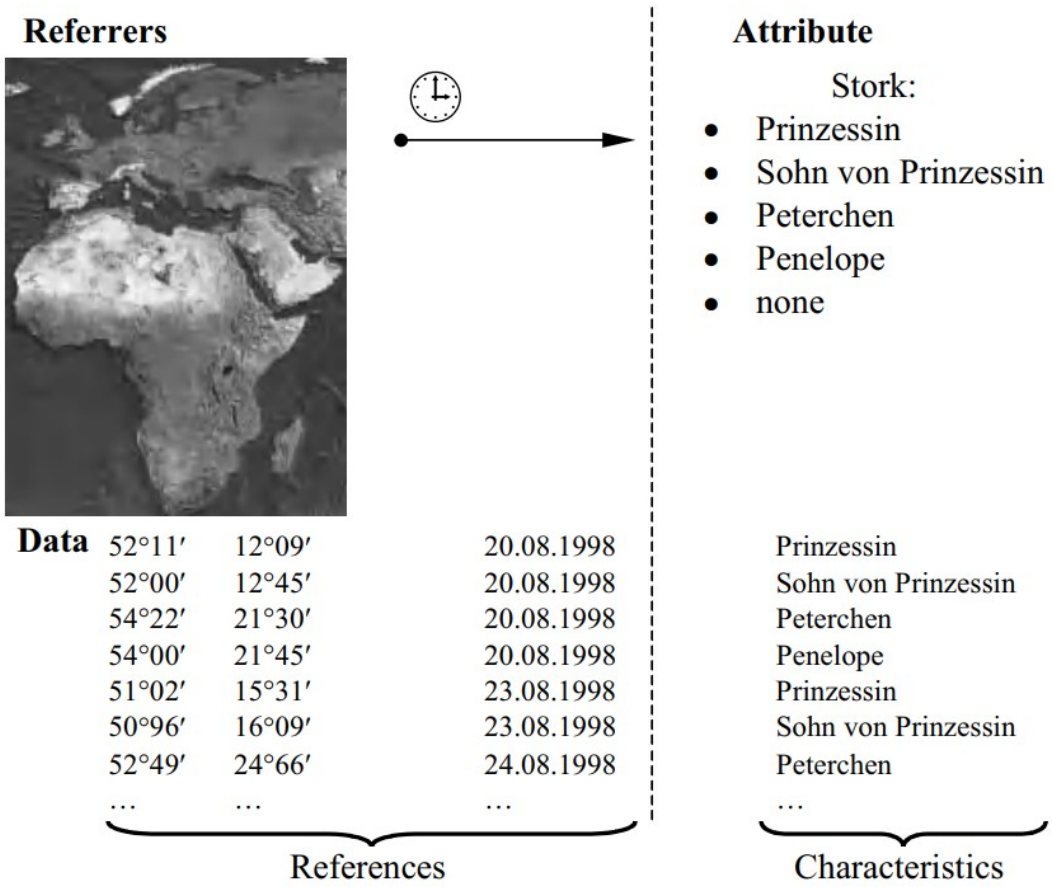


Figure 2.21: Another structure of the white stork migration dataset. Taken from Andrienko, N. and Andrienko, G. [9].

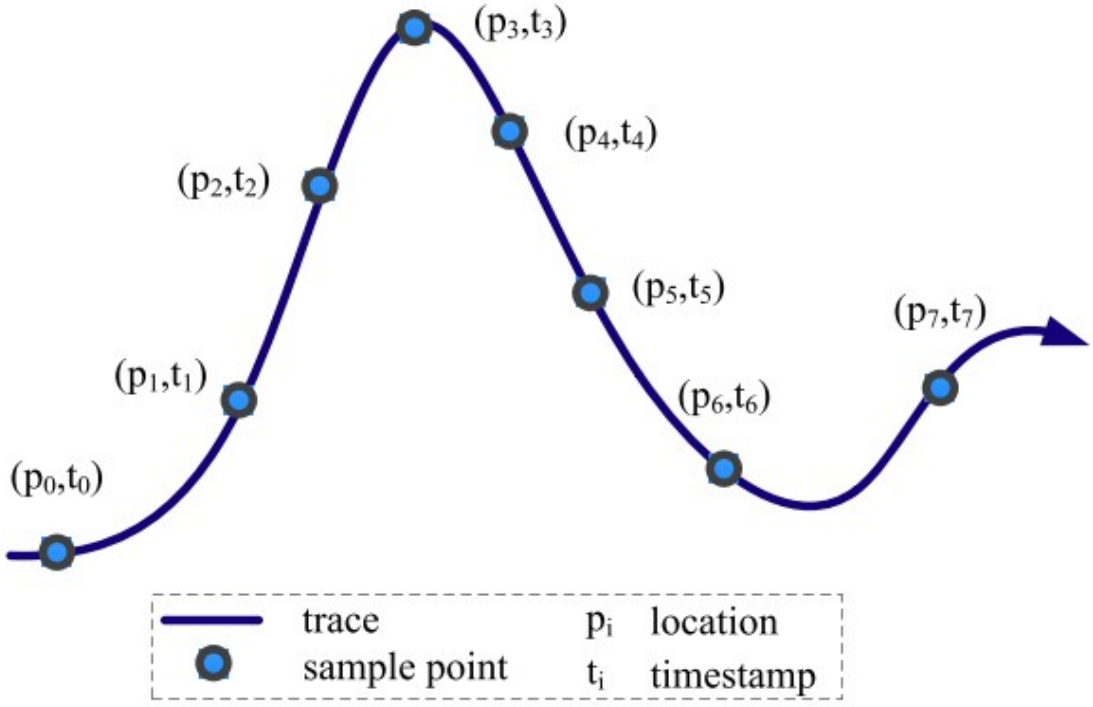


Figure 2.22: Trayectory fenerated by sampling a continuous trace. Taken from Feng and Zhu [10].

social networks such as Twitter. Creating these trajectories has various applications, such as path discovery, location and destination prediction, and movement behavior analysis. These applications can be widely used for urban mobility development, benefiting public and private organizations.

2.12 Dataset examples

We provide some examples of datasets taken from Andrienko and Andrienko [9].

2.12.1 Portuguese Census

This dataset corresponds to the 1981 and 1991 censuses carried out in Portugal. The data have attributes: population, number of men, number of women, number of people in different occupations (agriculture, industry, and more), and the number of unemployed

Space and time are referrers. Space is of the discrete type since Portugal is divided into administrative districts. The attribute values for the districts were aggregated. Spatially, the attributes are continuous since they are defined in all parts of Portugal. The attribute values of time are only two moments in time, 1981 and 1991. The attributes are temporally continuous since they always exist at any time, even if they are not always known. The structure of the census dataset can be seen in Fig. 2.23

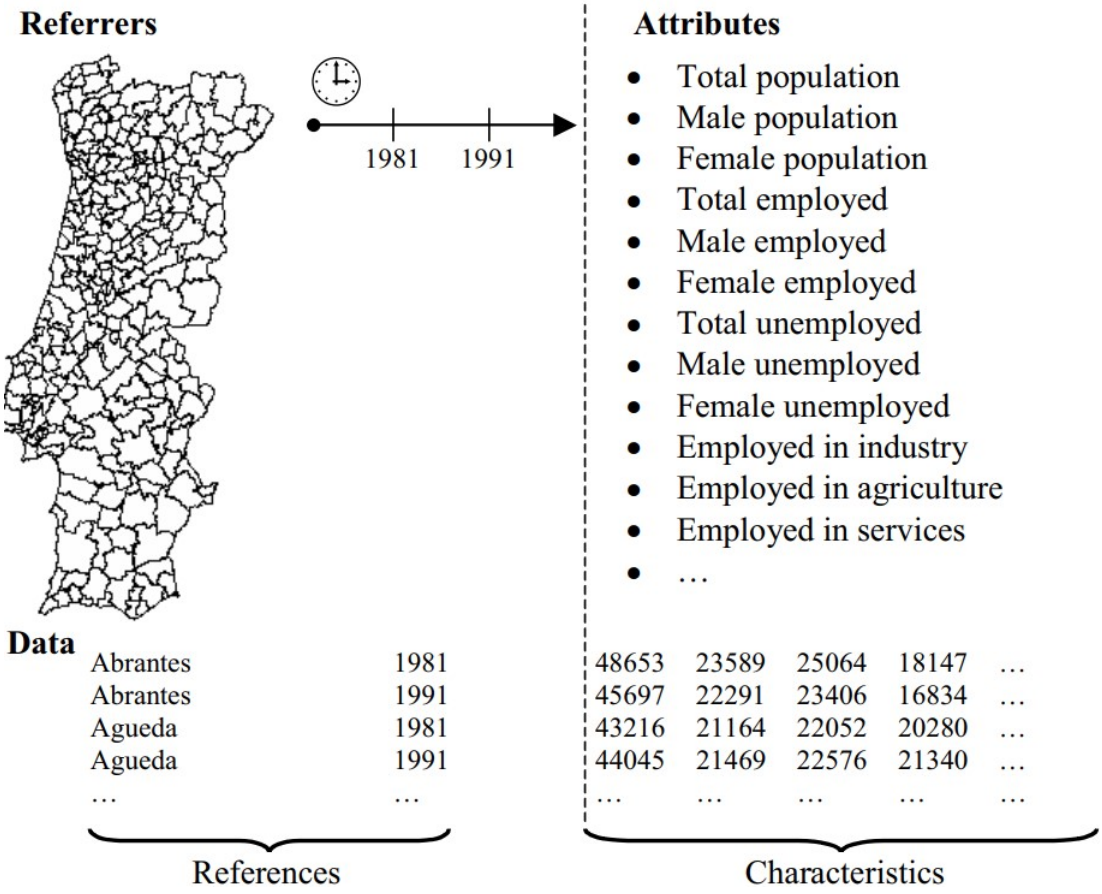


Figure 2.23: The structure of the Portuguese census dataset. Taken from Andrienko, N. and Andrienko, G. [9].

2.12.2 Crime in the USA

This dataset shows crimes data from 1960 to 2000. In this dataset, the referrers are space and time. The space is discrete since it pertains to the states that make up the USA, and the time is also discrete since the measurements are made every year. The attributes are the population, the total number of crimes of various types, and other related aspects. as seen in Fig. 2.24. The attribute values belong to each state and have been aggregated from individual instances. Similarly, the attribute values belonging to the yearly intervals are also defined by aggregation. Finally, all attributes in the dataset are spatially and temporally continuous; there is one value for each location and time.

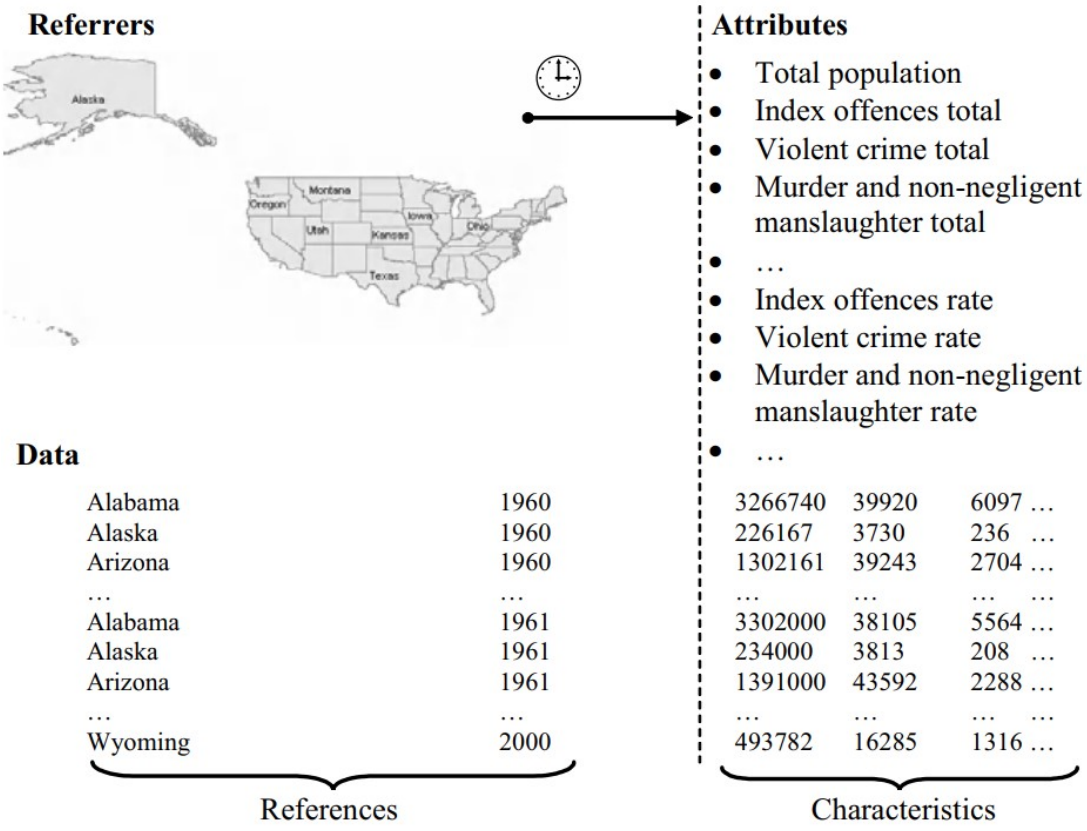


Figure 2.24: The structure of the USA crime dataset. Taken from Andrienko, N. and Andrienko, G. [9].

2.12.3 Twitter Datasets

Imran et al. [11] present a dataset called Two Billion Multilingual COVID-19 Tweets (TBCOV) in their work. TBCOV is a dataset born due to the need to analyze several situations generated during the COVID-19 pandemic. TBCOV is a dataset containing sentiment, entity, geographic, and gender tags, which can provide valuable information about public opinions, sentiments, and situations to help authorities understand various social phenomena during the pandemic. The dataset contains two billion multilingual tweets posted from 287 countries by 87 million users in 67 languages, all stored in various

tsv format files. Authors used the official Twitter API and a series of keywords and hashtags related to countries and COVID-19 to retrieve the Tweets. The collection period was from February 1, 2020, to March 31, 2021. Fig. 2.25 shows the distribution of the Tweets collected per week, and Fig. 2.26 shows the distribution of the languages of more than 10K tweets.

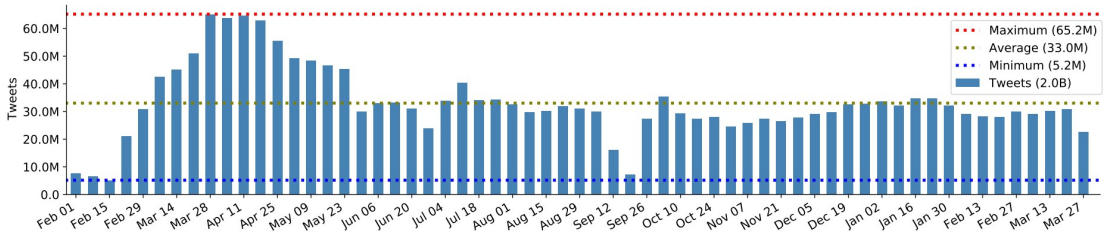


Figure 2.25: Weekly distribution of tweets collected from 1 February 2020 to 31 March 2021. Taken from Imran et al. [11].

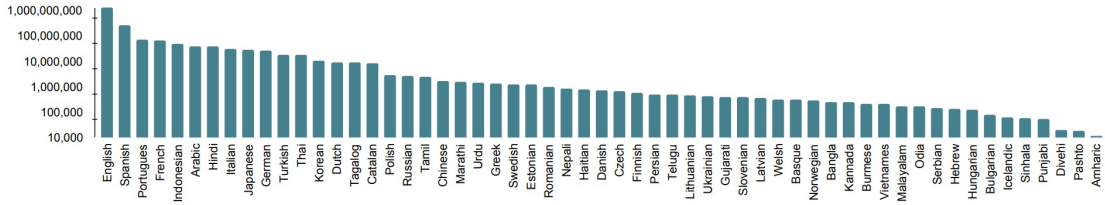


Figure 2.26: Language distribution with the y-axis indicating the number of tweets on a log scale. Taken from Imran et al. [11].

In this work, several machine learning models were used to provide the labels mentioned above to each tweet. The sentiment labels were obtained using the XLM-T model (Barbieri et al. [105]). Many Tweets do not have latitude and longitude, so the text of the Tweet itself and entities related to the Tweet was used to have a location. The paper proposes an approach to segregate the user's gender. The final dataset does not contain the text of each tweet. Instead, it has a list of entities from the text. Each tweet in the dataset is identified with an id, and some of the most recoverable fields are the date, the language, and the user identifier. If the tweet is a retweet, it will contain a retweet_id that points to the source tweet. Likewise, if it is a quoted tweet, it will have a quoted_id and an in_reply_to_id if it is a reply. Also, tweets contain gender, sentiment, and location fields.

The time and geographic location data can be useful to analyze the trajectory of the Tweets and also study the impacts caused by the COVID-19 pandemic and manage the consequences on people's social welfare. The work has found peculiarity when collecting Tweets is that only 1-2% of the Tweets collected have accurate geographical latitude and longitude. Therefore, the work uses techniques to infer the city or country where the Tweet was posted, but the exact location cannot be known. The trajectory can be traced exactly on a geographic map with the latitude and longitude. However, unfortunately, most users do not allow Twitter to access their location via GPS due to user privacy issues.

In the work of Villa-Cox et al. [106], a dataset for learning stance in Twitter conversations is presented. This dataset arises from the need to train and evaluate models for

extracting stances (denying and supporting opinions) from conversations within the field of opinion mining. Most of the existing datasets for training models are generally too small. These existing datasets also have the shortcoming that they do not have well-distributed classes and usually do not have a clear stay. Most datasets also have the shortcoming of not distinguishing between social network conversation types. In the case of Twitter datasets, they do not distinguish whether a tweet is a reply or a quote. This distinction is important because what happens in one type of event does not generalize what happens in the other type. In the end, in the work of Villa-Cox et al. [106], they determine that the two modalities behave differently in stance learning.

The dataset elaborated by Villa-Cox et al. [106] solves the shortcomings explained in the previous paragraph. The dataset is one of the longest and most hand-labeled datasets for detecting stances in conversations. The dataset contains more than 5,200 Twitter conversation stance tags for quotes/replies and is balanced in terms of stay type (support/denial). The dataset was obtained by collecting tweets through the Twitter API. Different types of hashtags were used to collect tweets, keywords, and dates to separate specific events (e.g., the Santa Fe shooting event in Texas in 2018).

The obtained dataset is stored in a JSON format with several parameters:

- event: Event to which the target-response pair corresponds to.
- response_id: Tweet ID of the response, which also served as the unique and eternally persistent identifier of the labeled database.
- target id: Tweet ID of the target.
- interaction_type: Type of Response: Reply or Quote.
- response_text: Text of the response tweet.
- target_text: Text of the target tweet.
- response_created_at: Timestamp of the creation of the response tweet.
- target_created_at: Timestamp of the creation of the target tweet.
- Stance: Annotated Stance of the response tweet. The annotated categories are: Explicit Support, Implicit Support, Comment, Implicit Denial, Explicit Denial and Queries.
- Times_Labeled: Number of times the target-response pair was annotated.

2.13 Summary

In this chapter, several concepts related to InfoVis were analyzed. The definitions of visualizations were addressed, and several fields parts of it, such as InfoVis, SciVis, and GeoVis. The characteristics of the various types of visualizations were discussed, and examples were analyzed for a better understanding of each field. A general pipeline for creating visualizations was discussed. The process to create Munzner's visualizations and Munzner's nested

model were studied, both very important since they generalize in a structured and helpful way all the steps and processes that must be taken into account to develop visualizations. Concepts about codifications and interactions that are very important for developing visualizations within the InfoVis field were also addressed. Concepts related to data, such as its properties, were discussed. We talked about data with temporal and spatial characteristics and data trajectory. Finally, we studied several dataset examples where it is noticeable that tweets generally do not have geospatial coordinates. All the concepts discussed above are essential and are the foundation for the development of visualizations. These concepts will be used to develop the visualization tool and fulfill the objectives of this work.

Chapter 3

State of the Art

This section will analyze state-of-the-art papers on information visualization related to movement visualization. The papers are organized according to the taxonomy presented by Schottler et al. [12]. First, some research with mapped geographic information will be analyzed, then with distorted and abstract visualizations. Later, some works related to Twitter will be analyzed. Finally, a summary is presented.

3.1 Trajectory Visualization

Visualization of trajectories can be set in two: geographic and abstract spaces (Fadloun et al. [107]). When a trajectory is set in a geographic area, it is usually a 2D map visualization. When a trajectory is represented in a conceptual space, it uses abstractions necessary to encode the data visually.

Geospatial networks are nodes and links associated with geographic locations (Schottler et al. [12]). These geospatial networks can be seen as trajectories in space, and examples of this type of network can be the social networks such as Facebook and Twitter. Schottler et al. [12] proposed a space for visualization of geospatial networks. They use four dimensions to describe techniques:

- Geography (GEO): How geographic information is displayed (explicitly or abstractly).
- Network (NET): How network information is displayed (explicit or abstract).
- Composition (COMP): As the two previous points, they are composed and integrated visually. For example, GEO and NET could be represented in a juxtaposed way, superimposed or nested.
- INTERACT: How interactivity is used in visualizations.

Within the GEO dimension, there are three subcategories (see Fig. 3.1). The first subcategory is called Mapped, the information visualization technique that explicitly has geographic representations. It must be taken into account that for these visualizations to be presented in 2D, there will always be some kind of distortion to represent the geographic

maps. Mapped also considers 3D representations, which have the most accurate geographic information but may still have certain distortions.

These types of visualizations have several advantages. They are suitable for displaying geographic information, i.e., Users can easily see which objects are far away or near if there are geographic features near the points of interest. Another advantage is that it shows geographic information about the topology. For example, users can see if the network is well connected, how far one node is from another, and how long the network links are. Another advantage is that people are already very familiar with maps, making it easier for end-users to understand and use them.

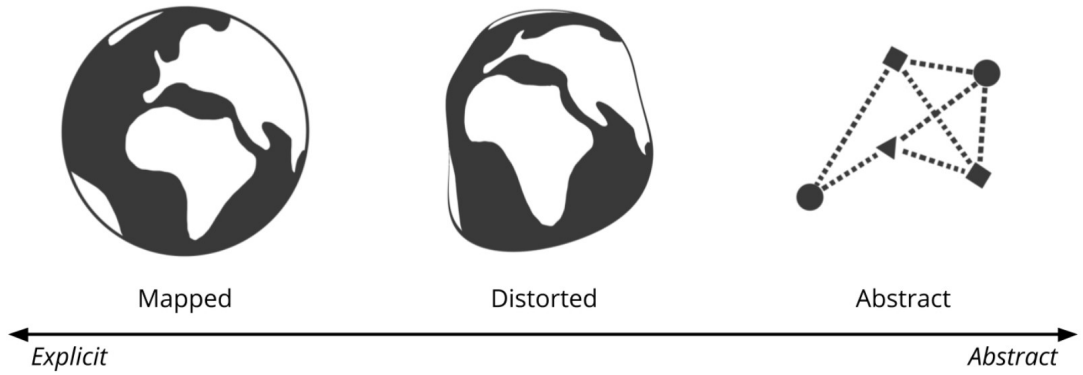


Figure 3.1: Geography representation (GEO). Taken from Schottler et al. [12].

There are also some drawbacks to Mapped visualizations. There may be visualization problems when nodes are very close or when there are long-distance links. Another disadvantage is that links that cover long distances can be misinterpreted and thought of more critically than shorter links. The last sentence can be true in some cases and not in others.

The second subcategory within the GEO dimension is Distorted. In this type of visualization, the geospatial positions are not exactly in their natural location, i.e., displaced. Unlike the last subcategory, here, there are distortions in the visualization to improve the understanding of the networks formed by the data. This type of visualization tries to avoid, in a certain way, the problems treated in mapped visualizations.

The last subcategory within the GEO dimension is the abstract. In this type of representation, projections on a map are no longer used. There is no longer a natural mapping between the position of an object on the screen and its geographical location. Most commonly, colors and shapes differentiate the things displayed and their locations.

Within the NET dimension, something similar to the GEO dimension is found. There are three subcategories, the explicit, the aggregate, and the abstract. The explicit subcategory displays nodes explicitly as points in the visualization, whether on a map or not. Communicating the nodes in groups as metanoids is an aggregation. Finally, not showing the node representations as points in the visualization is the abstract subcategory. An example of the latter can be hexagonal pattern maps. Similarly, within the network dimension, the same is valid for links.

The compositional dimension deals with how GEO and Net dimensions are integrated into a single visualization. Within this dimension, we have juxtaposed visualizations, where the network representations are placed next to the geographic graphics to show

complementary information. Within this dimension are the superimposed visualizations, which show the network and geographic pictures on top of each other. Then, there are the nested ones, where either the geographic or network visualization is nested within a visual element that belongs to the other type of representation. Finally, we have the integrated one where the geographic and network visualization is shown together because they cannot be shown independently to give sense to the visualization.

Lastly, there is the interaction dimension. Interaction has to do with how the visualization elements behave around an action performed by the user. In some instances, this dimension may be optional. An example of interaction may be the zoom that can be performed on the visualization or moving elements as desired.

3.1.1 Mapped

One way to visualize connections between different spatial locations is by using Origin-destination flow maps. Origin-destination (OD) flows are often used when the routes between origin and destination are not of interest or are unknown. A common problem with OD flows is that even if there is a small number of connections, many overlaps will not allow users to visualize and understand the connections in the visualization correctly, as for example in Fig. 3.2.

There are several ways to reduce visual clutter, such as matrix, clustering, grid-based approaches or glyph-based visualizations, and edge bundling. In much of the early research on visual clutter, flow direction has yet to be considered as in Selassie et al. [108]. The direction of the links is essential because there may be links between the exact locations but in different directions. The latter produces a great deal of link clutter. However, in many techniques to reduce visual clutter, the locations of these links overlap, and appropriate measures still need to be taken.

In the work of Graser et al. [13], the problem of visual clutter is addressed by considering the directions of the links. Four innovations are proposed: i) a new clustering technique for OD flows, ii) The use of an edge building approach using matrix computations, iii) A new technique to determine the local strength of a bundle, and iv) Geographic information system-based technique to spatially compensate for beams describing a different flow direction. The authors' contributions are based on applying the concepts of real-time implementation of forced edge building (FDEB) proposed by Zielasko et al. [109], but in a GIS framework approach[1] and taking into account the direction of the links:

1. The edge clustering technique uses the k-means algorithm to determine the number of clusters. The objective of clustering the links is to reduce the computation time in the last steps of edge bundling. The clusters must divide the links so that the groups formed are compatible with each other, and one way to achieve this is by k-means. Additionally, using this algorithm, the edges are evenly distributed. Graser et al. [13], apply a heuristic approach to determine the optimal number of clusters. Fig. 3.3 is an example of clustering.

2. Use matrix computations analogies to speed up the FDEB calculations. Limiting the copy and loop operations required. FDEB calculations are used to bundle the edges.

[1] `https://www.qgis.org/es/site/about/index.html`

Figure 3.2: Point-based OD dataset that consists of 15812 edges representing flight connections between European airports. Taken from Graser et al. [13].

This process by Holten and Van Wijk [110] is described in the following. FDEB uses specific points called control points of an edge and uses elastic and electrostatic forces to move the control points closer together. These forces are calculated over several cycles, and in each cycle, additional control points are added to the edges. The elastic forces are calculated between control points on the same edge, and the electrostatic forces are calculated between points on different edges. This technique for grouping edges should only affect compatible ones, so a compatibility measure is also defined that takes into account the angle of two edges, the length of the edges, and the position, which ensures that edges that are too far apart are not grouped. Finally, it must be ensured that parallel edges with similar sizes and positions are not grouped if they are displaced (as a parallelogram). The control points are arranged in a 2D matrix. This matrix is called a control point matrix, where the rows represent the edges and the columns represent the control points on the edges. This way, the elastic forces are calculated along the rows, and the electrostatic forces along the columns. In the Graser et al. [13] proposal, the control points always have a fixed position in the edge-point matrix, thus avoiding copying operations at each cycle and only updating operations.

3. A local bundle strength is added to the local segment attributes to activate the rendering of the resulting bundles. The previous steps modified the structure of the edges with the control points, but how to know the local strength of a bundle to create a flow map. The edges generated in the edge bundling are separated into segments that are reacting lines between consecutive control points. Then, all other segments that begin and end at approximately the same place are identified for each segment. The segments whose beginning and end are between half the distance of the studied segment are selected. Finally, the strength of the identified segments is summed.

During edge grouping, compatible edges are bundled together regardless of directionality. During the display phase, Graser et al. [13] apply a positive offset to each line drawn. This offset shifts each line drawn to the right concerning the flow line's direction. This ensures that opposite flows within a bundle do not overlap.

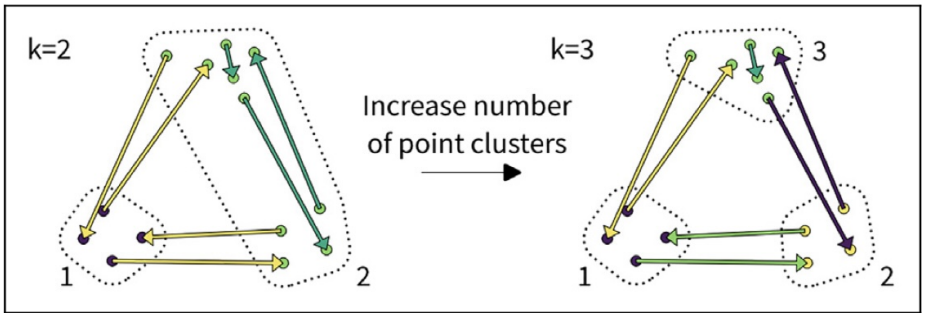


Figure 3.3: Clustering example. The number of edge clusters increases as k increases from 2 to 4 edge clusters. Taken from Graser et al. [13].

A gradient is used to indicate the direction of the flows. The color gradient used is Viridis and indicates the direction from dark to light color as in Fig. 3.4. In the work of

Graser et al. [13], it is explained that such a gradient is chosen as it is uniform and color blindness resistant. Finally, the bundles are graded by their weight. Stronger bundles are drawn on top of weaker ones to reduce visual clutter and make strong connections easier to see.

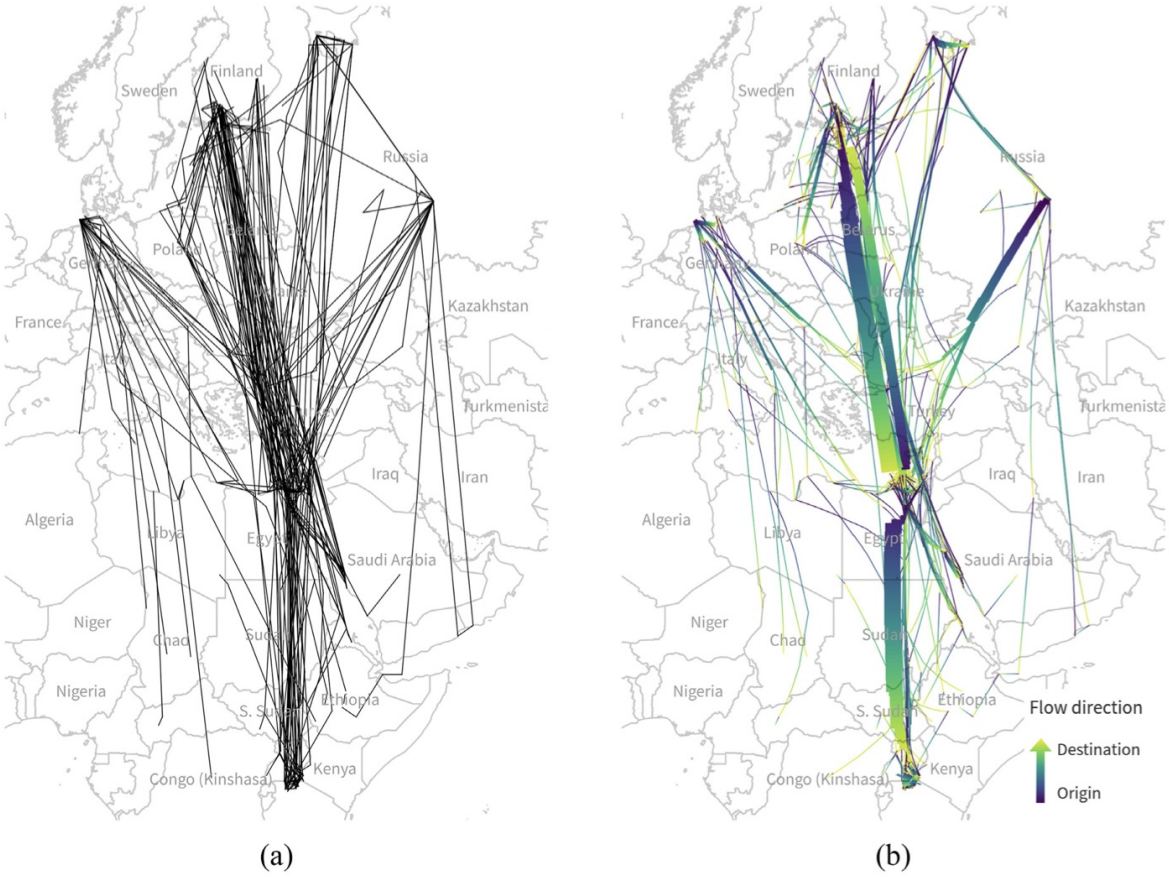


Figure 3.4: Migration of gulls. a) Original GPS trajectories converted to OD flows. b) Migration edges grouped with a dark to light gradient indicating the direction of migration (OD flow map with edge bundling). Taken from Graser et al. [13].

Fig. 3.4 compares a flow map (a) with the flow map obtained using the edge bundling approach (b). This Figure shows the migration of Gulls from a dataset containing point-based flows and containing information from 2009 to 2015. In (a) of Fig. 3.4, each edge represents one bird's movement. A threshold of 100 km was taken to differentiate between local and migratory travel. In image (b) of Fig. 3.4, birds migrate from the North to the Nile delta and Lake Victoria. It is also determined that there is an opposite flow back north from the Nile delta.

The above article presents a series of interesting proposals to handle the clutter of OD streams using edge bundling. A new clustering technique is proposed that speeds up the computations. Also, the overlap is avoided by using offsets during rendering. In general, the proposals meet the requirements of the flow maps. They present an overview of the current situation. The spatial configuration of the study area is preserved. With the proposed technique, the flow origin and destination are distinguished in most cases.

Nevertheless, due to edge bundling, the visibility of the individual OD locations is reduced. Also, the flow direction can be appreciated thanks to the color gradient. However, it is possible to use another type of gradient that reduces occlusion by using transparent colors.

With edge bundling, the occlusion is reduced, but it only solves some occlusion cases. For them, other techniques could be used, as in Ersoy et al. [111], which also has disadvantages. The important point is that the procedure's parameters must be adjusted to the dataset of interest and set correctly to avoid lousy clustering. With the proposed edge bundling, it is impossible to determine which critical characteristics of the dataset determine whether it can be efficiently clustered. The ideal parameters for the data must be found by trial and error, which makes the task time-consuming. Finally, as we have seen, the strongest bundles are placed at the front, resulting in different bundles overlapping each other's locations. This limits readability and can obscure patterns from the visualization.

There is another primary approach besides flow maps to visualize data flows between different geographical locations. This other approach is through the OD map. The OD map can be seen as an improvement to the OD matrix first seen in Voorhees [112]. The OD matrix (see Fig. 3.5) locates the origins and destinations of the flow in rows and columns, and the flow is displayed within each cell. Additionally, the OD matrix can be compatible with heatmap to encode the size, as seen in Wilkinson and Friendly [113]. The OD map divides the canvas into grids based on actual geographic locations. Each cell contains a smaller version of the overall map, and each cell's color indicates the flow's magnitude. Matrix ODs have a fundamental disadvantage. They do not have a spatial location mapping of the origins and destinations as a geographic information map would allow. This drawback of the OD matrix is what OD maps try to overcome. OD maps (see Fig. 3.6) use nested mosaic maps to provide geographic information.

SUNDAY

Origin \ Dest		Brisbane East		Brisbane North				Brisbane South						Brisbane West				Brisbane Inner			
		30101	30103	30201	30202	30203	30204	30301	30302	30303	30304	30305	30306	30401	30402	30403	30404	30501	30502	30503	30504
Brisbane East	30101	841	1515	7	10	19	15	122	31	42	14	6	4	1	21	6	2	45	34	77	10
	30103	1495	11690	22	36	88	94	856	211	232	63	65	71	12	201	59	25	340	508	253	90
Brisbane North	30201	10	41	3765	1014	290	1509	13	39	15	12	11	12	14	1	120	214	105	12	199	75
	30202	9	37	989	6576	403	454	21	110	46	36	11	13	13	5	115	305	278	24	1040	157
	30203	19	120	297	453	4042	918	56	112	47	57	41	15	38	3	135	47	353	34	862	121
	30204	25	248	1393	468	892	10043	72	66	57	35	25	20	10	5	72	56	140	20	327	60
Brisbane South	30301	128	1086	15	32	50	63	7150	1068	837	177	58	93	16	289	82	45	553	578	326	133
	30302	27	191	46	163	163	79	1045	11067	1172	1122	410	194	86	26	473	218	2254	498	1505	502
	30303	34	221	11	33	56	53	811	1175	13909	807	291	1071	44	25	156	75	666	114	411	193
	30304	7	65	6	36	76	35	178	1188	815	5392	504	1090	54	15	193	34	651	122	470	109
	30305	5	67	8	24	43	18	60	437	294	529	7403	1289	102	7	263	24	266	37	157	53
	30306	4	66	6	14	36	24	78	187	1013	1085	1277	6534	31	4	75	21	355	34	210	68
Brisbane West	30401	2	11	9	13	25	10	14	84	39	53	103	28	4484	23	395	35	118	16	150	130
	30402	9	108	2	2	5	2	168	25	23	5	9	6	38	657	19	7	37	15	18	14
	30403	7	49	90	120	168	80	85	422	144	200	246	62	330	42	7752	251	768	96	1177	1121
	30404	4	31	190	323	45	78	50	160	56	26	27	19	25	5	223	4647	411	36	701	609
Brisbane Inner	30501	71	363	87	322	361	175	606	2262	705	669	261	335	122	49	901	436	24048	635	3507	1297
	30502	39	503	11	32	37	22	618	523	112	161	37	36	16	22	113	47	694	2899	375	146
	30503	82	336	190	982	884	381	347	1385	427	458	157	226	150	31	1178	694	3481	407	18112	1190
	30504	16	64	66	131	120	60	151	417	198	99	35	76	121	15	1201	624	1296	141	1191	7060

Lowest value 50th Percentile Highest value

Figure 3.5: OD matrix showing flows on a specific day. Taken from Krishna et al. [14].

In the work of Yang et al. [16], MapTrix is proposed, which tries to unite the advantages of matrix ODs without losing the visualization of geographical locations as in flow maps. The MapTrix design is shown in Fig. 3.7. The MapTrix has three components: the source map, the destination map, and an OD matrix with a single guideline connecting each source and destination to the corresponding row or column in the OD matrix. The advantage of guidelines is that they eliminate the clutter in flow maps. These guidelines are made using a non-overlapping algorithm based on a one-sided mimic labeling model of Bekos et al. [114]. Additionally, the guidelines are separated, so they are not difficult to read.

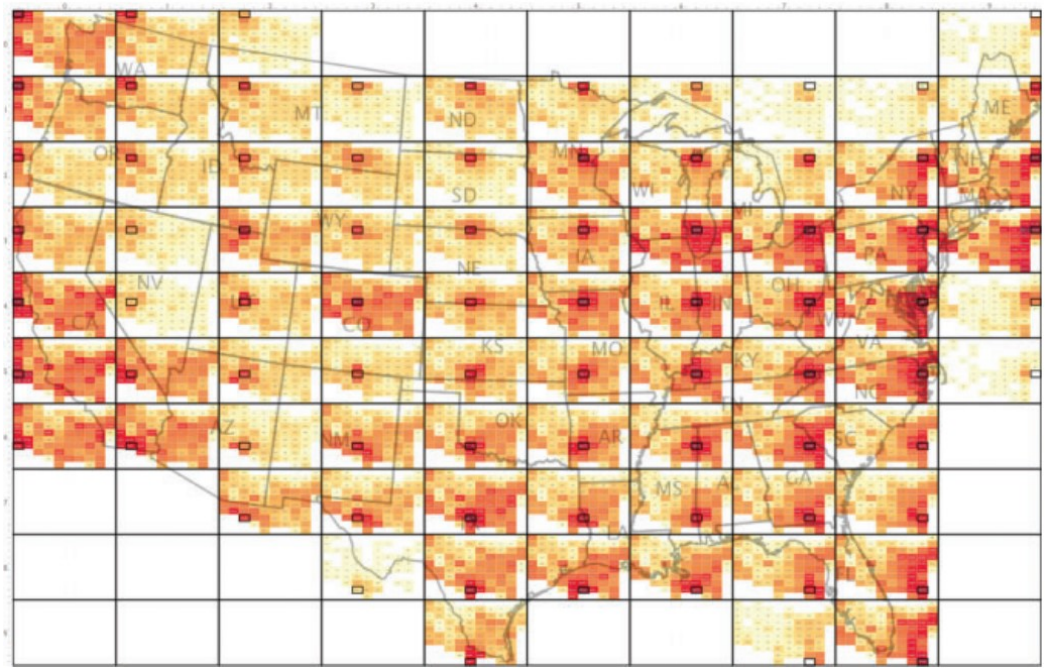


Figure 3.6: Large grid cells represent origin locations of US country-country migration vectors. Density maps of their destinations are drawn within them. Taken from Wood et al. [15].

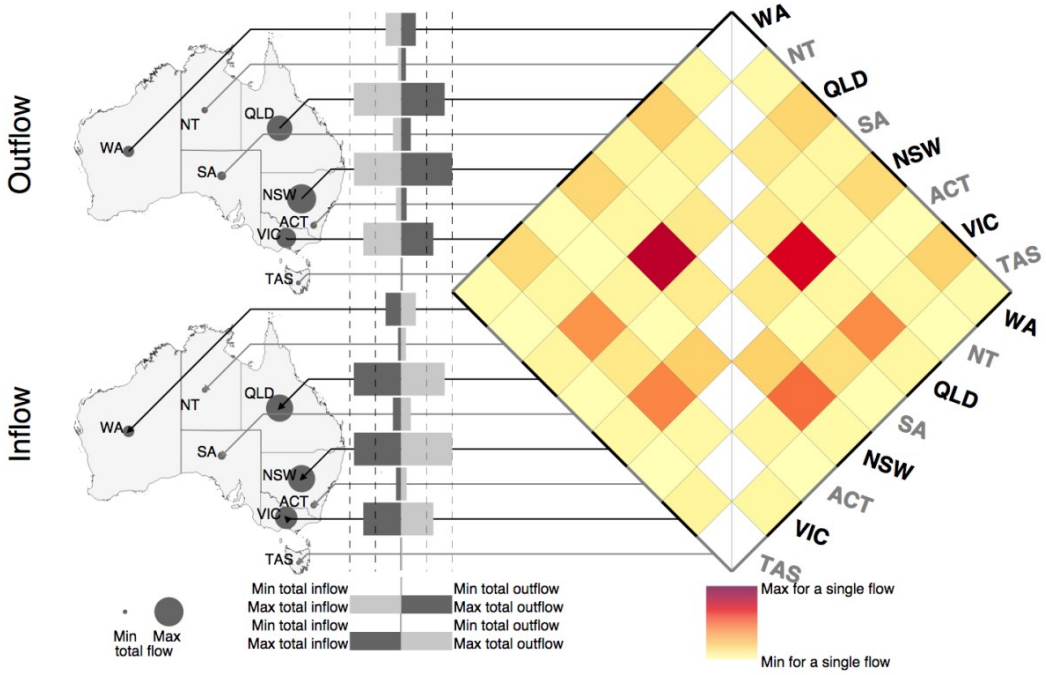


Figure 3.7: MapTrix example. Synthetic dataset from Australia about internal migration. Taken from Yang et al. [16].

Yang et al. [16] conducted two studies showing the benefits of MapTrix and comparing it with OD flows and OD maps. From the study conducted by the authors, it is highlighted that MapTrix is the preferred one in terms of user preferences. A similar performance between MapTrix and OD maps is found. Both outperformed the bundled flow map presented in Wood et al. [15]. When dense datasets with up to 51 origins and destinations are used, comparing flows between or within regions becomes very complicated. This happened both for MapTrix and OD maps. Then, thanks to the authors' studies, it was determined that in the future, more interaction could be added in MapTrix with the use of filtering and region zooming. Finally, it can be added that this work can clearly show flows, but it does not consider time, which is an important issue. Next, we will talk about a work that considers the time dimension.

In the work of Boyandin et al. [17], an approach for exploring temporal Origin-Destination Data called Flowstrates is proposed. Due to the emergence of many origin-destination datasets, many techniques have been developed to visualize these types of datasets. However, many of them do not address the problem of temporality, which is why the work of Boyanding et al. [17] is essential. In short, what is proposed by Boyandin et al. is to develop a new visualization approach in which the origins and destinations of the flows are shown in two separate maps, and the changes over time in the flow magnitudes are shown in a separate heat map in the middle.

Many data can be found in the form of Origin-Destination, and to analyze and visualize this type of data, one of the most common techniques is flow maps, as discussed above. These flow maps help answer many questions about data spatiality, such as which are the largest or smallest flows, where is the location of the origins and destinations, and what is the direction of the flows. Nevertheless, flow maps are not designed to answer questions about temporal and spatial dimensions and their relationships.

In the work of Boyanding et al. [17], Flowstrates have proposed to try to answer questions related to spatial and Spatio-temporal tasks. Flowstrates can be seen as a hybrid solution allowing geographic and temporal representations in one. Often it is not necessary to see exact flow paths, and it is usually unknown in origin-destination datasets, so it would not be inconvenient to see the origins and destinations of flows on two separate maps. The time information is abstracted through a heatmap, where the columns correspond to different time periods. Flow lines connect origins and destinations through corresponding rows in the heatmap.

Many design considerations were made in the work of Boyanding et al. [17]. Maps were used to represent the origins and destinations as they allow for efficiently answering questions about geographic locations. The reason for using two independent maps is that the flow directions are visible, any appropriate representation of the temporal data (heatmaps) can be used, and different regions can be focused on both the source and the destination. Links are used for connecting the origins to the destinations as they clearly show the origins of hundreds of flows in the heatmap. The heatmap was chosen to easily represent temporal changes in flow magnitudes at different zoom levels. Moreover, using the same color scheme in the heatmap, source, and destination maps makes it possible to compare the totals in the geographic maps with the individual values in the heatmap.

Fig. 3.8 shows refugee flows between East Africa and Western Europe. Sudan is highlighted in yellow. The heat map shows the magnitudes of the flows per year. The lines show the origins of the flows, the destinations, and the changes in the magnitudes over

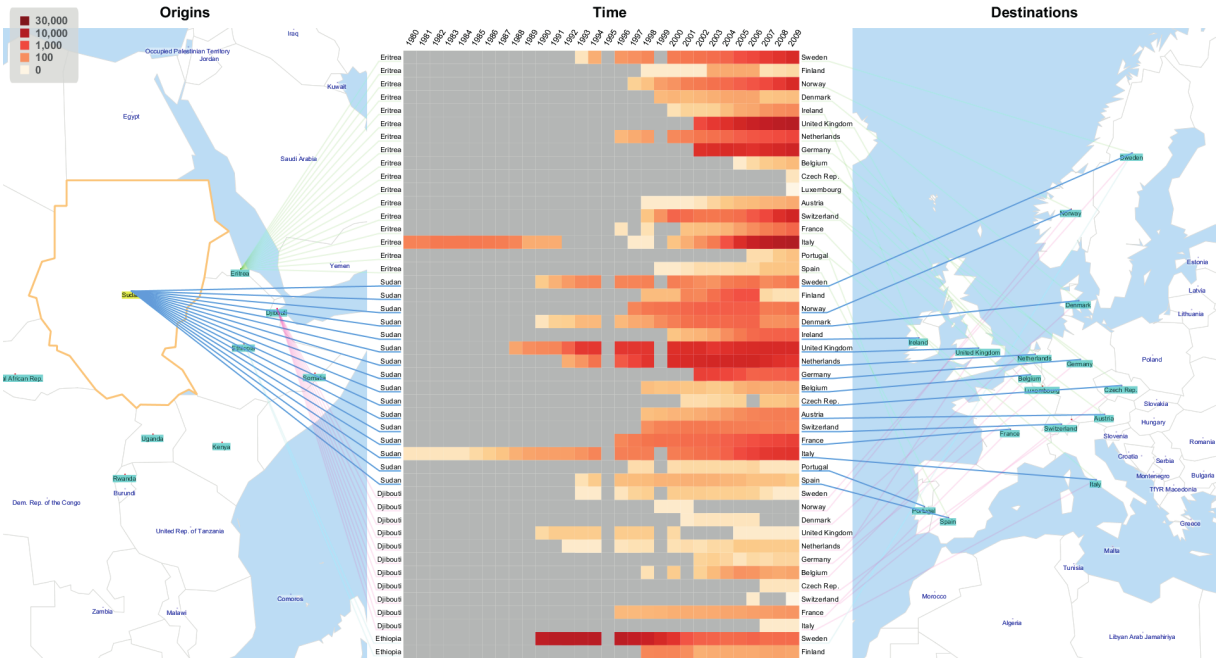


Figure 3.8: Flowstrates showing flows of refugees between East Africa and Western Europe. Taken from Boyandin et al. [17].

time. From this figure, it is possible to find distinctive patterns (shown by the intensity of color and the years in columns), such as high refugee flows between Sudan and the UK. Boyanding et al. [17] allow some interactions, such as selecting locations on the source and destination maps by name or a lasso tool. Zooming and panning are also allowed. Reordering the heatmap according to geographic positions or maximum flow magnitudes is allowed. The heatmap also allows row aggregations, which can be grouped according to their origin or destination.

In summary, Flowstrates fulfill its objective of presenting a technique for the visualization and exploration of origin-destination data and temporal characteristics. The Flowstrates proposal presents a series of advantages. Having maps allow knowing the locations of the flows in their origin or destination and relates functionally temporal and spatial dimension in a visualization that consists of three parts. Despite all the advantages, there are still certain limitations of the proposal. The first one is produced by the links that join the maps with the heatmap. A large number of lines produces clutter and visual overlapping. There is a way to try to solve this, and it is employing heatmap reordering. The heatmap rows can be reordered according to the origin, but there will be a disorder in the destination lines. The same will happen if the heatmap rows are ordered according to the destination. The second limitation is that it is impossible to see the flows on a single map. Then, the orientation of the flows is not realistic, and the distances between the origins and destinations can not be estimated or compared to the length of the flow lines (this is possible in traditional flow maps).

3.1.2 Distorted

For the visualization of networks, it is possible to use distortions based on the network data. The distortions can be the product of an algorithm or interaction technique. In the following, we will analyze some works that have developed visualizations with some distortions.

Deformation maps have several characteristics. Map distortions display quantitative data and attempt to preserve the adjacency and shapes of geographic regions while modifying the area (see Fig. 3.9). One of the most common ways to apply distortions to maps is through contiguous area cartograms (see Gastner and Newman [115], Sun [116], Ahmed and Miller [117], Shimizu and Inoue [118], Böttger et al. [119], Lin et al. [120], Jenny [121]), but some of them have certain limitations. One limitation is that the proportions between two areas cannot be guaranteed to precisely match the proportions of their data quantities. There must also be limits to the deformation since, with a highly deformed map, one cannot easily recognize the original geography. Another limitation is that old area cartograms present single scalar values. However, it is possible to use the distances of multiple locations and compare them with data attributes simultaneously, as seen below.

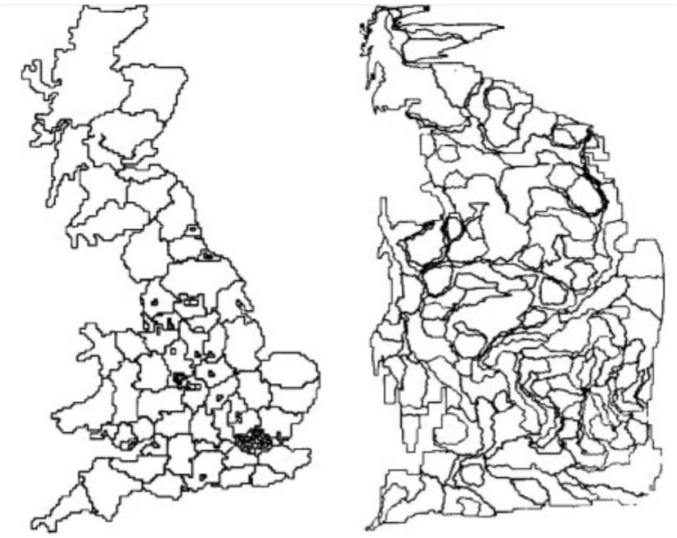


Figure 3.9: The original map and distorted map showing the population of Britain by country. Taken from Gastner and Newman [18].

In the work of Bouts et al. [19], a map deformation technique is presented for application to dissimilarity data visualization that preserves topology and balances geophysical shape preservation with data transmission. Bouts et al. [19] build upon and extend the concepts of Multidimensional Scaling (MDS). MDS provides a multidimensional scale of data points on a mesh that is then deformed and mapped to the map to be transformed. The mesh deformation is subject to certain constraints so that the mesh and data vertices obtained with MDS do not pass through mesh edges and do not cause excessive deformations, using an algorithm developed by the authors of [19]. In order to preserve the mesh topology, the Karush-Kuhn-Tucker [122] conditions must be satisfied. An example of what MDS does, but only one variable can be seen in Fig. 3.10. In the map are marked the capital cities of Australia and in brackets is the average percentage increase in house prices in 2013.

MDS is used to analyze the similarity of these values. MDS result is shown on the right side of the figure. Sydney and Canberra are farther away than Melbourne and Darwin. The positions are explained by calculating the difference between Melbourne and Darwin, $6.8 - 6.0 = 0.8$, while the difference between Sydney and Canberra is 10.8.

After deforming the map with the use of MDS and the use of a mesh and with the use of a topology preservation algorithm developed by Bouts et al. [19], we arrive at a map as seen in Fig. 3.11c. Fig. 3.11 shows the central locations in Great Britain and also the railway network. The map is deformed according to the travel time between each location. In Fig. 3.11b, only MDS is used, and it can be seen that the map does not preserve the topology. In contrast, Fig. 3.11c shows that the map is deformed as a data function, and the topological characteristics are preserved. It can be seen that the cities Thurso, Pembroke and Penzance are displaced outwards, that is, they do not have good train=travel time services, unlike Liverpool which has no significant drawbacks. Finally, the black lines shown in Fig. 3.11b, d help to perceive the deformations in a better way. The lines make it possible to see where the points were before (Fig. 3.11a). These simple lines are vectors showing the displacement of the previous state concerning the current position of the points.

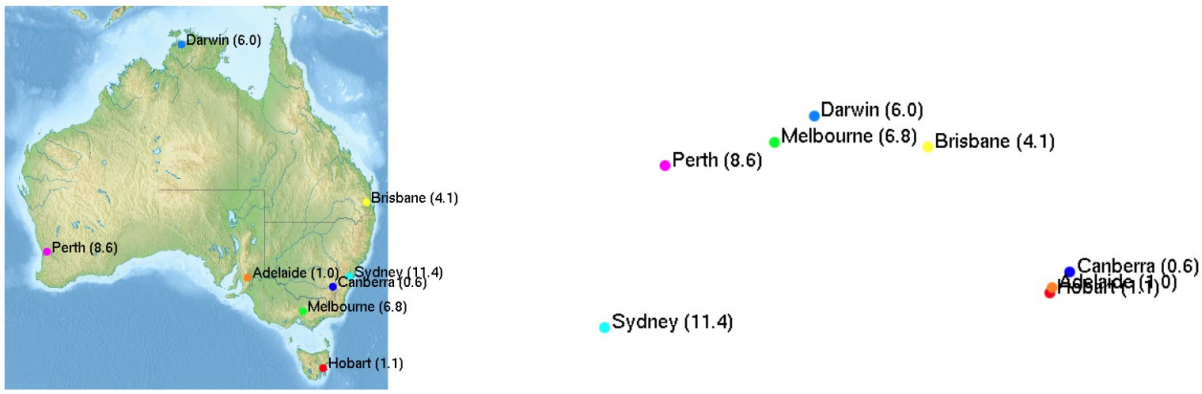


Figure 3.10: Univariate example of MDS to visualize dissimilarities between geographic locations.Taken from Bouts et al. [19].

One of the essential features of the work of Bouts et al. [19] is that it can use multidimensional data for map deformation. Also, the paper shows that the implemented algorithm is quite fast for deforming a geographic map and, therefore, could be compatible with real-time deformations. A limitation to work presented above is that when there is a large number of data points, it will be challenging for the algorithm to scale those values. Also, when the dissimilarities between points vary greatly, there may be occasions when how similar points can be approximated would only be with a complex folding of the map that would compromise the fidelity of the data and the geographic fidelity.

Another way to visualize networks with geographic components and distortions is using geo-located graphs on maps. Usually, many objects in the world have some geographic location, and those objects can form networks, for example, transporting materials between several factories. These networks form graphs, and by using a map, they can be located according to their geographical position. In general, geo-located graphs on maps work well for small networks but fail for networks connected by many points in small dense areas that

Figure 3.11: Distorted Map, based on the length of time it takes to travel by rail between 35 locations in Great Britain. a) Map of the Stations and Rail Network, b) Deformation Map using MDS, c) Deformation map using MDS and the topology preserving algorithm. Taken from Bouts et al. [19].

are distributed over much larger areas. One approach to visualizing geo-located graphs on maps is through large-scale visualization that shows the entire graph but loses detail.

Brodkorb et al. [20] present a technique for visualizing geo-located graphs on maps that allows to visualize the whole of a graph and simultaneously visualize details of certain parts of the graph. The technique presented by Brodkorb et al. [20] involves using inserts to visualize details of small areas of the maps. A local area reorganization is performed to prevent the graphs of the inserts from overlapping and crossing edges. Additionally, insets are created or hidden according to the user's navigation. These inserts are placed to avoid overlap and are located close to their original position on the map. Insets are enlargements of parts of the map, which use a local distortion based on graphs. This minimizes overplotting and edge crossing in areas where there are a large number of nodes. Finally, the authors of the paper add interaction in their graphs. Nodes can be expanded or collapsed. While exploring the graph, the insertions are automatically shown or hidden depending on whether the user is in a zoomed out view, zoomed in, or scrolled.

A view produced by Brodkorb et al. [20] can be seen in Fig. 3.13. This visualization shows a part of the global Internet network. The network has 39 nodes and 62 edges. Some nodes are located in America, East Asia, and the most prominent part in Europe. In general, it is possible to see that the overall geographical structure of the graph is maintained, and it is easy to see the details of particular locations and the connections between nodes. Without the need to zoom in, the general structure and details of the densest part can be seen. On the other hand, in Fig. 3.12, the network cannot be seen correctly due to the high density of nodes in the European region.

The insets also allow for local distortion to improve the readability and understanding of the network. Fig. 3.14 shows the result of applying the Kamada-Kawai [123] algorithm.

The structure of the graph can be seen in a better way. Its structure can be seen more clearly. It is clear that there is a direct connection between England and America, and France, Switzerland, and Italy are directly connected to the other two countries. By zooming in on the Europe Region, the view gets an image as in Fig. 3.15. The Europe region is visible, the network structure is visible, and the visualization allows sight of the nodes in the more distant regions such as the United States.

Figure 3.12: Simple geo-located graph on maps. Taken from Brodkorb et al. [20].

Figure 3.13: Geo-located graph on maps with a inset showing more details in a specific region. Taken from Brodkorb et al. [20].

As can be seen, the advantages of the visualization technique presented by Brodkorb et al. [20] are related to the use of insets that allow viewing regions of the map without the need to zoom and maintain a global view of the network. However, there are aspects where

visualizations could be improved—for example, adding search elements to find a node of interest to the user. Another aspect that could be addressed is when visualization of the places visited by several people is made. It would not be possible to visualize temporal aspects. It would not be possible to know how often a place is visited, how long a person stays in a place, and finally, it would not be possible to differentiate between people.

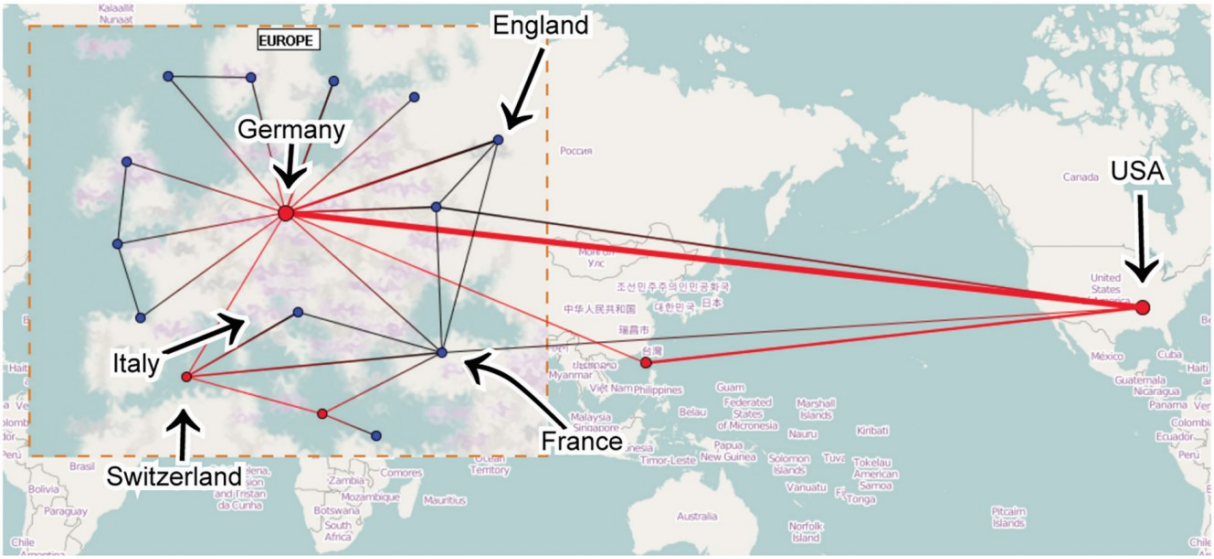


Figure 3.14: Inset with local distortion for showing details of the graph structure. Taken from Brodkorb et al. [20].

3.1.3 Abstract

The research and analysis of social networks have been increasing, and their importance is becoming more noticeable (see Tabassum et al. [124], Enos and Nilchiani [125], Kolli and Khajeheian [126]). One of the most commonly used ways for network visualization systems is using node-link representation. The work of Heer and Boyd [21] presents the design and implementation of a system for the visualization and exploration of large-scale social networks. The design, based on node-link representation, is shown in Fig. 3.16. it is feasible to visualize community structures and user data quickly. What is shown in Fig. 3.16 is a network formed by the network users for networking and dating. Each node represents a user, and the links represent their connections with other users.

The node-link based representation is good for showing a sparse network. However, in general, social networks are globally sparse and locally dense, as seen in Fig. 3.16. The work of Ghoniem et al. [127] shows that the readability of node-link diagrams is affected by density. For this reason, researchers have tried to develop techniques that allow us to visualize the general structure of a network and simultaneously visualize and understand the communities that can be dense.

The work of Henry et al. [22] presents NodeTrix, which attempts to solve the problem described above. NodeTrix is a hybrid approach that combines the best of node-link diagrams to visualize the general structure of a network, with the advantages of adjacency matrices (as in Henry and Fekete [128]) to show dense communities. NodeTrix

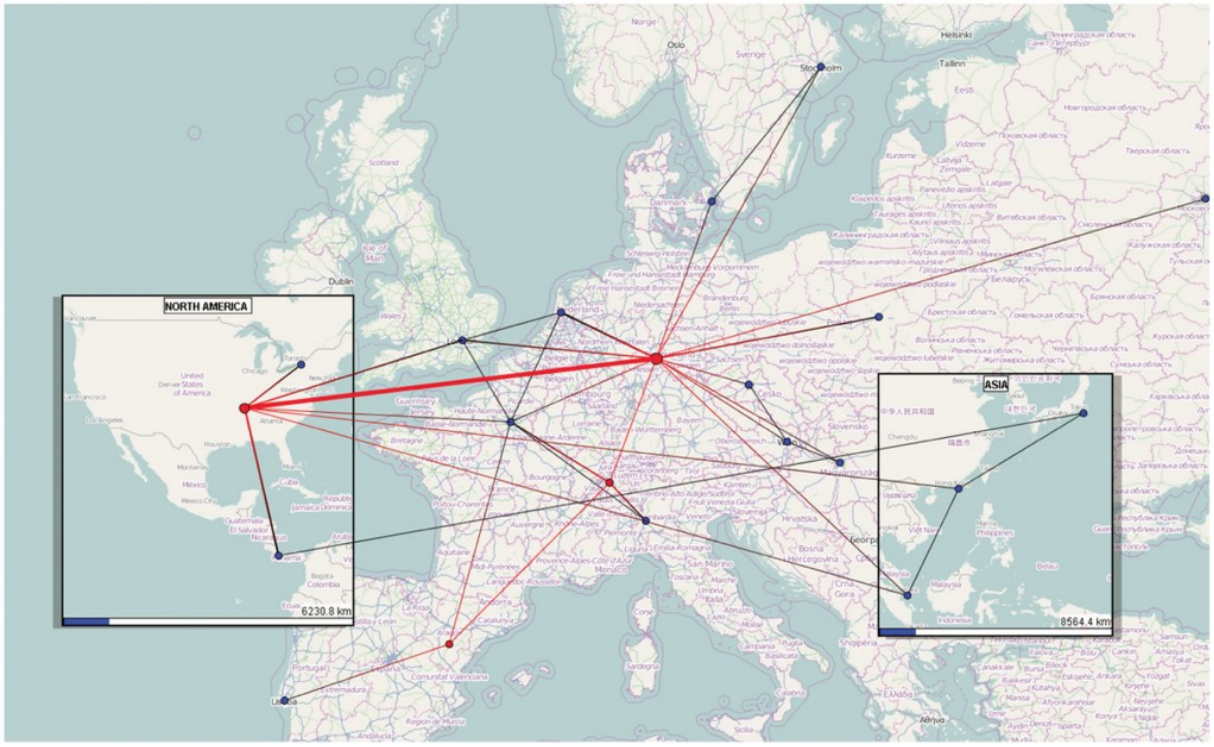


Figure 3.15: Result of applying zoom. Taken from Brodkorb et al. [20].

uses adjacency matrices to show intra-community relationships and regular links to show inter-community relationships.

Fig. 3.17 shows a visualization resulting from MapTrix. The visualization shows a co-autorship network in an information visualization field. This visualization is based on a standard node-link layout, but in areas with a higher density of nodes, they are visualized in the form of an adjacency matrix. Adjacency matrices have two advantages. The first advantage is that they make the visualization readable. Rows represent the nodes, and columns of the matrix allow that there are not so many crossings with the links. The second advantage is that the links can be drawn from either side of the matrix, which avoids overlaps and crossings. Additionally, the rows and columns can be reordered to reduce the number of link crossings. Finally, the visualization can be aggregated, as seen in Fig. 3.17. If a zoom is applied, for example, in the PARC (inside the figure), you get a visualization as in Fig. 3.18.

The work of Henry et al. [22], can be considered to accomplish its task fairly straightforwardly and easily understood. This research can be used effectively for social network analysis and by almost any user as the node-link graphs are easy to understand. The work described above has inspired others that improve on certain aspects that the Henry et al. [22] visualizations lacked. The work of Bach et al. [23] is inspired by the use of node-link representations and adjacency matrices to implement a technique called OntoTrix to represent Ontological networks. What differentiates this work from Henry et al. is that the latter is focused on displaying different parts of data based on semantic and structural properties. What the work is about is not only applying visualizations to social networks

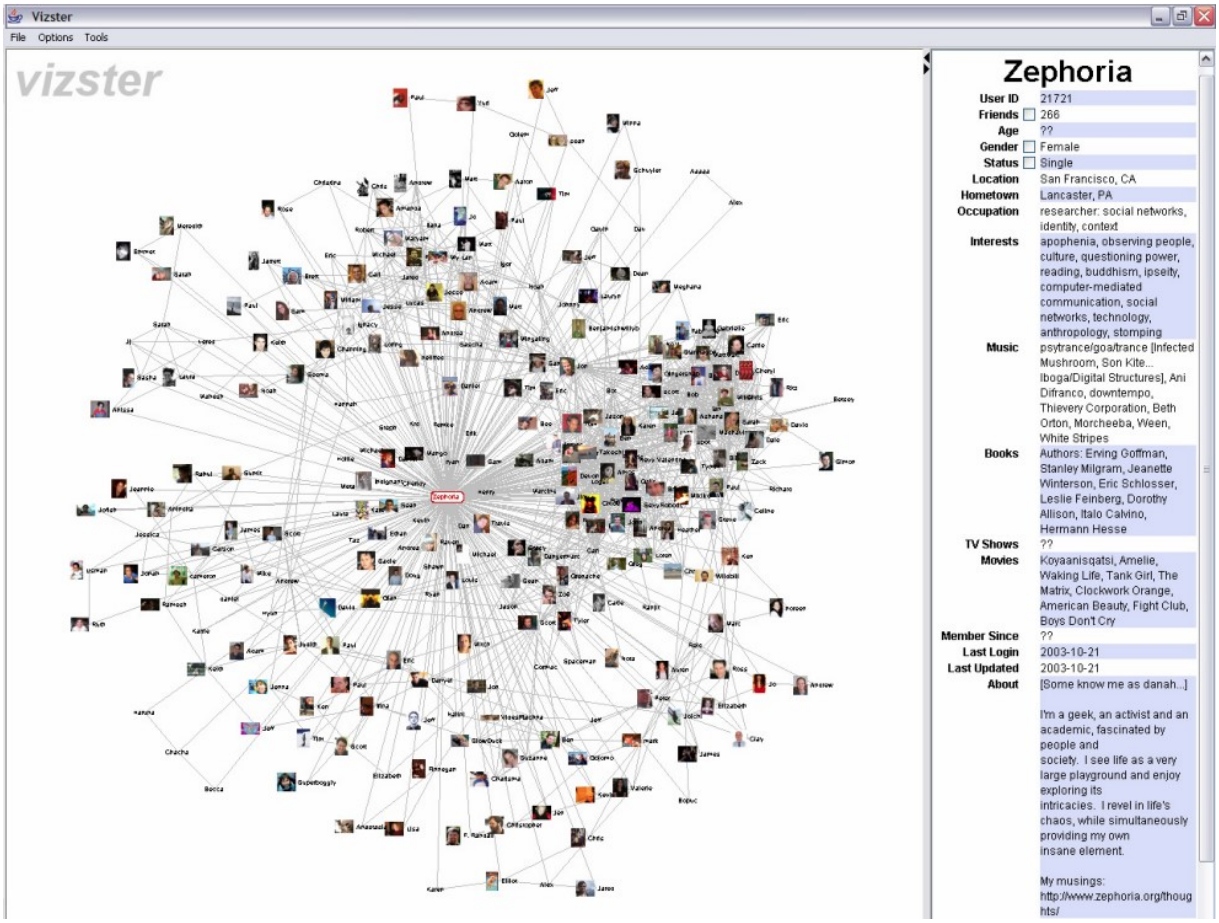


Figure 3.16: Node-link visualization system. Taken from Heer and Boyd [21].

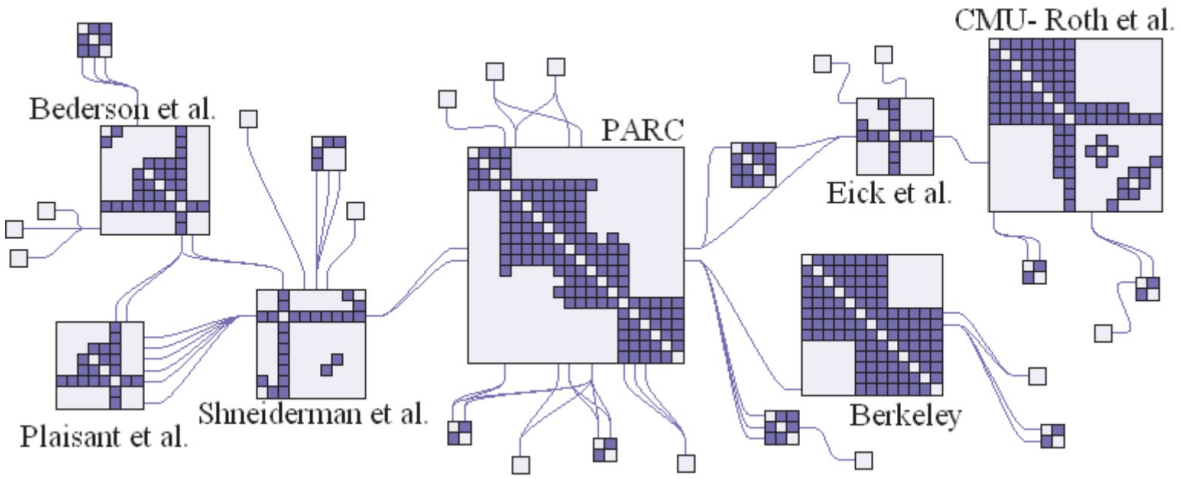


Figure 3.17: NodeTrix result of the InfoVis Co-autorship network. Taken from Henry et al. [22].

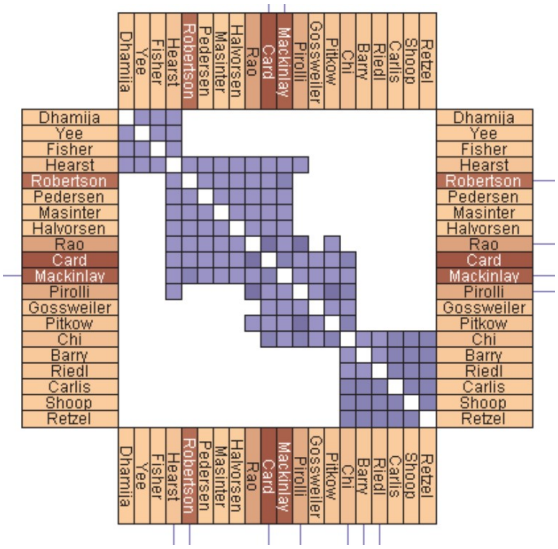


Figure 3.18: Zoom of PARC adjacency matrix from the InfoVis Co-autorship network. Taken from Henry et al. [22].

but also Ontologies[2]. According to Bach et al. [23], "In ontologies, instances belong to different (and possibly multiple) classes, and are connected by different types of object properties". Those types of properties or categorical variables are color-coded. They also add interactive elements such as bird's eye view zoom, filter, and details on demand. The work of Bach et al. is shown in Fig. 3.19.

Another way to represent networks is by using circular chord diagrams. Examples and uses of circular chord diagrams can be found in the works of Hennemann [129], Abel and Sander [24]. An example of a circular chord diagram is shown in Fig. 3.20. These diagrams with circular layouts are one of the oldest and most used methods for drawing graphs. In these diagrams, the nodes are drawn in a circle. The connections between the nodes passing inside the circle are the links. Usually, colors are used to distinguish the paths, and edge bundling techniques can also be used to reduce visual clutter. This technique is appropriate for applications where it is necessary to emphasize the clustering decomposition of a graph, where each cluster is drawn in a different graph.

In the work of Gansner and Koren [130], three techniques are proposed to reduce the visual clutter of a circular graph:

- Edge Bundling: Serves to deform the edges that join the edges and calculates the length of the curved edges.
- Node Ordering: It is based on achieving an optimal ordering of the nodes to eliminate edge crossings and shorten the length of drawn edges.
- Exterior Routing: It is based on taking a subset of edges from the inside of the circle and routing them around the outside.

[2] `https://ieeexplore.ieee.org/abstract/document/747902`

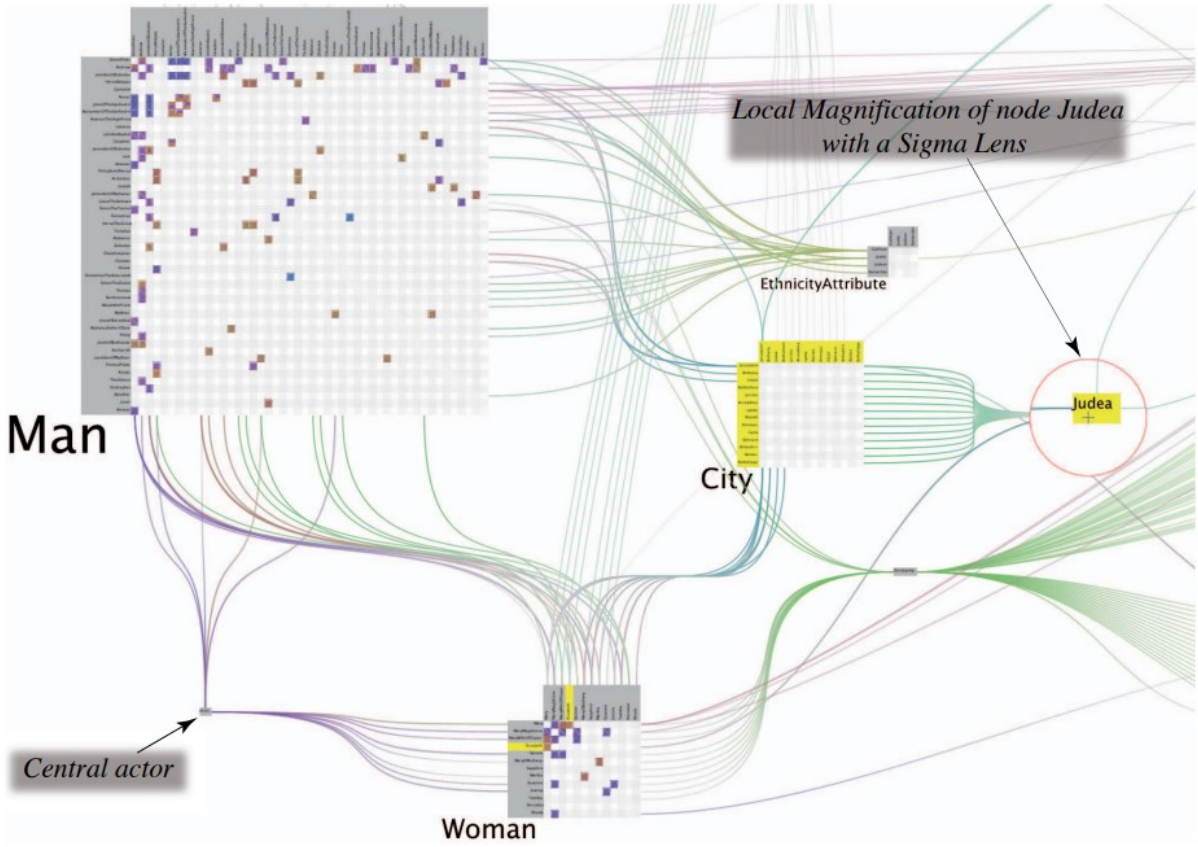


Figure 3.19: OntoTrix result from ontology corresponding to men and women who live in Judea. Taken from Bach et al. [23].

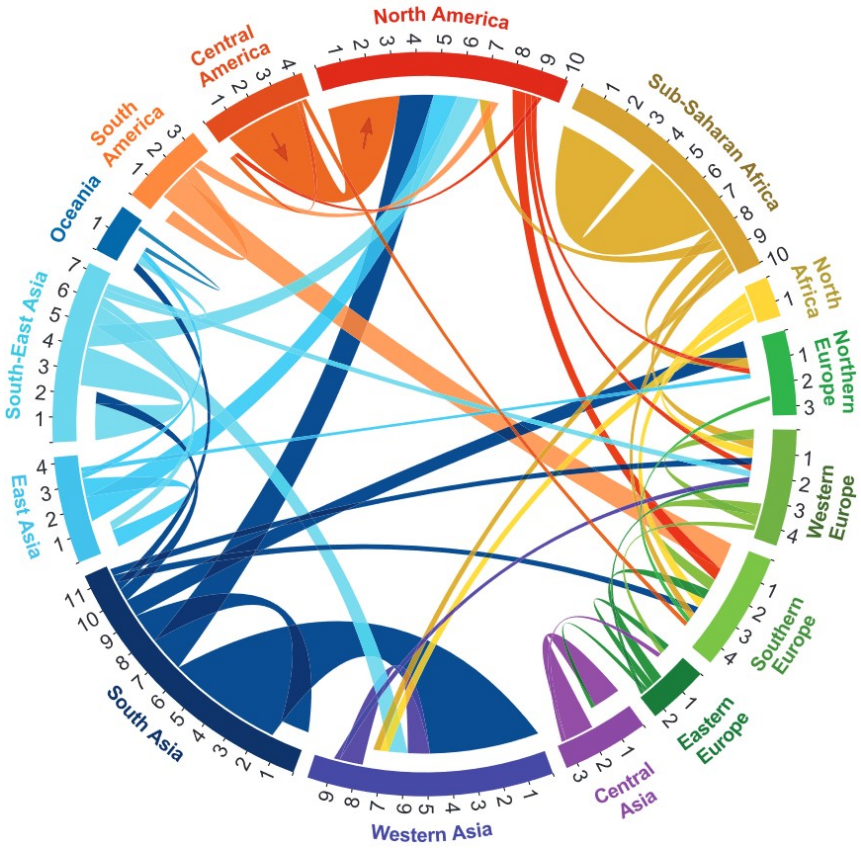


Figure 3.20: Circular chord diagram flows between and within regions during 2005 to 2010. Taken from Abel and Sander [24].

3.1.4 Twitter

In the literature, we can find several works with Twitter data that analyze the networks formed in this social network. Within these works, there is a technique that predominates. These are the node-link based graphs. Bongsug [25] proposes an analytical framework for analyzing tweets related to supply chains and analyzes the use of Twitter in this context. In Fig. 3.21, a network graph was constructed where the nodes represent the users who replied or received a replica of tweets containing the hashtag supplychain. The links are the relationships between users that occur through the replies. According to Chae, "The path length of most nodes is between 3 and 8. The network diameter is found to be 19, which is the longest path between two nodes in the network". From the visualization of the graph, it can be determined that the network is extended and sparse. It is determined that there are certain nodes that have larger connections than the average of the rest, called hubs. It is determined that there are several groups dispersed within the network, and by applying a community detection method by Blonded et al. [131], it is determined that there are more than 400 communities in the #supplychain network.

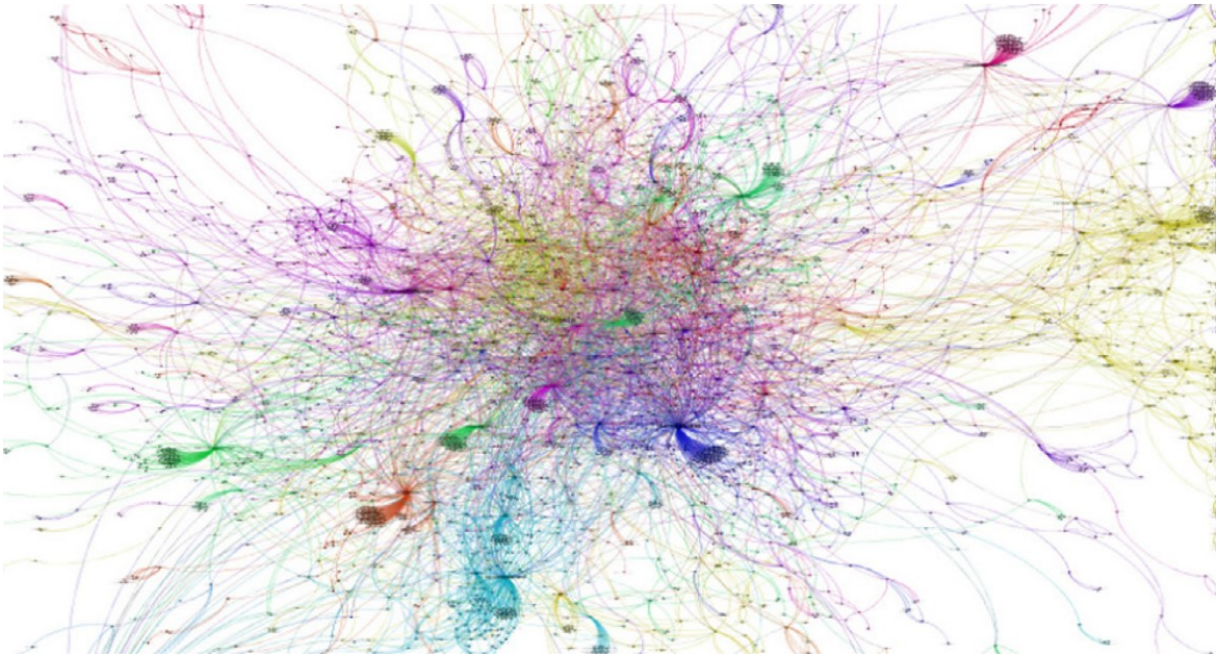


Figure 3.21: Node-Link graph of supply chain. Taken from Bongsug [25].

In the work of Guille and Favre [26], event detection is discussed, and a visualization based on node-link graphs is performed. In the paper, mention-anomaly-based event detection (MABED) is proposed. MABED is based on tweets and uses the mentions embedded in tweets to detect events and estimate their impact. Experiments were performed on English and French tweets where MABED was applied, and an event graph was subsequently constructed. Fig. 3.22 shows the event graph constructed. The main terms are represented with gray nodes (1), and the diameter is proportional to the event's magnitude. Related words are represented with blue nodes (3) whose links (2) have a thickness proportional to the related weight. The visualization allows clicking on a gray node to see the primary term. This graph helps to find similar events by topic and discover words repeated in

several events. Finally, nodes (a) and (b) represent two presidential candidates and appear together in a single event, the gray node that links them.

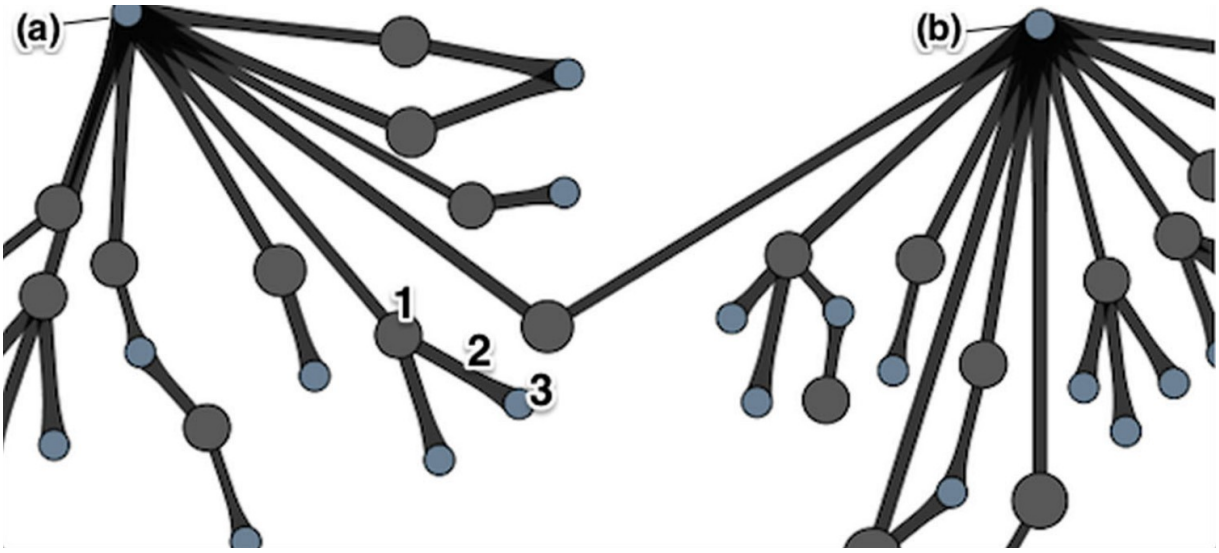


Figure 3.22: Topic oriented visualization. Taken from Guille and Favre [26].

In the same way, other works perform different types of Twitter analysis through a node-link graph. Some of them deal with content extraction and visualization, such as Kim et al. [132], Brummette et al. [133], and Molla et al. [134]. Other works focus on how information spreads through tweets. The work by Froio and Ganesh [27] focuses on the far right transnationalization discourse on Twitter. The study uses a dataset on activities and audiences of far-right Twitter users in France, Germany, Italy, and the UK. From the dataset, accounts that are far right representatives are selected. Then, a retweet network analysis is performed to explore the resonance of specific topics on Twitter. Finally, we quantify the level of topics and organizations with high levels of attention across borders. Fig. 3.23 shows the far right network, where users who retweeted the selected accounts more than five times were taken. Each node represents unique Twitter users, and the edges represent retweets. In the graph, there are 55,983 retweets, of which 1,617 were transnational. Four different national communities were detected, corresponding to the sampled countries, using the community detection algorithm of Blondel et al. [135]. Finally, transnational users were identified as those who retweeted content from more than one national community, for example, when retweeting content from Italy and the United Kingdom.

Similarly, the work of Tsubokura et al. [28] visualized the spread of information by influencers after the Fukushima Daiichi nuclear power plant accident. The authors examined how original tweets and retweets from Twitter were used for dissemination related to the nuclear plant accident. Tweets and retweets about the event were obtained from NTT DATA Corporation[3]. In the analysis, it is determined that only 2% of the accounts were the source. These accounts were defined as "influencers". In Fig. 3.24, the network of retweets generated by influencers is shown. Each node that has a label is to identify the influencers. The size and color of the nodes indicate the total retweets and their group.

[3]https://www.nttdata.com/global/en/

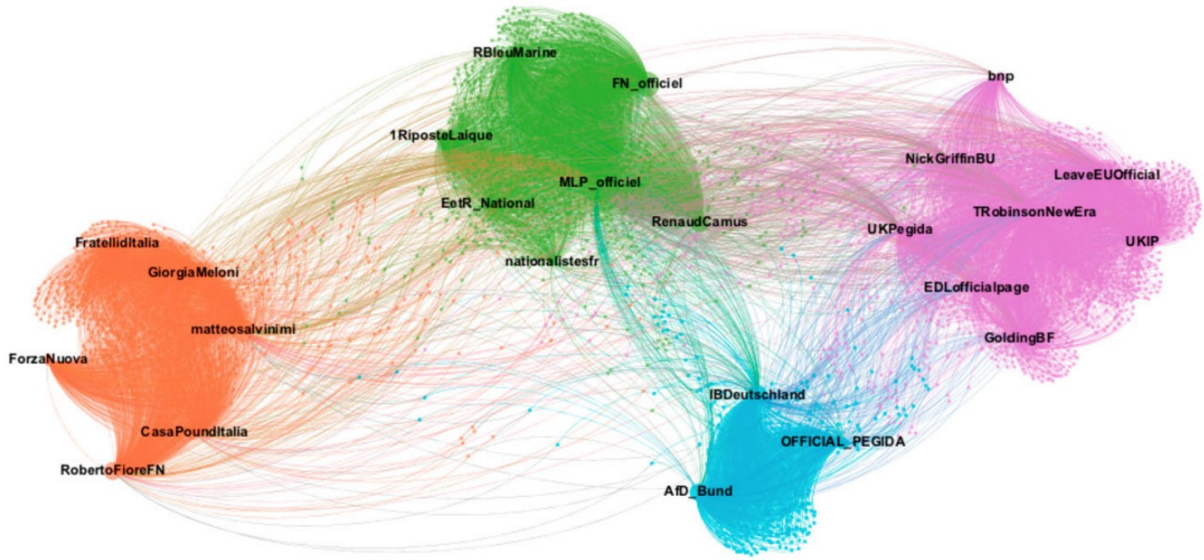


Figure 3.23: Retweet Network. From left to right: Italy, france, Germany and UK. Taken from Froio and Ganesh [27].

The visualization determined that there are several groups and that the interactions within the same groups were relatively high while the interactions between groups were low.

The predominance of methods based on node-link graphs is notable, but other methods have been used for moving data from Twitter. For example, the analysis of Hawelka et al. [29]. In this paper is analyzed geo-located tweets to discover global patterns of human mobility and compare the mobility characteristics of different countries. First, the paper emphasizes that the geo-located Twitter data are very few, approximately 1% of the total obtained, which limits the research to some extent. The data was obtained through the Twitter API in 2012. We obtained 944M records from 13M users. After obtaining the dataset, these data were cleaned to remove tweets from accounts that did not belong to individuals. Subsequently, each user was assigned a country of residence so that the origin of the users was known. This contributes to understanding human mobility since it is known whether a user is a resident or a visitor. Now it is possible to know to which country he travels and when. Additionally, any interaction on Twitter in any part of the world other than their country of residence is considered a trip.

Once users were assigned a place of residence, a global network of flows between countries was constructed. Each country corresponds to a node, and the edges were weighted by the number of users traveling between countries. As the network was sparse, it was decided to take the top 30 flows between countries. Countries with less than 500 users on Twitter were filtered out. The result is shown in Fig. 3.25. Finally, it is determined that there was an increase in human mobility in the period studied and a great diversity in the destinations. Some countries, such as Australia and New Zealand, are affected by their isolation. Finally, it is seen that there are patterns of mobility determined by cultural conditioning and special events.

As we have seen, most of the papers presented above have focused on structural aspects of trajectories and not on spatial and temporal aspects, but these dimensions should not be neglected. Yin et al. [30] focus on applying a visual-analytics approach to study multi-

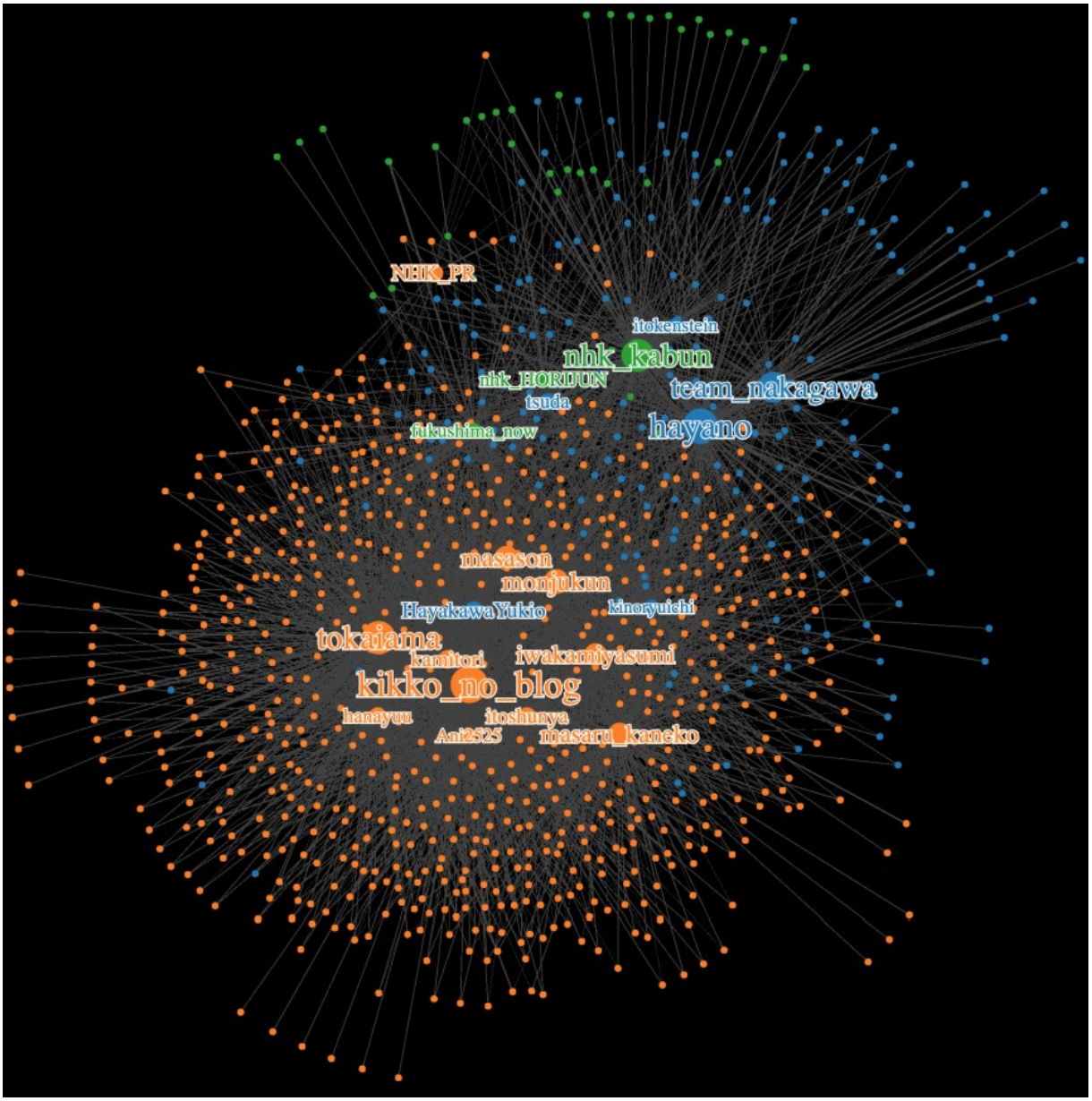


Figure 3.24: Retweet network diagram about Fukushima Daiichi nuclear power plant accident. Taken from Tsubokura et al. [28]

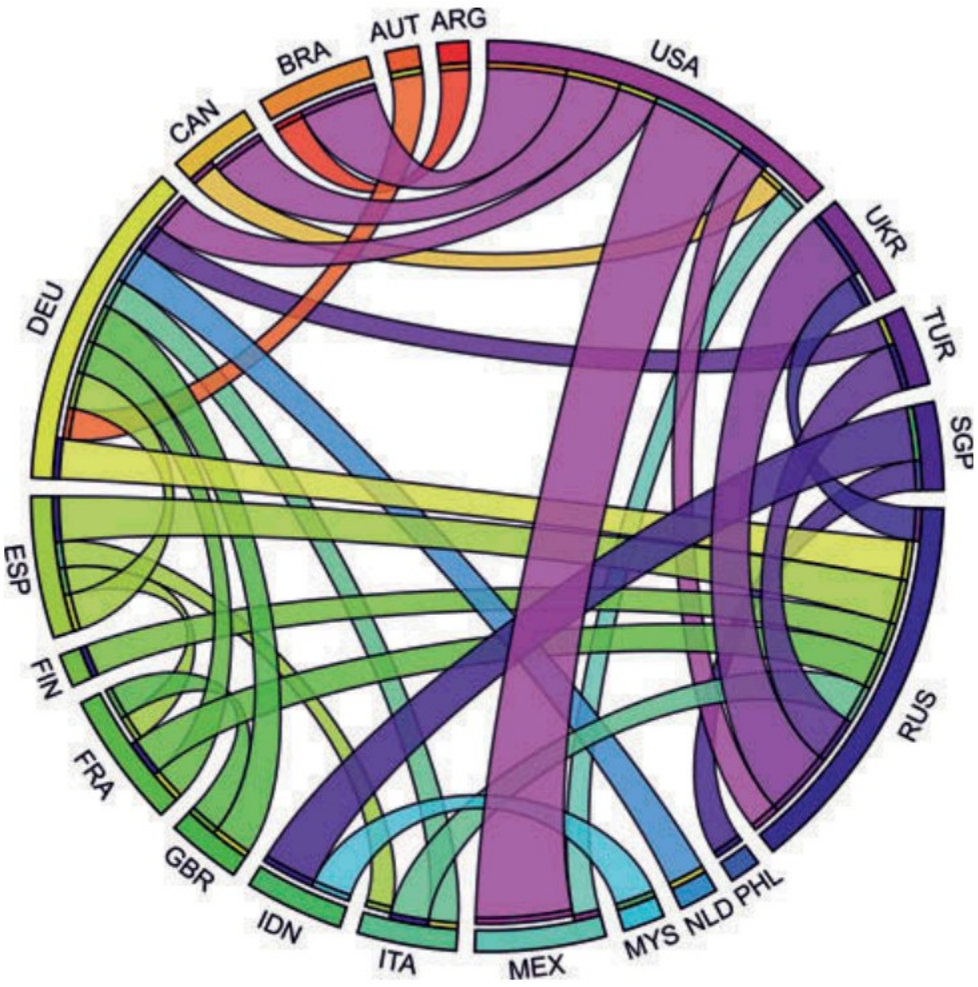


Figure 3.25: Circular chord diagram showing flow between 30 countries. The links' colors codify the destination. A thin dash indicates the country of origin at the end of the links. Taken from Hawelka et al. [29]

scale spatio-temporal Twitter user mobility patterns in the United States in 2004. They extracted 6,147,430 geo-located tweets using the Twitter API specifying the area of interest in the United States. Using Apache Hadoop, the tweets were managed and processed. Space-time trajectories were constructed with each tweet's location, timestamp, and ID information. The trajectories constructed have different levels of detail through spatial aggregations using a MapReduce algorithm. The geographic space of the United States is divided into ten hierarchical spatial layers. This is why the analysis is multi-scale. Each location of a user's trajectory is distributed to corresponding spatial units. Fig. 3.26 shows a 3D virtual globe developed with an open-source WebGL virtual globe and map engine. The map is adapted to different spatial scales and allows several interactions such as time window and zoom. In Fig. 3.26, the movement flows are visualized, where the time window can be specified. Hovering the mouse over each line of the map displays the value of the movement flows for the in and out directions. Zooming in and out automatically provides a different level of detail. For example, Fig. 3.27 shows a new, more granular level of detail when zooming in and out around the city of Chicago. To conclude, the work of Yin et al. [30] provides insight into mobility patterns across multiple spatial scales and temporal ranges. However, Twitter data with geo-location cannot generalize the entire population and may not reflect a complete picture of real-world human movements.

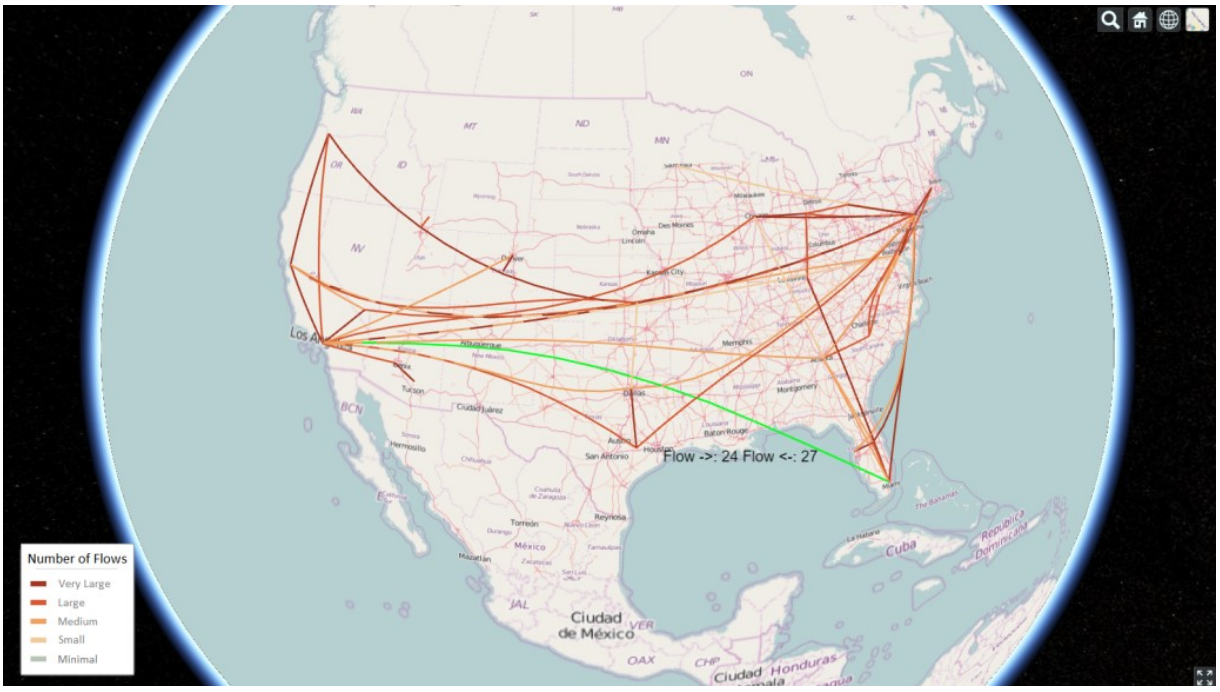


Figure 3.26: 3D interactive web mapping interface. Taken from Yin et al. [30]

3.1.5 Other Works Related

Other works do not focus precisely on InfoVis but give a guideline of its usability. One example is the work of Ferrara et al. [136]. This paper discusses the importance of bots in online social networks (good or bad), then presents a taxonomy of bot detection systems.

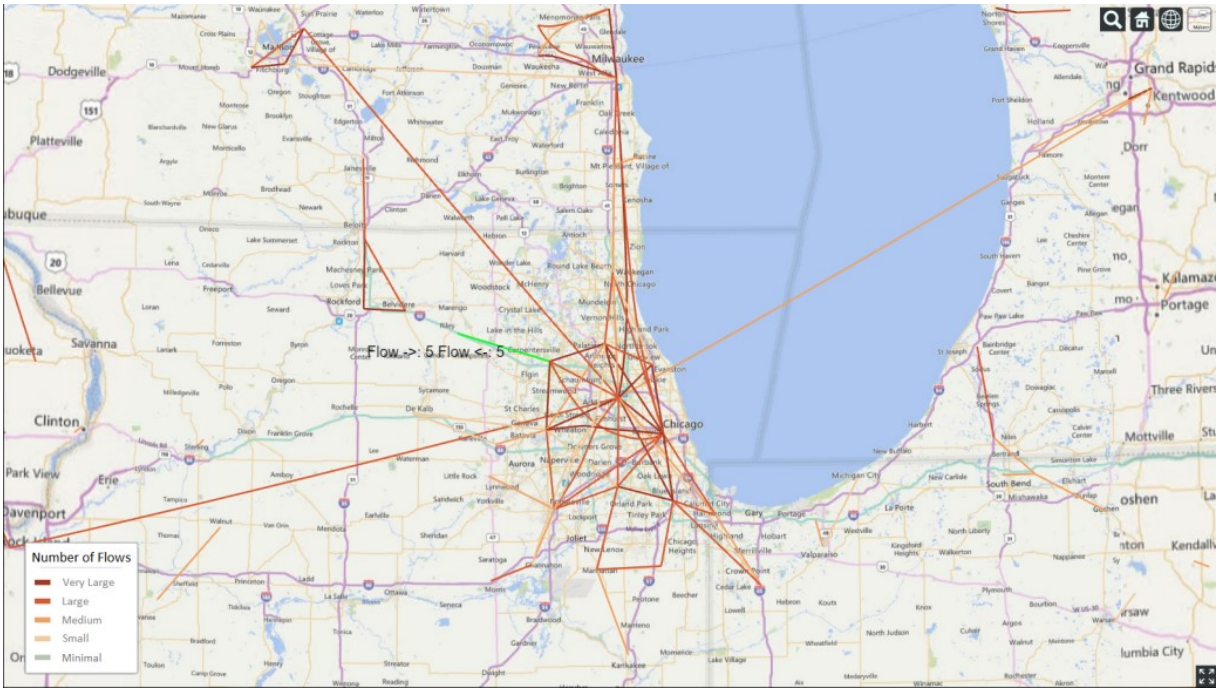


Figure 3.27: Movement flows around Chicago city. Taken from Yin et al. [30]

The article states that bots, in certain cases, are benign and useful. One case in which bots are "good" is when they are used to aggregate content from different sources such as news.

Bots can be dangerous and cause harm. Bots in social networks can be used to spread unverified information or rumors. Unverified information can manipulate, mislead or alter many users. Bots can be used to infiltrate political speeches and change users' opinions and then influence political elections. Even bots have the power to manipulate the stock market by spreading incorrect information about a product or company.

For the reasons explained above, working on techniques to detect bots is important. The most common way is to use several approaches for a more robust detection system. In the article by Ferrara et al. [136], a taxonomy for bot detection approaches is proposed. One approach is Graph-Based Social Bot Detection which focuses on the structure of social graphs. Another approach is Feature-Based Social Bot Detection which can use machine learning and deep learning methods for bot detection.

An example of a detection system is the one proposed by Davis et al. [31]. The tool proposed by the authors is called Bot or Not? and was one of the first systems for detecting bots on Twitter. The system implements a detection algorithm that relies on highly predictive features. These features capture suspicious behavior and separate bots from humans. Machine learning algorithms are used and achieve detection accuracies better than 95%.

Additionally, Bot or Not? Presents interactive visualizations that provide information about the system's exploited features. An example of the system visualizations is shown in Fig. 3.28. The same figure illustrates how the bots affect the online debate on vaccination policy and how the online debate is manipulated. The size of the nodes indicates the

influence, i.e., the number of times a user is retweeted, and the nodes in red are most likely to be bots.

There is work such as Bliss et al. [32], focusing on replicas within Twitter. The replicas within Twitter usually form a thread. Then, it is possible to form a graph with the network of replies, where nodes are users and edges represent reply events over a defined period. In this work, a network with user similarities is formed higher than typical networks of users and followers.

In the article by Bliss et al. [32], the structure and dynamics of the Twitter social network are constructed and examined on time scales of days, weeks, and months. As the article mentions, many papers work with networks derived from the mutual following on Twitter, but the structure formed can be misleading. The article explains that users and their followers exhibit an affiliation but do not necessarily interact. In other words, followers do not necessarily read or respond to tweets.

The article proposes that a reciprocal-reply network can be constructed, where two nodes are connected if the two nodes have responded to each other. The paper relates the users' happiness to the network's structure. Fig. 3.29 shows the reciprocal-reply network of tweets obtained through the Twitter API in a week. For the layout, they use the Force Atlas 2 algorithm [137], where nodes are plotted together if they are highly connected. The Gephi [138] software package is used to create the visualization. Fig. 3.29 shows the result of the reciprocal-reply network. Colored nodes represent the connected components. It is visualized that a giant component of 15,342 nodes (blue color) belongs to 76% of all nodes.

The Bliss et al. paper [32] concludes that users' average happiness scores positively and significantly correlate with users one, two, and three links away. Finally, the article found evidence that better-connected users write happier status updates.

3.2 Summary

In this chapter, we have analyzed different state-of-the-art papers categorized according to how the geographic formation is displayed. Mapped emphasizes the importance of showing the direction of links and the magnitudes of flows. It is seen that map-based visualizations can have many problems related to link overlaps and map region overlaps. There may be too many links that can be mitigated with edge bundling techniques, but occlusions may still occur. Also, it is appreciated that interactions and the use of colors can be fundamental to improving the visualizations or modifying them according to the need. An important detail of several techniques within Geo is that there can be occlusions in the visualizations when there are many points of origin or destination. Additionally, due to the techniques implemented to reduce clutter and occlusions, points of interest cannot be seen in a granular manner. Finally, the temporal aspect of the trajectories is often neglected.

Within the Distorted section, it is determined that too many distortions would result in the loss of the proportions and geography. One of the problems within this type of visualization is that the algorithms to deform the maps can be slow when dealing with many data points. There can also be large deformations that compromise the fidelity of the data and the geography. This section emphasizes the use of interactions and that neglecting the temporality of the data is common. At last, flows are also not seen in a

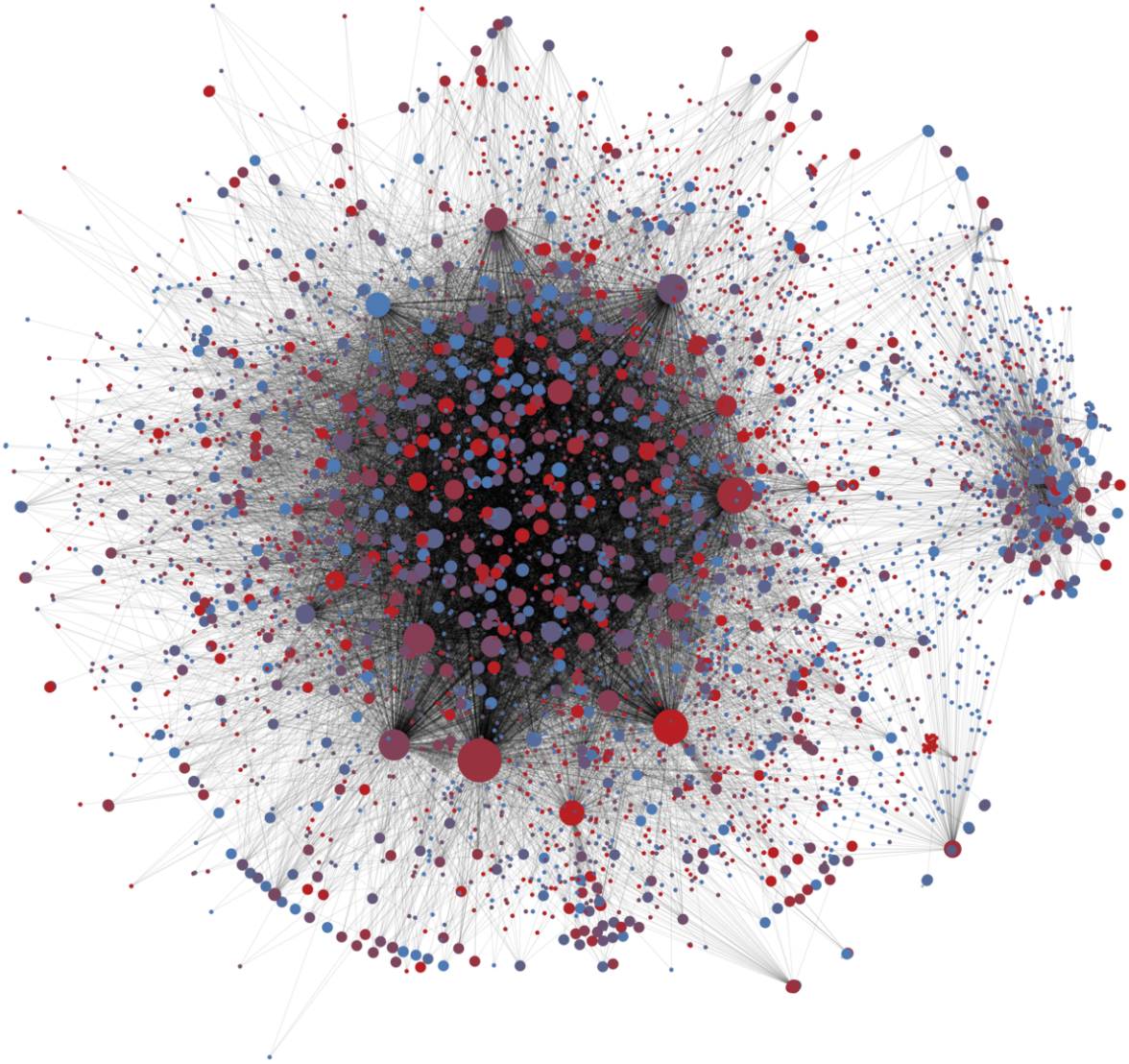


Figure 3.28: Visualization showing the spread of the Hashtag, “SB277” about a vaccination law in California. The nodes are Twitter accounts posting the Hashtag “SB277”. The lines between them show the retweet of tagged posts. Larger nodes are accounts that retweet more. Red nodes are probably bots; blue nodes are probably humans. Taken from Davis et al. [31]

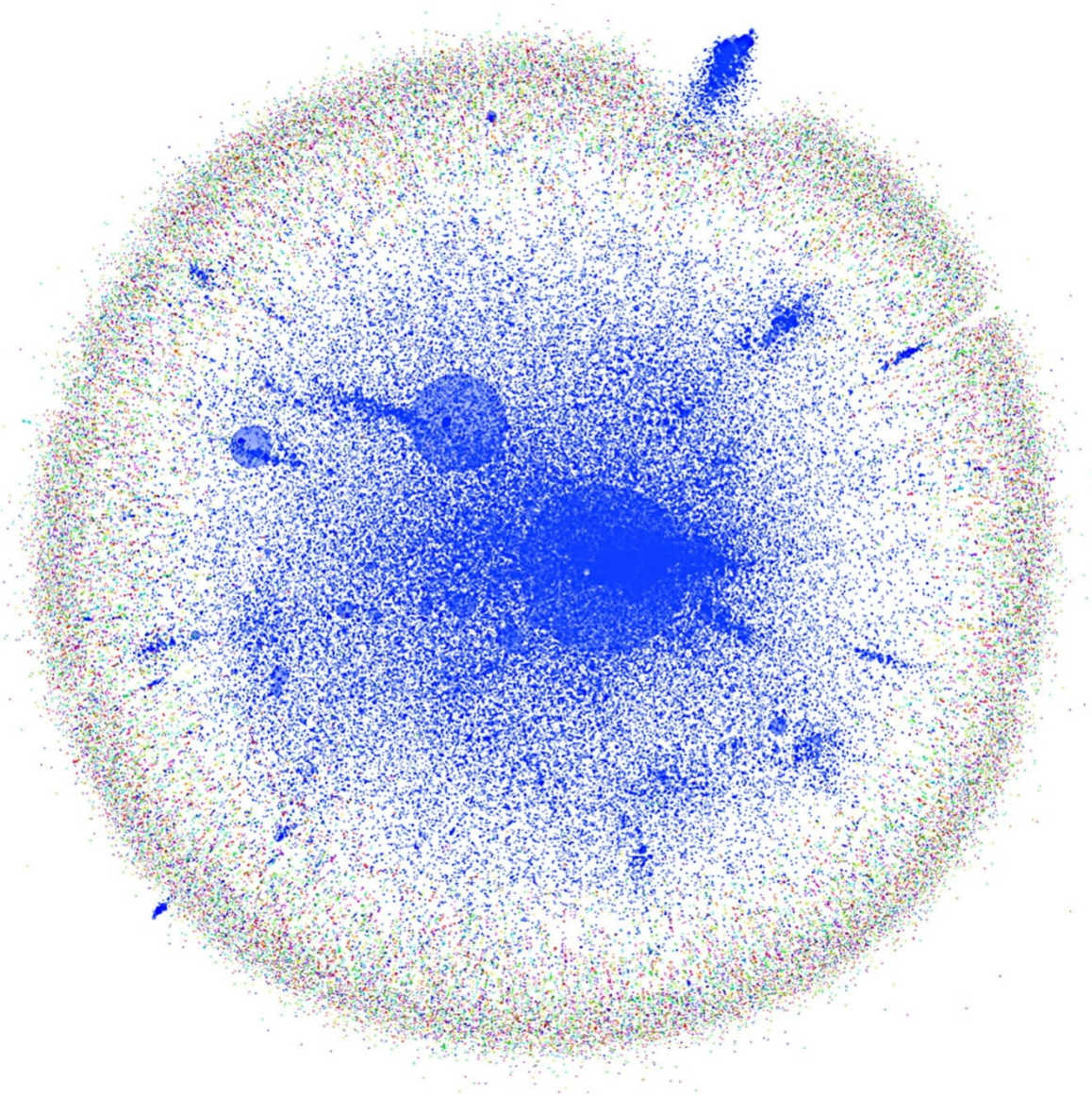


Figure 3.29: Visualization of 162,445 nodes of the reciprocal-reply network in a time window of one week on December 2008. The colors identify connected components, and the nodes' size is proportional to their degree. Taken from Bliss et al. [32]

granular way when grouping trajectories, making it difficult to differentiate between people moving from one place to another, for example.

In the section on abstract visualizations, it can be seen that the node-link representation is used for the representation of networks. The graph representations can be complemented with adjacency matrices to reduce the visual clutter and cross-linking. However, in the case of large numbers of nodes, the matrices increase in size rapidly, and the links tend to cross. It is also seen how interactions are significant in these works, such as zooming and using colors to differentiate paths. The circular chord diagram was also explored, but this visualization is commonly used with edge bundling and is more suitable for visualizing flows of several edges and not for visualizing individual edges.

Graph visualizations can be used in various ways and contexts, such as community detection, event detection, transnationalization discourse, and information dissemination. In other related work, visualizations are used in different contexts, such as bot detection and how they affect discussions, and can relate users' happiness to the network's structure. In the Twitter section and the other related works section, it is clear that most of them work with graphs to represent the networks in the social network, where the nodes are usually the users. Also, the most common is that they work with retweets and hashtags and leave aside what are the conversations within Twitter. In these works, working with different interactions and channels is crucial to encode different data types. Finally, Once the state-of-the-art works have been analyzed, it is shown below how the best ideas can be used for developing a visualization tool in the next chapter, considering that the visualization depends on the dataset.

Chapter 4

Methodology

This section explains the process for developing a visualization to analyze and understand the trajectory of Twitter conversations following Munzner's model [8] detailed in 2.4 and 2.8. First, the requirements analysis will be carried out, where the conditions the developed information visualization tool must fulfill will be specified. Then it is described how the data sets are obtained with the Twitter API. Later, it is explained the visual encodings used and the implemented interactions that satisfy the requirements presented in the first part of this section. Finally, a validation analysis is performed.

4.1 Requirement Analysis

As discussed in the State of the Art section, many papers in InfoVis focused on Twitter data do not work with Tweets. They work with users. Much of the research focuses on users and the propagation of Hashtags. One aspect worth analyzing is the trajectory formed by Twitter conversations. The conversations aspect has been focused on tasks related to sentiment analysis or information propagation but not on conversations that occur within the Twitter platform. Conversations in real life are a fundamental activity, as they are used to share information and can be very important on a social level. Conversations serve to exchange ideas, can influence thoughts, and is a way to keep in touch with other people. Finally, Twitter conversations emulate conversations that happen in real life.

By developing a visualization tool, it will be possible to discover and explore what is happening within Twitter conversations. Important questions can be asked, such as what happens to a conversation over time, how soon a conversation will lose its reach, or how a conversation interacts with other users' Tweets over time. There will also be questions about the structure of conversations on Twitter. The user may wonder whether or not all conversations may have the same patterns. The question may also arise as to what conversational structure a Twitter bot may have and use the tool to find possible bots. Visualizing the trajectory of Twitter conversations powered by sentiment analysis could explain how positive or negative information is transmitted from Tweet to Tweet.

Conversations on Twitter can be very important for commercial brands. The more conversation a brand generates, the more popular it is. More conversation about a product could mean more excitement. Twitter's research claims that increasing the conversation

about a brand on Twitter can also increase the sales volume related to the brand. One could also determine the volume of participation of people on Twitter by gender and age and see how their Tweets influence conversations.

Visualizing conversations on Twitter over time will answer the questions posed here, so the following list of requirements is identified:

- R1: Visualize relationships between Tweets. This requirement has to do with the fact that in the Tweets, there are several interactions such as retweets, quotes, and replies that can relate several tweets to others.
- R2: Show the time aspect of Tweets. Tweets have a timestamp at the time of their creation. With this dimension, it is possible to reveal patterns over time. The user can discover a trend or visualize how the creation of Tweets evolves. Leaving out this dimension reduces the ability to understand the dynamics within Twitter.
- R3: Determine the trajectory of Tweet conversations. Thanks to the different relationships between Tweets, it is possible to visualize the traceability of the trajectory of conversations through Tweets.
- R4: Interact dynamically with the application. It is necessary to use interactions to visualize the data easily and correctly according to the requirements R1, R2, and R3. These interactions can be zoom, highlight, select, and filter. The interactions will allow analyzing the information quickly and easily and a better understanding of the trajectories of the conversations.

4.2 Data Acquisition

4.2.1 Twitter Introduction

A Tweet is the main object within Twitter that can contain up to two hundred and eighty characters and be published publicly or privately. Tweets can contain various multimedia elements such as images or videos. They can also contain locations, polls, and URLs.

Within Twitter, several interactions can be generated from a Tweet. A Tweet can receive likes. A Tweet can have a reply or replies, it can be retweeted, and it can also be quoted. Replication occurs when someone replies to another person's tweet. Retweets are a re-posting of the Tweet and serve to share a Tweet with more users. Finally, Quotes are retweets with comments. The latter part is useful for sharing someone else's Tweet with an added comment.

Thanks to reply interactions, conversations can be generated on Twitter. A conversation starts when a reply to a Tweet is created. This conversation can continue to grow as more replies continue to appear.

4.2.2 Twitter API

Twitter provides an API (Application Programming Interface) that allows the extraction of data from Twitter for further analysis. The Twitter API arises due to the need to

investigate the public conversations generated in this social network. The Twitter API includes many functions and several levels of access so that developers and academics can feed on the daily information on Twitter. The latest version of this API is version two. The Twitter v2 API allows you to obtain objects in JSON format that contain the information you want.

There are three access levels in the Twitter v2 API. There is the basic, elevated, and Academic Research level. The access level used in this work is the Academic Research level, as this level has the goodness listed below, which is useful for this work.

- Obtaining up to 10 million Tweets per month.
- Search queries up to 1024 characters.
- 100 request for every 15 minutes.
- Access to public Tweets from the complete archive.
- Get up to 100 Tweets per request when using the tweet.fields parameter.

There is a large amount of metadata that relates to a Tweet. Twitter provides information about the fields that a Tweet can contain on its data dictionary page. The list below is a summary of this information. The default fields are obtained by not specifying field parameters when requesting with the Twitter API.

- id (default) (string): Unique Tweet Identifier. This field is used to retrieve a specific tweet.
- text (default) (string): The UTF-8 formatted text of the Tweet. The text could be used for Sentiment analysis and classification.
- author_id (string): Unique identifier of the user who posted the Tweet. This is used to hydrate with information about the user.
- conversation_id (string): Id of the original Tweet from which a conversation originates (includes direct replies and replies of replies). This field is used to reconstruct a Tweet conversation.
- created_at (date ISO 8601): Tweet creation time. Created_at field could be used to understand the date and time the Tweet was created.
- geo (object): Location details. This field could be used to detect if a Tweet is related to a location.
- referenced_tweets (array): List of Tweets to which the consulted Tweet refers. For example, if the consulted tweet is a replica of another one. The Tweet queried within the referenced_tweets field will contain the id of the parent Tweet and the type of interaction that originated this Tweet (Retweet, Reply, or a Quote). This field is used to understand conversational aspects of interactions.

The Tweet, as mentioned above, is an elementary component that relates to all the dynamics of Twitter, but in addition to the main object that is the Tweet, there are secondary objects that contain information about the user, place, media, and others[1]. The Users and Places object are summarized below.

Users Object:

- id (default) (string): Unique user Identifier. Id Used to retrieve information about a specific user.
- name (default) (string): The name of the user defined in the profile. It usually has a character limit of 50 characters.
- username (default) (string): Alias of a user on Twitter. It usually has a maximum of 15 characters.

Place Object:

- full_name (default) (string): Full place name. It can be used to classify a Tweet according to a specific place name.
- id (default) (string): Unique identifier of a site. It can be used to retrieve a place with programming.
- country (string): The name of the country to which the place belongs. It can be used to classify a Tweet according to the country name.
- geo (object): This field contains location details in GeoJSON format.

The Twitter API has different endpoints used to obtain the available objects mentioned above and others[2]. For example, the Tweets lookup endpoint uses the HTTP GET method to return one or more Tweet objects knowing the Tweet ID.

There is an endpoint called users lookup for obtaining the User object, and it can also be obtained as an expansion of the Tweet object. The Place object can only be obtained as an expansion of the Tweet object since it is not a primary object in any endpoint. In this work, we use the Full Archive Tweet Search Endpoint. With this endpoint, you can retrieve Twitter conversations on a specific topic. Tweets can be retrieved according to keywords, text matches, hashtags, URLs, and more. You can also use filters to make the data retrieved relevant to the topic you want to investigate. Queries can be performed using Boolean operators and parentheses to refine the search. We suggest checking the official Twitter documentation[3] for a complete overview of the construction of queries for searching Tweets.

The Full Archive Tweets Search Endpoint has other search parameters. A time period can be specified. You can specify the maximum number of resulting Tweets to retrieve in one request. You can also specify whether or not you want several object fields present on

[1] https://developer.twitter.com/en/docs/twitter-api/data-dictionary/introduction

[2] https://developer.twitter.com/en/docs/twitter-api/migrate/twitter-api-endpoint-map

[3] https://developer.twitter.com/en/docs/twitter-api/tweets/search/integrate/build-a-query#build

Twitter, such as Tweet, User, and Place objects. For example, you can get Tweets that include, in addition to the text and the id, information about their public metrics, the place, and the user who posted the Tweet. The way to include additional information can be understood more deeply within the information provided by Twitter on its developer site[4].

Tweets can be obtained according to a specific period, keywords in the text, hashtags, URLs, and more.

4.2.3 Conversation_id as a Filter Operator

A Twitter conversation is a series of Tweets, each of which has a conversation_id that matches the ID of the original Tweet that started the conversation. All tweets, whether replies to a given Tweet or replies of replies are part of the conversation that grew out of a single original Tweet. Within a conversation, regardless of whether multiple reply threads are generated, all Tweets will have the same conversation_id. As mentioned above, Twitter conversations can grow in length and complexity as more users participate in a conversation.

The Twitter API allows us to use some conversation_id to reconstruct a complete conversation. A conversation can be obtained using the Full Archive Tweets Search Endpoint and specifying in the query the conversation_id.

4.2.4 Tweets Acquisition

Once certain concepts involving the Twitter API have been explained, it will be explained how to get Tweets. The Twitter API provides sample codes for the use of its endpoints. These codes are available in the official Twitter repository for developers on GitHub[5].

Before getting tweets using the Twitter API, it is necessary to create an account on Twitter and apply for one of the access level modalities Twitter has on its page for developers. In this work, an account with academic research access is used. Once the configuration to have access to the API is done, it is possible to obtain tweets that belong to a conversation using the Python programming language. A simple way to obtain the ID of a conversation is to enter the Twitter feed and select a Tweet that preferably has a large number of replies and retweets. When a Tweet is opened in the last section of the URL address of the Tweet, a number will appear; this number is the identifier of the tweet. This ID can be used in the query made when making requests in the Twitter Full archive search endpoint. With this Id, It can be obtained data such as the Tweet creation date with the same API, and use this information together with the Id to start obtaining Tweets. As the Twitter endpoint allows a maximum of one hundred Tweets per request, loops must be performed to obtain more than one hundred Tweets. Within the response to the request, the next token is found, and with this value, it is possible to continue with the extraction of Tweets. If there is no next token in the response, there are no more Tweets corresponding to the query. Also, programming in Python can set the maximum number of Tweets required. The latter

[4] `https://developer.twitter.com/en/docs/twitter-api/tweets/search/quick-start/full-archive-search`

[5] `https://github.com/twitterdev/Twitter-API-v2-sample-code`

is useful when obtaining certain samples of the total number of Tweets participating in a conversation. The Tweets obtained are in JSON format and have the format presented in App. 1.

The returned JSON has three main elements: data, include, and meta. The value of Data is an array that has several objects. These objects are each of the Tweets that have the same coversation_id that was used when making a request. The value of Include is another object that contains, in this case, two elements, users and tweets. The value of users is an array of several objects, and these objects belong to each of the users that made a tweet and are found in the Data. The value of tweets is also an array of objects. Each object in the Tweets array is the information of the tweets referenced in Data. Finally, the meta value is an object with information about the request and contains the next token, which must be used in the next request to continue getting tweets from the same conversation.

4.2.5 Tweets Treatment

Once all the necessary tweets are acquired with the JSON format presented in the previous section, It is important to process this information. Using Python, it possibly creates a new JSON file containing two "name: value" pairs. The first is called nodes and is an array of objects. The second element of the new JSON file has the name links (see a sample in App. 2).

A feature that differentiates this work from others is that Tweets can be represented as nodes, and the interaction between one Tweet and another is a link. Each Tweet obtained through the API can be viewed as a node containing id, username, and time (even text). A link between two nodes can be seen as an object containing a source and a destination. In this case, links can be obtained by taking the tweet identifiers inside the referenced_tweets field as the source and the tweet's id as the destination. Finally, in this way, we get a new JSON object that contains nodes and links suitable for easy use in creating visualizations. An example of the JSON the visualization tool takes is the one seen in the App. 2. For the visualization tool to work correctly, the following information is needed in the nodes: tweet id and tweet time. The links need to have the source, the destination, and the type of interaction. For informative purposes, it has been decided to have the following information: the author_id, name, and username of the person who posted the tweet, conversation_id, and the tweet's text.

As seen in section 2.12.3 , the amount of Tweets containing latitude and longitude is very poor. Therefore this parameter within each of the node objects cannot exist. Knowing the type of data available and by the Munzner model, it is determined that visualization can not be created in which the trajectory of tweets can be seen on a geographic map. An alternative to visualizing tweets on a map is using graphs.

4.3 Visual Mapping and Functionality

4.3.1 Design Summary

Based on the requirements analysis, a visualization tool is proposed that facilitates the analysis and exploration of Twitter conversations using graphs (see Fig. 4.1). A series of visual encodings and interaction techniques are proposed to solve the requirements. Forced direct graphs are proposed to visualize the relationship between Tweets within a conversation [R1]. A linear color scale bar and a temporal radius slicer are used to show the temporality of the tweets [R2]. The graph and the colors of each node are used to show the trajectory of tweets in conversations [R3]. Finally, interactions such as Semantic Zoom, Filter, Drag, Drop, and Graph control parameters facilitate visualization exploration [R4].

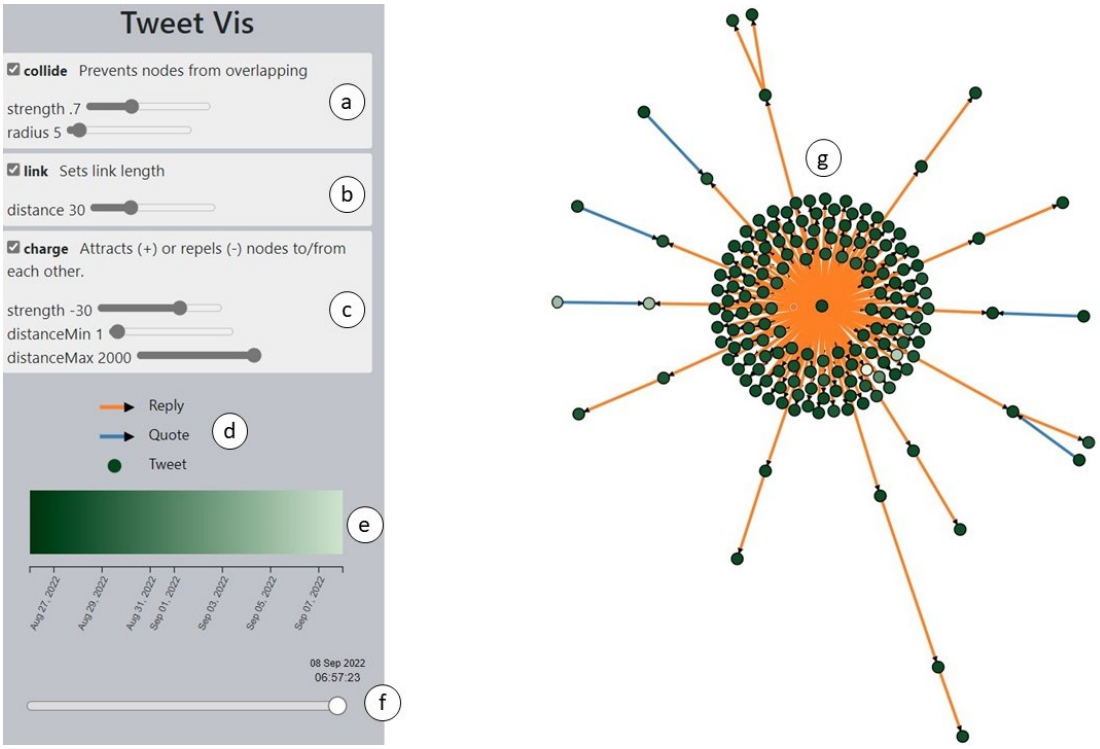


Figure 4.1: Twitter conversation visualization tool. (a) Node control. (b) Link control. (c) Upload control. (d) Legend. (e) Color bar with date annotations. (f) A temporal filter of the graph. (g) Graph produced from a Twitter conversation.

All the proposed visual encodings and interactions have been implemented to be used and shared via the web. The tool can be easily disseminated, so the InfoVis tool proposed has been done with HTML, CSS, and JavaScript. SVG images have been chosen to visualize the elements. These images are created based on vectors and do not lose their structure despite zooming. SVG is highly recommended for developers and designers because it allows using various types of objects in vector form as lines, figures, text, and images with great clarity and detail. Additionally, various transformations, translations, and animations can be applied to the elements created with SVG.

A javascript library called D3.js[6] is very useful for creating dynamic and interactive visualizations. It has been decided to use this library because it has great versatility for modifying elements of the DOM of a web page. D3.js is used to manipulate documents based on data, binding data to the DOM. Additionally, D3.js allows great flexibility, is efficient and fast and supports dynamic behaviors for interaction and animations. D3.js uses SVG, HTML5, and CSS, so it is convenient to use this library.

4.3.2 Graph

There is a great demand for graphs, and this work is no exception. According to Rodriguez and Neubauer [139], a graph is a universal model that helps to organize and relate entities of interest. Many structures can be represented as graphs, for example, networks of enzymes, people, the social behavior of people, and even images or texts.

According to Munzner [7], a synonym for a graph is a network within the field of visualizations. A network is a dataset that reflects relationships between two or more elements. An element in a network is called a node. The relationship between two elements is called a link. In addition, nodes can have attributes as well as links. As already seen in some state-of-the-art works in a social network, nodes can be people, and links can be a relationship if it exists between two people (following each other or being friends).

There are several design choices for organizing networks. Many papers have focused on drawing graphs (see Tamassia [140], Lisitsyn and Kasyanov [141], Hosobe [142], Tamassia [143]) and generating layouts using different aesthetic criteria. Among the most common are Node-Link Diagrams and Adjacency Matrices. As already discussed in this paper, adjacency matrices are impractical when there are many nodes, so Node-Link diagrams are the best choice for a visual representation of a network.

The Nodes of the Node-Link diagrams are represented as point marks in the visualization, and the links are drawn as line marks. Twitter conversations should be represented as directed graphs since the graph not only represents relationships but also the trajectories of the conversation. Also, there are two types of relationships, replica, and quote, which must be differentiated in the graph of the visualization tool.

Force-directed Algorithm

For the positioning of nodes and links of the JSON obtained in the previous section, it has been decided to use the language called force-direct placement. According to Murray [33] , force-direct placement places the network nodes according to a simulation of physical forces. The nodes repel each other, and the links have spring forces, which bring the nodes closer together (see an example in Fig. 4.2). Force-direct placement usually initially places the nodes at random positions. The algorithm then uses forces to move the nodes to positions where the forces stabilize, thus improving the network layout (avoiding node overlaps).

With the layout provided by the Force-direct placement, the node positions are not directly used to encode attribute values. In other words, the designer is not in charge of providing locations for the nodes since the nodes (the tweets) do not have positional

[6] `https://d3js.org/`

Figure 4.2: Simple Force Layout. Taken from Murray [33].

attributes in a plane or space. The nodes are placed in the best possible way according to their strengths, so the algorithm generates positions in a plane (x, y). It iteratively adjusts them, but it should be emphasized that these positions do not come from any attribute mapping of the tweets' JSON file. This situation may result in visual effects or artifacts that do not represent a feature of the dataset. In this type of placement, it could be possible to observe closely interconnected nodes that would form a visual group. This clustering that would form would cause a large visual effect. Sometimes these groupings are due to the dataset when there are large connections between nodes or if the forces depend on one or several parameters of the dataset. However, in other cases, these groupings could be because the nodes were pushed close. After all, they were repelled from another sector.

Due to force-direct placement, proximity can sometimes be meaningful, and at other times it can be arbitrary. According to Munzner [7], this situation "occurs in any language where the spatial position is implicitly chosen rather than deliberately used to encode information". Despite this weakness, the algorithm successfully reflects the network's structure and allows users to discover conversations on Twitter.

Force-directed Graph using D3.js

Within the D3.js library[7], many features facilitate the creation of many visualizations, such as bar charts, chords, clusters, and graphs (see Zhu [144]). Within D3.js, there is a way to use the forced-direct placement algorithm through the d3-force module. The d3-force module of D3.js allows creating a force layout with the given data and making various configurations. D3-force can use the JSON file of nodes and links to attach them to SVG elements and display them in the visualization tool. Then, the algorithm acts on the nodes and links by fitting them depending on the settings. In summary, to use the module, the designer must create a simulation for a set of nodes and links obtained in Sec. 4.2. Later, define the desired forces and renders the nodes in a visual system such as Canvas or SVG.

[7] `https://github.com/d3/d3/wiki/`

The d3-force module implements a numerical integrator called Verlet to simulate physical forces on particles. This module allows the simulation of physical particles and is suitable for studying networks and hierarchies. Additionally, the module allows the simulation of collisional disks and mesh simulation.

As its name indicates, the layout that allows creating the d3-force module is based on the forces configured with the programming. The forces you set up will allow you to position the elements in a way that would be difficult to do otherwise. The forces adjust the position and velocity of the elements to create attraction, repulsion, and collision effects. Forces can, for example, allow the following:

- The elements are attracted to one or more centers of gravity.
- Prevent elements from colliding.
- All elements repel each other.
- The elements are kept at a certain distance from each other.

The forces act on the nodes iteratively. When rendered, the nodes will start to move as the final layout is obtained. The force simulation for creating a network works with a node object and a link object. The nodes and links object are inside the same JSON file for this particular work. The nodes contain the ids of the tweets with various attributes, and the links contain the source and target along with the type of interaction between tweets (quote or reply). In general, to create a visualization with d3-force, the following steps must be followed:

1. Make a call to "forceSimulation". This function creates the simulation, and the nodes and links must be specified.
2. Add force functions. Various force functions can be configured and added to vary the behavior of the final layout.
3. Configure a callback function. This function must be configured since the positions are updated iteratively. Each of the iterations is called a "tick". D3 triggers a tick event after each iteration, and the positions (x,y) of the elements are updated in each iteration.

D3 provides several built-in force functions, although others can be defined manually (Cook [145]). The forces in the Visualization tool developed have been configured to avoid overlapping and have default values. An example of a Twitter conversation produced with d3-force is shown in Fig. 4.1g. The force functions provided by D3.js will be explained below:

1. Centering Force (forceCenter). This force is quite useful as it adjusts the center of gravity of the whole system. All elements can be centered around a central point (x, y), and elements will be prevented from flying out of sight.

2. Attraction and repulsion force between nodes (forceManyBody). This force allows elements to attract or repel each other. This force of attraction or repulsion is set with ".strength()". Elements are attracted to each other when ".strength()" is passed a positive integer value, and elements are repelled when passed a negative integer value. The strength has a default value of -30, i.e., if no default parameter is passed, the elements will be repelled from each other.

3. Overlapping force (forceCollide). This force avoids overlapping between nodes. A large number of nodes tend to overlap, and this force avoids that. The radius of action of the force is specified with the ".radius()" method.

4. Forces in X and Y axes (forceX and forceY). These forces attract the elements to a given point or points. You can use this force on all elements or apply it per element. The value of the attraction force is set with the ".strength()" method. These forces are useful, for example, when you want to visualize a disconnected graph. In a disconnected graph, if only the ".forceCenter" force is used, the separate subgraphs will jump out of the display window. However, with forceX and forceY this behavior is avoided.

5. Link force (forceLink). This force allows a fixed distance to existing between connected elements. In other words, This function helps to keep a constant distance between connected nodes. The developer must specify the source and destination of the links in an array to use forceLink.

4.3.3 Legend

In addition to rendering the graphs with nodes and links, arrows have been placed on the links to show the direction of the Twitter conversations (see Fig. 4.1g). Two distinct colors have been used to identify reply type interactions from Quote type interactions. As seen in Fig. 4.1d, the orange color encodes that the type of interaction is a reply, the blue color encodes the quote type interaction, and the arrows encode the direction of the interactions. The reason for using these two colors is that the two interactions can be differentiated and identified at a glance. As already seen in section 2.5.2, an excellent channel for categorical attributes is the Color Hue, so two different colors encode the two categories present in Twitter conversations (Replies and Quotes). Finally, this encoding to differentiate reply from Quote allows users to differentiate and visualize the relationships between Tweets [R1].

4.3.4 Color Bar and Color Nodes

The color bar is the element that helps to visualize the temporality of the Tweets (nodes) in the graph (see Fig. 4.1e). The time within a Twitter conversation can be viewed linearly. Tweets in a conversation starter at a date, and the conversation expands or not over time. Because time is linear, a linear color coding scale has been chosen. In this type of scale, the color evolves linearly from one intensity to another. In this case, the color saturation is the channel to encode the time. This is a very good channel for this case because the

time has an order. With this encoding, it will be possible to express how old or recent a tweet is by how intense the color is. To make the nodes easily distinguishable from the links, we have chosen a green color that contrasts with the colors of the links.

The color of the nodes reflects the temporality (see Fig. 4.1e and 4.1g), where more intense colors reflect older dates and less intense nodes reflect more recent dates [R2]. The color bar (see Fig. 4.1e) indicates the dataset's oldest and most recent dates. Each time a different dataset is used, these dates change according to the dataset. The colors of the nodes, together with the directed graph and the colors differentiating the interaction types, satisfy [R3].

To create the color bar, two scalers, a linear and a time scaler, are necessary. With these scalers, it is possible to generate a color bar that represents the time range and has the annotations according to the dates of the dataset.

Scale functions

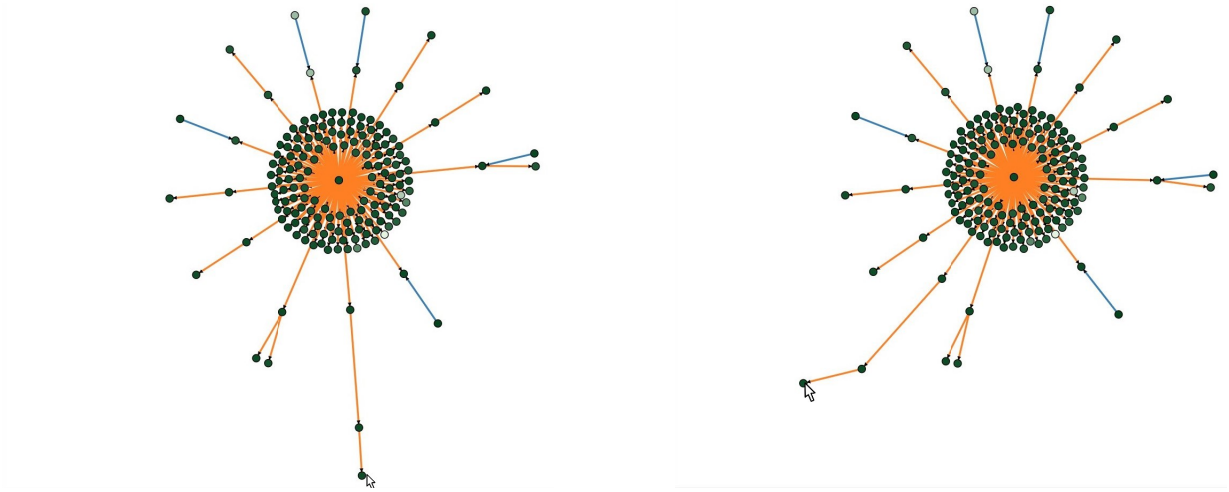
Scales "are functions that map from an input domain to an output range"[8]. According to Murray [33], it is difficult for the values of a dataset to correspond to pixel measurements for use in visualization. This is the reason for working with scales, as these functions allow data values to be mapped to new values useful for visualization. Within D3.js, different types of scales are used according to the need. The D3.js scaling functions can be configured with parameters. Once the scaling functions are configured, the scaling function is called with a value, and the function returns a scaled output value. "A scale is a mathematical relationship, with no direct visual output" (Murray [33]). The scale can have a visual representation, either an axis with markings indicating the progression of values or a color bar.

Scales in D3.js must be defined in an input domain and an output range. The scale domain must contain the dataset's minimum and maximum values. The range of a scale is the range of possible output values. The output range depends on the developer. For example, these values can be displayed in pixel units or even colors. What happens within a scaling is normalization. Normalization assigns a numerical value between 0 and 1, depending on the minimum and maximum values. Scaling in D3.js is done quickly. The input value is normalized according to the domain, and the normalized value is scaled to the output range. The two types of scalers used in this work are explained below:

1. Linear scaler (d3.scaleLinear). This scaler is used to create the color bar and to color the nodes depending on their date. This scaler takes the oldest date as the minimum and the most recent date as the maximum of the dataset. This scaler takes as output range the most intense color as the minimum and the least intense color as the maximum. Thus, the scaler shows the oldest dates with a more intense color and the most recent dates with less intense color.

2. Time scaler (d3.scaleTime). This scaler is similar to the linear scaling, but the time scaling can operate on JavaScript Date objects. This scaler takes the oldest date as a minimum and the most recent date as a maximum. The difference is that this scaler takes in-range pixel distances to be able to plot an axis with the dates. Without this

[8] `https://github.com/d3/d3/wiki/`



(a) Graph before the drag-and-drop.

(b) Graph after the drag-and-drop.

Figure 4.3: Drag-and-drop interaction.

scaler, the axis annotations in the color bar (see Fig.4.1e) would not be displayed in date format.

4.3.5 Interaction Techniques

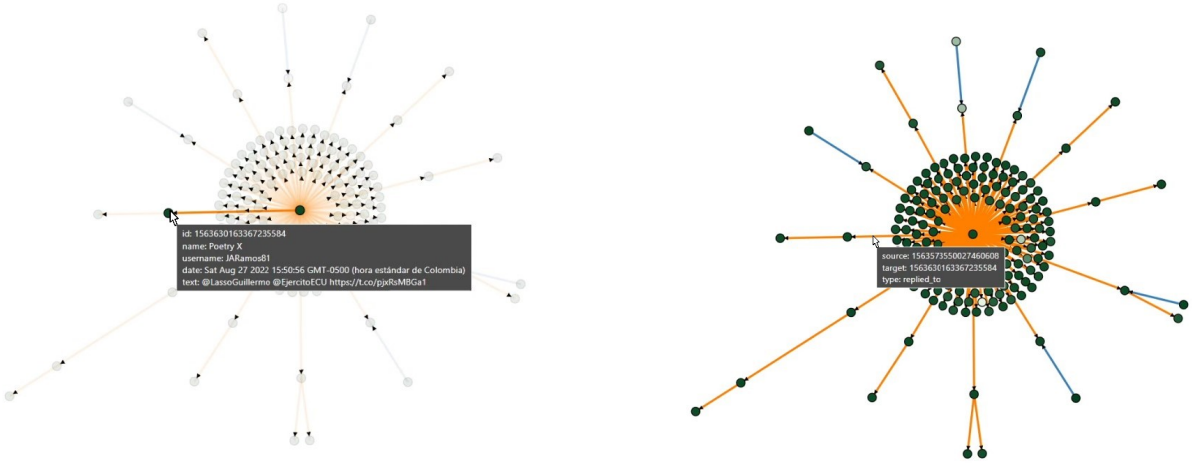
As already discussed in the theoretical fundamentals section, interactions are important to facilitate the use of the visualization tool. They allow complete data exploration and can help reduce visual complexity. Interactions allow easier exploration and improve the understanding of the information being visualized. With graphs, the interactions implemented in this work help the user, especially when the visualized graphs are very complex and have many nodes involved in a Twitter conversation. The following explains the interaction techniques [R4] implemented in the visualization tool.

Drag-and-Drop

A drag-and-drop interaction on the nodes has been implemented to facilitate the exploration of the nodes. This functionality is implemented with D3.js. Drag-and-drop allows the nodes to be grabbed with the mouse and moved in the desired direction. Then when the mouse is released, the nodes can be dropped in any desired position (see Fig. 4.3).

Node and Link Hover

Nodes and Links contain important attributes that are part of the conversation, and visualizing them allows a better understanding of the network shown with a graph. Nodes represent the Tweets in the visualization. Nodes have attributes such as the name of the user, their creation date, the tweet id, and, if desired, the text of the tweet (which could be used for sentiment analysis, but this is not addressed). Links have the id of the source tweet, the id of the destination tweet, and the type of interaction (reply or quote).



(a) Node hover.

(b) Link hover.

Figure 4.4: Hover Interaction.

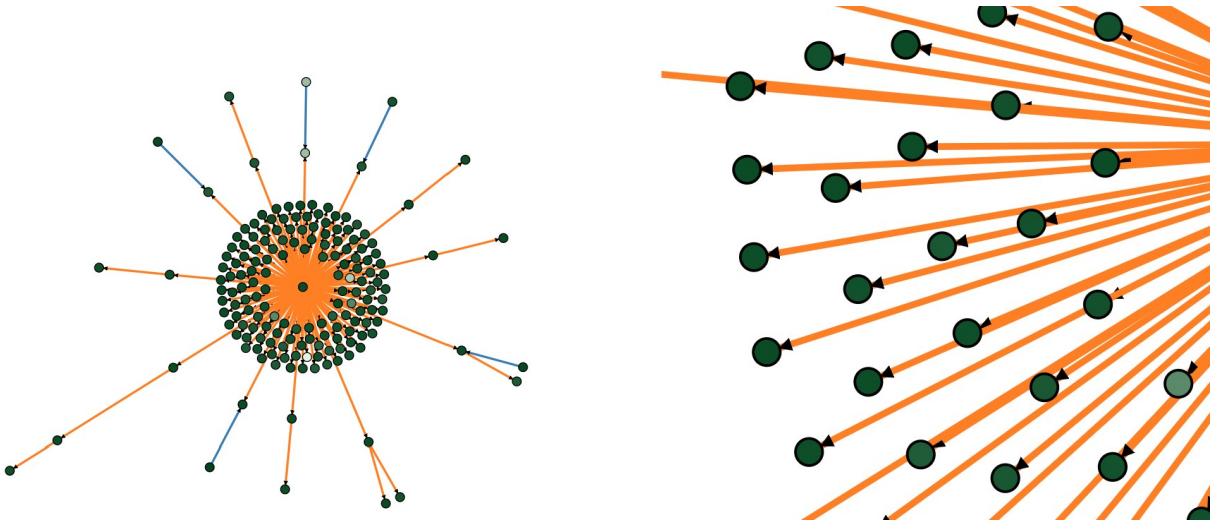
The information contained in the links and nodes could be presented as text next to the nodes and links, but it would cause a lot of visual clutter, visual overload, and occlusions. Therefore, it is decided to display the information when a user's mouse cursor is located over an element (node or link) without selecting it. Once the pointer is located over the element, a window with the element information is displayed. Then when the pointer is moved to a different location, the previously displayed information disappears (see Fig. 4.4).

Semantic Zoom

The layout obtained by the forced-direct algorithm allows to repel nodes from each other so that they do not overlap, but there may be areas with a large number of nodes. These zones do not allow for differentiating well the nodes from each other. Therefore, it has been decided to implement a Semantic Zoom. The Semantic Zoom is implemented with javascript and D3.js. It allows visualizing the objects with more detail when zooming in with the camera. At the same time, it allows nodes to be separated when zooming in to differentiate them (see Fig. 4.5).

Highlighting

The graphs rendered in the tool can have many paths. A large number of paths can cause visual overload. The highlighting interaction has been configured to allow the user to focus on a single path. The highlighting shows the path from the tweet that originates a conversation. The user can hover over any node, and the path from the originating node will be highlighted (see Fig. 4.6a). When the user moves the mouse pointer elsewhere, the highlight will disappear. In case there is no path from the origin node to the node over which the mouse is positioned, only the neighboring nodes will be highlighted (see Fig. 4.6b).



(a) The network is in an enlarged view.

(b) The network is in a close-up view.

Figure 4.5: Semantic Zoom Interaction.

(a) Highlighting a node that has a path from the origin.

(b) Highlighting a node that does not has a path from the origin.

Figure 4.6: Hover Interaction.

Node Controls

A panel has been created with HTML and JavaScript to configure the parameters of the nodes in the displayed graph (see Fig. 4.1a). As mentioned before, several forces can be configured in the positioning algorithm. The force "d3.forceCollide" has been chosen to prevent circular objects (nodes) from colliding. The node control panel has a checkbox to enable or disable this force.

This panel contains two radius sliders. The first slider controls the collision strength of the nodes. In the D3.js reference API[9], it is indicated that the range of this force is between zero and two. If not specified, the default strength is set to one. The overlapping nodes are iteratively separated depending on the force's strength.

The second slider controls the radius of the nodes and the radius on which the force "d3.forceCollide"" acts. The range of this radius is between 0 and 100. Its default value in the tool is 5.

Link Controls

This panel is also created to manage parameters of the forced direct graph links (see Fig. 4.1b). In this panel, "d3.forceLink" is used. The checkbox in this control section is used to enable or disable the links in the graph. If they are disabled, there will be no links connecting the network nodes. With the slider in this section, you can control the distance of the links. The default value in the tool is 30.

Charges Controls

In the third control panel, the force "d3.forceManyBody" is used (see Fig. 4.1c). In this area, there is a checkbox to activate or deactivate this force. In other words, the force of attraction or repulsion between nodes becomes zero. With the first slider, you control the force of attraction or repulsion.

The second slider controls the minimum distance between the nodes on which the force "d3.forceManyBody" is considered. This parameter prevents an infinitely strong force if two nodes coincide on the same slider. The default value of the minimum distance is 1 in the tool.

The third slider controls the maximum distance between the nodes over which the force "d3.forceManyBody" is considered. Specifying a finite maximum distance improves performance and produces a more localized design. The default value of the minimum distance is 2,000 in the tool.

Temporal Filter Slider

The Temporal filter slider (see Fig. 4.1f) allows displaying of the links and nodes with a creation date less than or equal to the time indicated by the slider (see Fig. 4.7). This filter allows users to see the evolution of the connection up to a given time. The slider can be useful to explore how a conversation increases in complexity over time. It is also

[9] `https://github.com/d3/d3/wiki/`

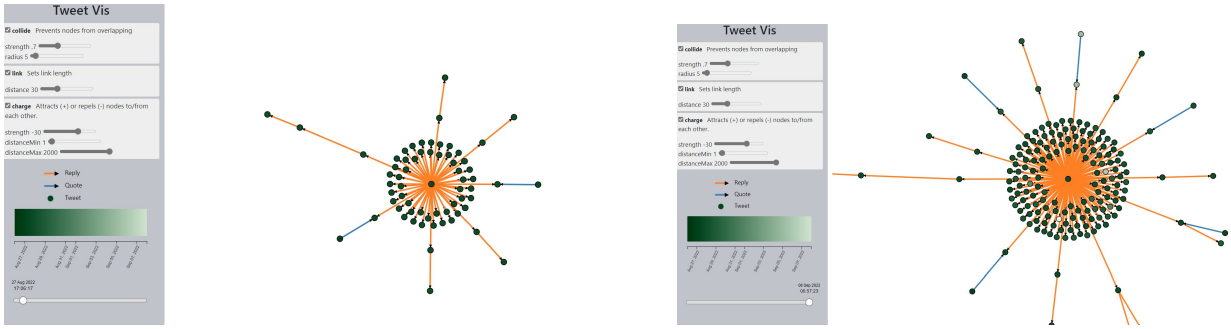


Figure 4.7: Graphs at different dates.

useful to visualize whether or not most of the interactions occur in the first moments of the conversation.

4.4 Validation

All four levels of Munzner's Nested model were addressed in this work. The characterization of the problem domain was approached with careful investigation within the specific domain of tweet visualization. The visualization tool's requirements are framed within the tool's exploration, discovery, and integration for other tasks. The data abstraction and operations design was addressed by understanding that most tweets do not have a latitude and longitude, so the JSON file that allows getting the Twitter API can become a dataset suitable for node-link graphs visualization. The third level of Munzner's nested model was extensively treated with the proposed choices for encoding the information and interactions raised. Finally, a positioning algorithm has been used along with other implementations to make the interactions work at the innermost level, such as implementing path search for path highlighting.

Different articles were studied to validate the first level of the nested model to demonstrate how the target audience benefits from visualization tools. A detailed list of the requirements in terms of the domain and data discussed at the abstraction level is included. At the Idiom and interactions validation, an immediate validation was performed, justifying the design of the languages and interactions concerning cognitive and perceptual principles. In addition, at this same level, it is validated as a result of the analysis of other visualization tools, their encodings, interactions, and the common problems they may have.

Chapter 5

Results and Discussion

This section exposes four different Twitter conversations using the InfoVis tool developed in this work. Each conversation is analyzed and explored, and the tool is discussed. The datasets are extracted using the procedure described in the previous section, and the resulting JSON files are used in the visualization tool. The conversations in this section have been selected because they have had some importance in social and political environments.

5.1 Guillermo Lasso Tweet

Guillermo Lasso, president of the Republic of Ecuador, posted a Tweet informing that Maria Belen Bernal was found on September 21, 2022. The Tweet is shown in Fig. 5.1. The Tweet that generates a conversation has the following id: 1572675339347955713. The date for obtaining the conversation is September 11, 2022. This Tweet is given due to a case of femicide of the citizen María Belén Bernal. The citizen disappeared in a Police High School in Ecuador, and her husband, Germán Cáceres (former lieutenant of the Ecuadorian police), is the main suspect.

Figure 5.1: Guillermo Lasso Tweet (id: 1572675339347955713)

The graph obtained from the generated conversation is shown in Fig. 5.2. Apparently, the Tweet in Fig. 5.1 has 4,992 replies. However, the number of nodes after processing the

data provided by the Twitter API is higher. After processing the data with python and filling in attributes of some missing Tweets, 6,748 nodes and 6,755 links are obtained.

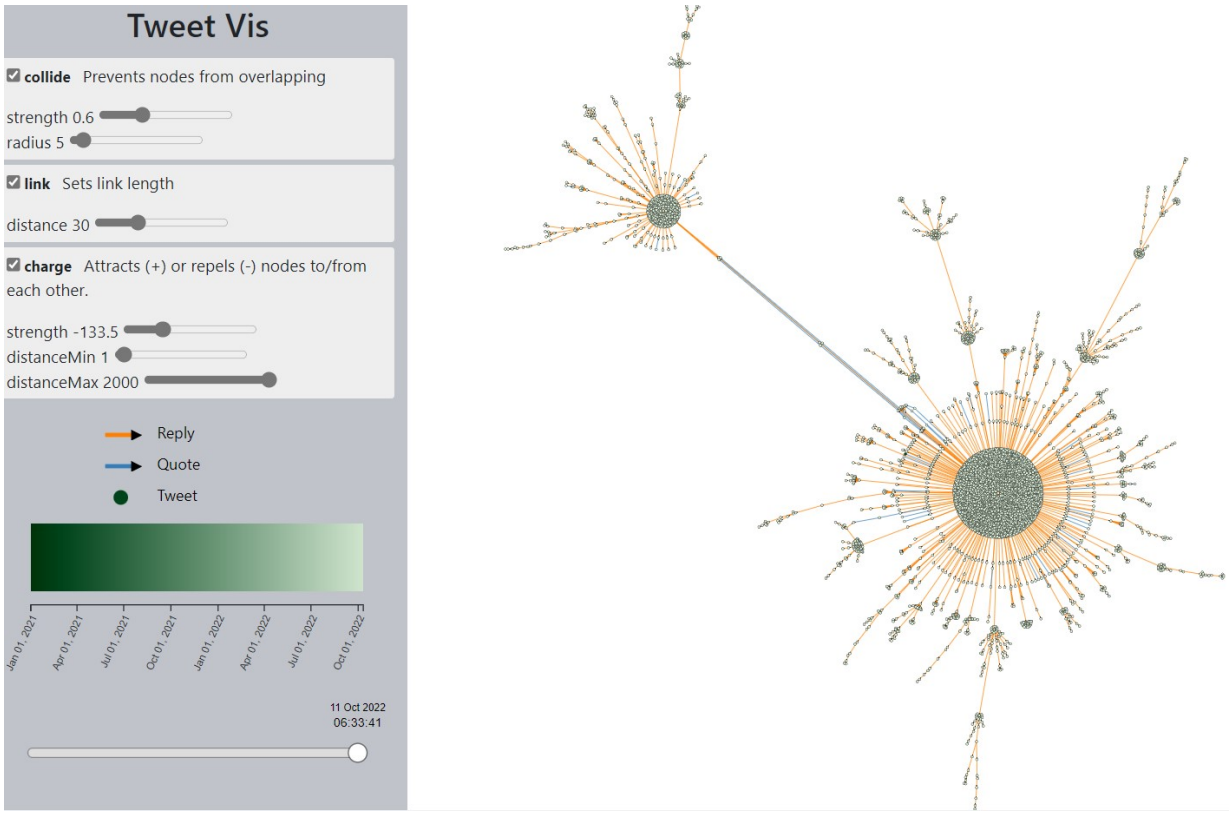


Figure 5.2: Result of the conversation visualization.

In Fig. 5.2, we waited for the algorithm to act interactively to position the elements. Drag-and-drop has been used to accommodate some paths. Also, some forces have been relaxed so that the nodes do not move too far from their origins. In this figure two, well-marked clusters can be seen at first sight. These clusters show a strong influence of two Tweets. The largest cluster originates from Guillermo Lasso's Tweet 5.1. The second tweet that forms the second largest cluster is produced by a response from Maria Belen Bernal's mother. Due to the social context of Ecuador due to the femicide, this Tweet receives the most attention after Lasso's Tweet. It should be noted that the Tweet chosen for the reconstruction of the conversation is already highly influential, so it has many nodes in the visualization.

A closer view of the Lasso tweet (see Fig. 5.3) allows users to appreciate more details. One of the most notable details is the color of the nodes (tweets). In this case, you can see that there is no major variation in the color of the nodes. A look at Barcode (see Fig. 5.4) indicates that the oldest Tweet belongs to January 1st, 2021, far from when Lasso's Tweet was posted. This indicates that a Tweet extends much further back in time than the rest of the Tweets.

The January 2021 Tweet is from a user named "jared jaguar". This Tweet seen in Fig. 5.5 has no relation with Guillermo Lasso's conversation. There is another node that is a neighbor to the latter. The tweet is from the author "Hablaxvo". This last Tweet replies to Lasso's tweet and quotes the Tweet in Fig. 5.5. The Tweet in Fig. 5.6 ("Hablaxvo" user)

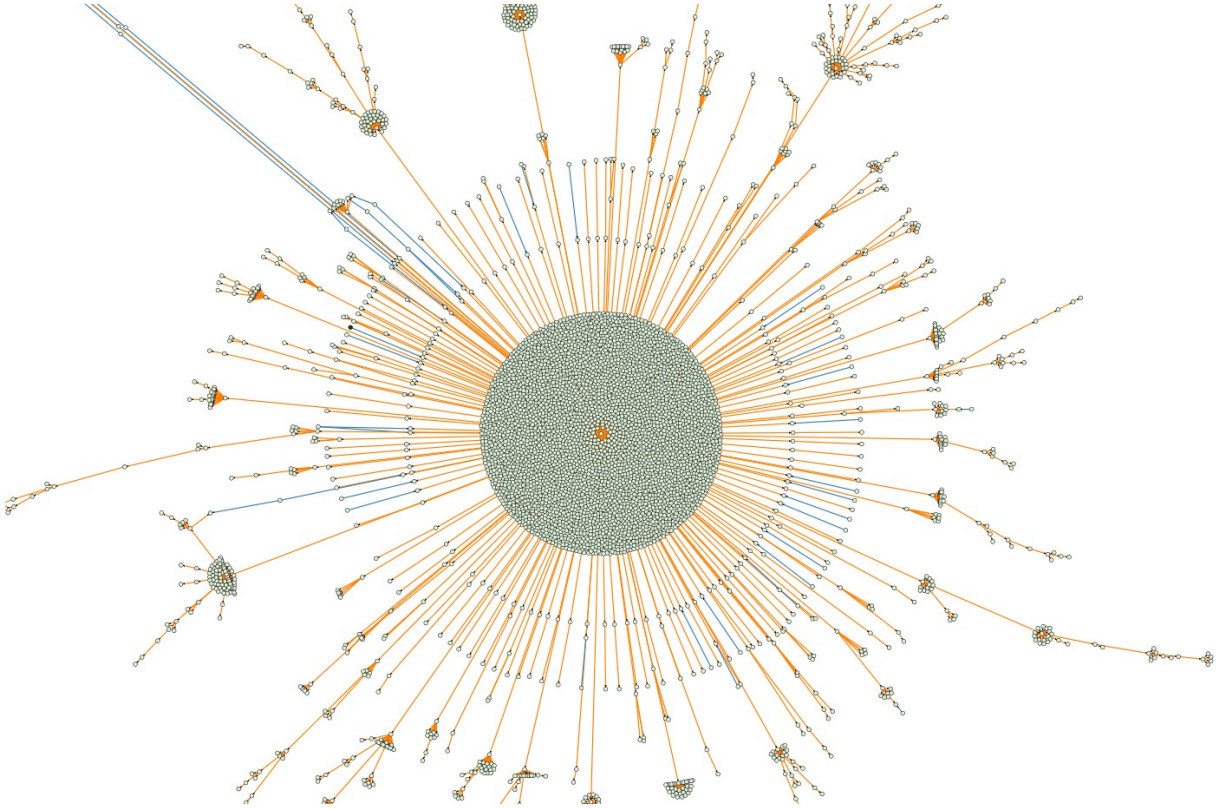


Figure 5.3: Cluster produced by Lasso's Tweet5.1.

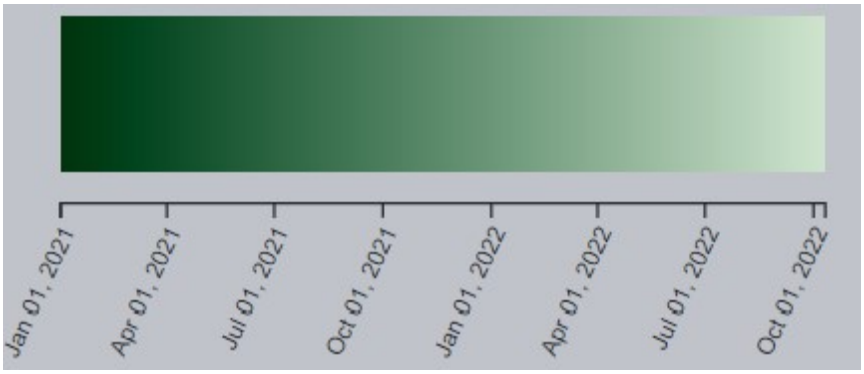


Figure 5.4: Barcode of Fig. 5.1.

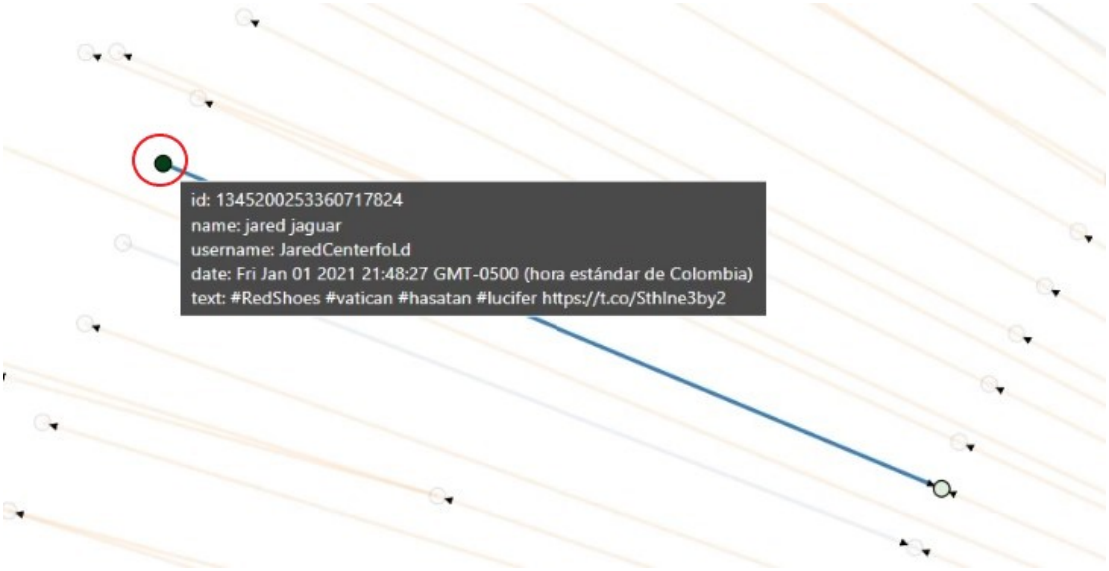


Figure 5.5: Node with the oldest date.

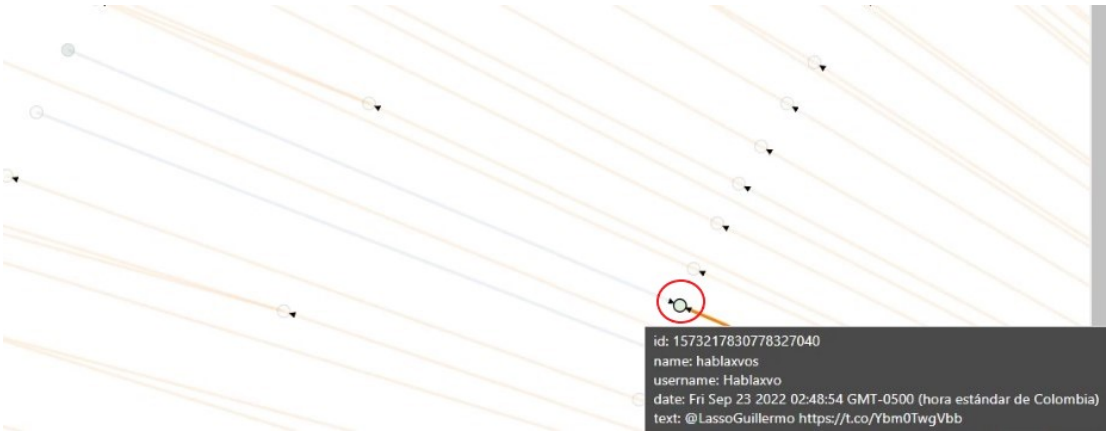


Figure 5.6: Node quoting the oldest Tweet and replying to Lasso's Tweet (Fig. 5.1).

does not contain any text or provide any information. The Tweet contains only a link that is produced by quoting another Tweet. The displayed behavior may indicate that Tweet (5.6) of the "Hablaxvo" user is a product of a Boot.

The behavior of tweets seen is found in many other nodes (see Fig. 5.7). They are quickly visualized as they are nodes that have a Retweet link (orange color) and a Quote link connected (blue color). There is a quite repeated pattern in which a Tweet replies to the Tweet of Guillermo Lasso while quoting another Tweet. Most of these Tweets seem to be from real users, but certain accounts may be suspected of being bots due to their usernames: 'VaPorTi_Ecuador,' 'Ecuadorprogres1', 'EcuadorNoAguantaMas.'

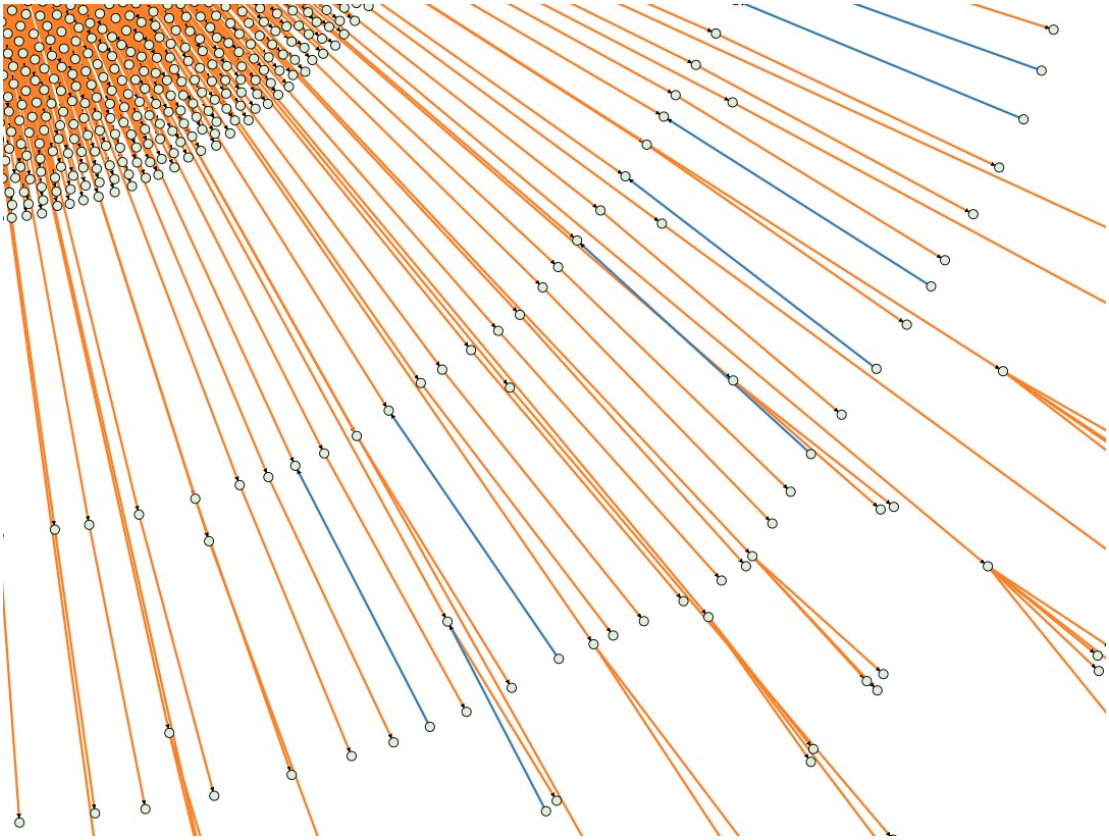


Figure 5.7: Zoom of the largest cluster in Fig. 5.1

The Temporal slider in Fig. 5.2 also indicates the date of the newest tweet. The most recent tweet is somewhat distant from Lasso's tweet post (Fig. 5.1). The date of the most recent tweet in the network is October 11, 2022. Thanks to the Temporal Slider, it can be determined that most interactions after posting a Tweet occur immediately. After a short time window, a Tweet's interactions decrease abruptly, but they still exist. In the case of Lasso's Tweet (Fig. 5.1), there are still interactions up to the date the conversation was obtained through the Twitter API. Since the Tweet seen in Fig. 5.1 was posted, most of the nodes in the conversation originated in the subsequent three days.

Finally, Fig. 5.2 shows that there are not very long trajectories in the conversation. The conversation's trajectory expands with many tweets, but these do not form paths of more than 11 Tweets. There may be many replicas of replicas, but the paths formed in this conversation are not very long.

5.2 LaHistoria Tweet

The Twitter account called LaHistoria, in its Twitter description, states that it is a news and media agency from Ecuador. This account published the Tweet of Fig. 5.8 at 11 a.m. on October 13, 2022, with a Twitter ID of 1580592828168534384 68. The Tweet in Fig. 5.8 does not have many interactions, and it can be seen that it has only 129 replies.

The generated twitter conversation was obtained at 7:20 a.m. on October 14, 2022. The conversation was obtained just twenty hours after the tweet was published. The news is not viral, the account is not famous, and the time to obtain the conversation is immediate, so we do not obtain a large number of nodes and links. Fig. 5.9 shows the result of the visualization tool. In the visualization graph, there are 130 nodes and 129 links.

Figure 5.8: LaHistoria tweet (id: 1580592828168534384 68).

There are no quote links in Fig. 5.9; only reply links are found. The resulting network paths are quite short. The largest path has four connected nodes. In contrast to the previous conversation studied, this graph shows more variety in the intensity of the node colors. One can easily distinguish the oldest nodes from the most recent ones. In this case, the nodes and links appear very progressive by sliding the Temporal Slider. The Temporal Slider has the advantage of filtering the nodes and links by hours, so it is easy to see how the conversation is increasing over the hours since the publication of the first Tweet. In this case, no nodes are too distant in time from the other tweets, so it is easier to appreciate the temporality in the intensity of the nodes.

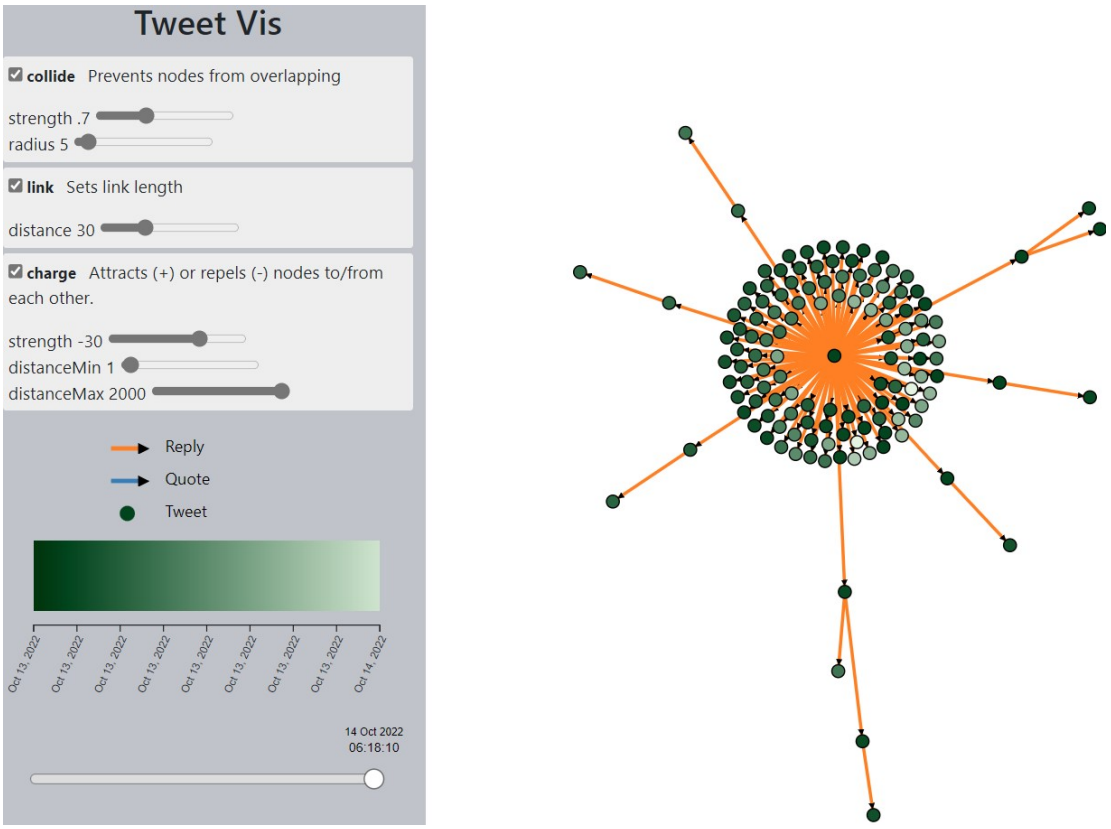


Figure 5.9: LaHistoria graph result.

5.3 The Royal Family tweet

The British Royal Family Twitter published the Tweet of Fig. 1 on September 8, 2022, with id: 1567928275913121792. This Tweet reports that former Queen Elizabeth II died. This Tweet has many interactions, as seen in Fig. 5.10. Approximately there are 88,8 thousand replies, a massive amount of tweets participating in the generated conversation.

The InfoVis tool developed is not focused on handling massive amounts of data. Therefore a sample of Tweets related to this Tweet is obtained. The collection of the Twitter conversation was performed on September 12, 2022. The code is configured so that approximately 10,000 tweets are obtained.

Figure 5.10: The Royal Family tweet (id: 1567928275913121792).

From the conversation from the Tweet in Fig. 5.10, 10,620 nodes and 9,874 links are obtained. The initial result of the visualization can be seen in Fig. 5.11 . The dataset obtained is approximately 12% of the number of replicas shown in Fig. 5.10. After the nodes are loaded, it is necessary to wait until the nodes are interactively placed in the most appropriate positions according to the forces configured in the left side panel of the tool.

Once the elements are arranged, they can be dragged, and the forces can be modified to make the visualization more understandable. After rearranging the resulting graph with the interactions, the graph in Fig. 5.12 is obtained. It should be noted that the tool is no longer as fluid as it was with other conversations. The InfoVis tool is still manageable, and its interactions work for filtering and rearranging nodes.

This time most of the nodes appear with the same intensity, and this is because there is a node that is very old as the other nodes. The oldest node is dated April 21, 2019, and it is very easy to find it with zoom and the ability to navigate through the visualization that integrates the tool. Figure 5.13 shows this node.

Fig. 5.12 shows that there are many isolated nodes. The most noticeable cluster in the center is the Tweet about the death of the ex-queen and many other nodes floating near

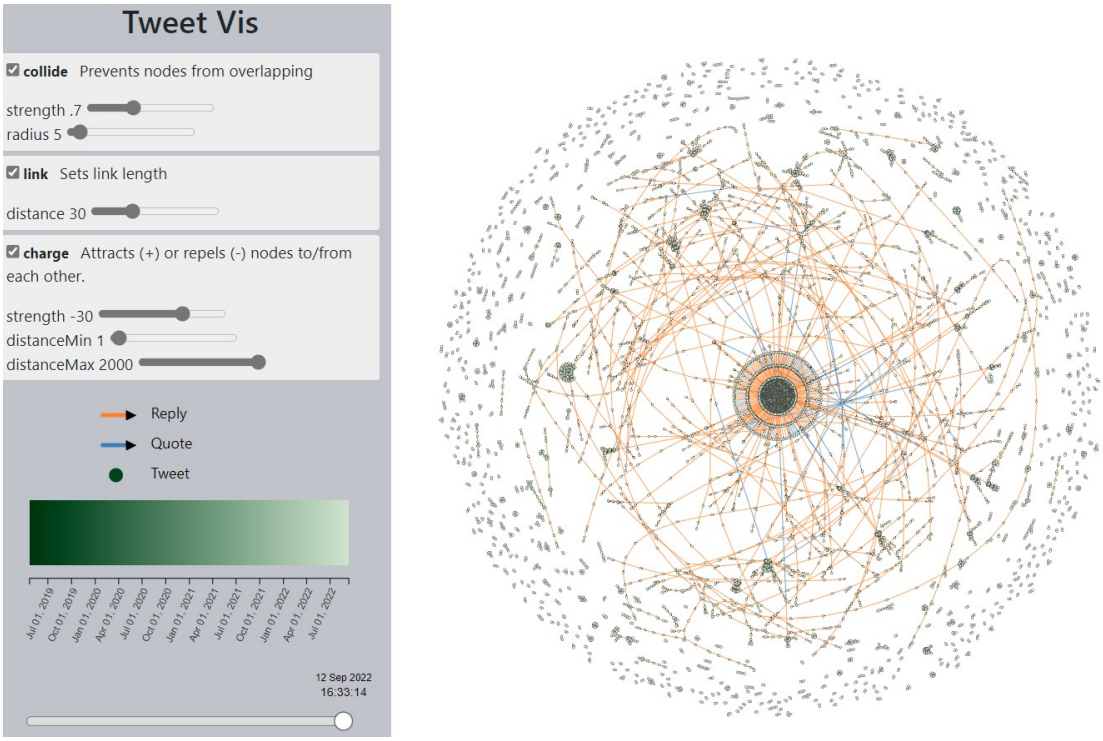


Figure 5.11: The Royal Family conversation.

and far. The isolated nodes exist because only a sample was taken from the total number of tweets participating in the conversation. These isolated tweets look like this apparently because there is no interaction linking them to the larger cluster, but this is not the case. If the API were to take all the conversation nodes, no isolated nodes would be visualized. By getting only a sample of tweets, all the nodes that would form the connections between all the visualized nodes are not obtained.

Unlike other conversations visualized in Fig. 5.12, there is a massive amount of nodes, and this is due to the big influence of "The Royal Family" account. As visualized as there are many interactions, the trajectory that tweets can take can be very long. Long paths are long chains of users responding to each other. Fig. 5.14 shows close up to improve the detail of the visualization.

In Fig. 5.14, the structure of the graph has remarkable characteristics. For example, finding more than thirty tweet trajectories of conversations is possible. In the right part of the main cluster (See Fig. 5.15) is shown some peculiar interactions of the quote type (blue color). The images in Fig. 5.16 show some remarkable features of the structure of the quote-type interactions formed. The user of the central node in Fig. 5.16a is called "aracuantrus". This user writes a text nothing related to the death of Queen Elizabeth II. All the tweets the user makes are the same as those in Fig. 5.16b. He writes about doubling people's monetary earnings. The tweets that serve as a connection between the user "aracuantrus" and other tweets, are generated when the user "aracuantrus" replies to his tweet (which is in the center) and replies to other tweets from other users (see Fig. 5.16c). In short, the user "aracuantrus" enters a viral conversation and starts replying to other tweets indiscriminately offering more monetary income. This event may be a strategy to attract followers to a possible pyramid group or a bot trying to have more reach and

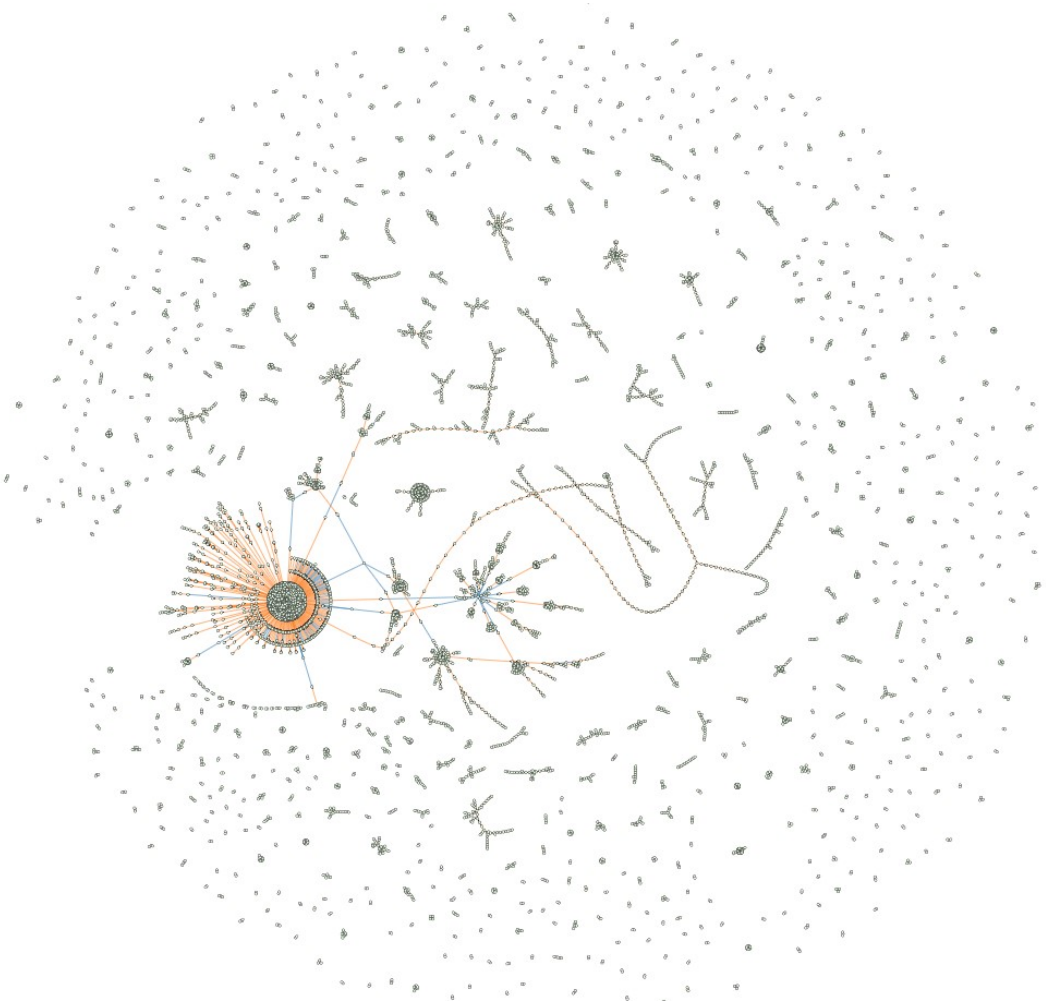


Figure 5.12: The Royal Family conversation after reordering the nodes.

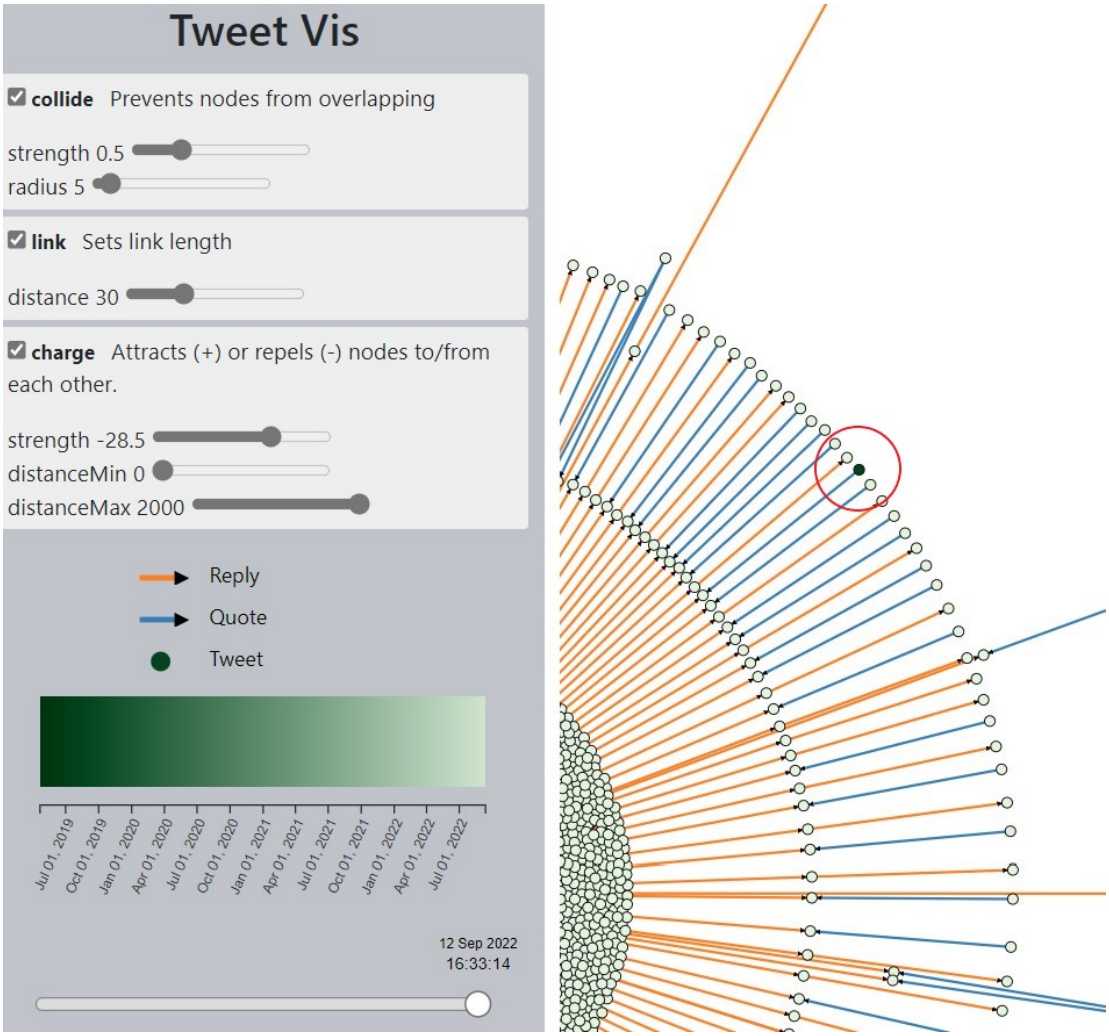


Figure 5.13: Old node modifying the color intensity of the other nodes.

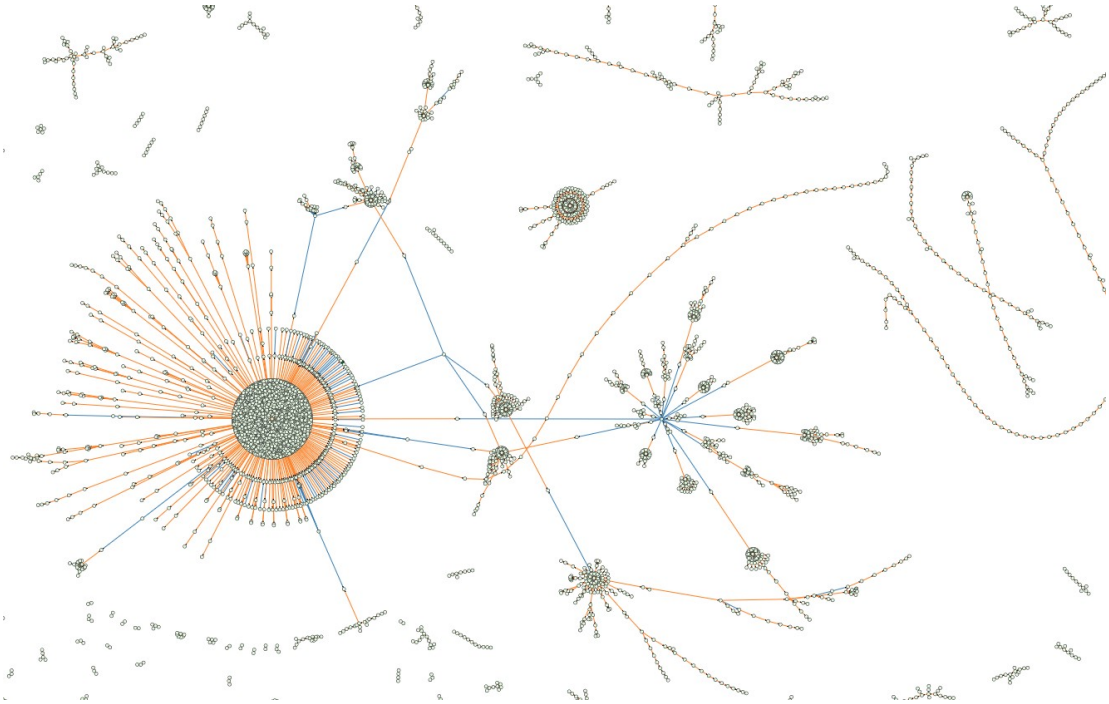


Figure 5.14: Zoom of the graph in Fig. 5.12.

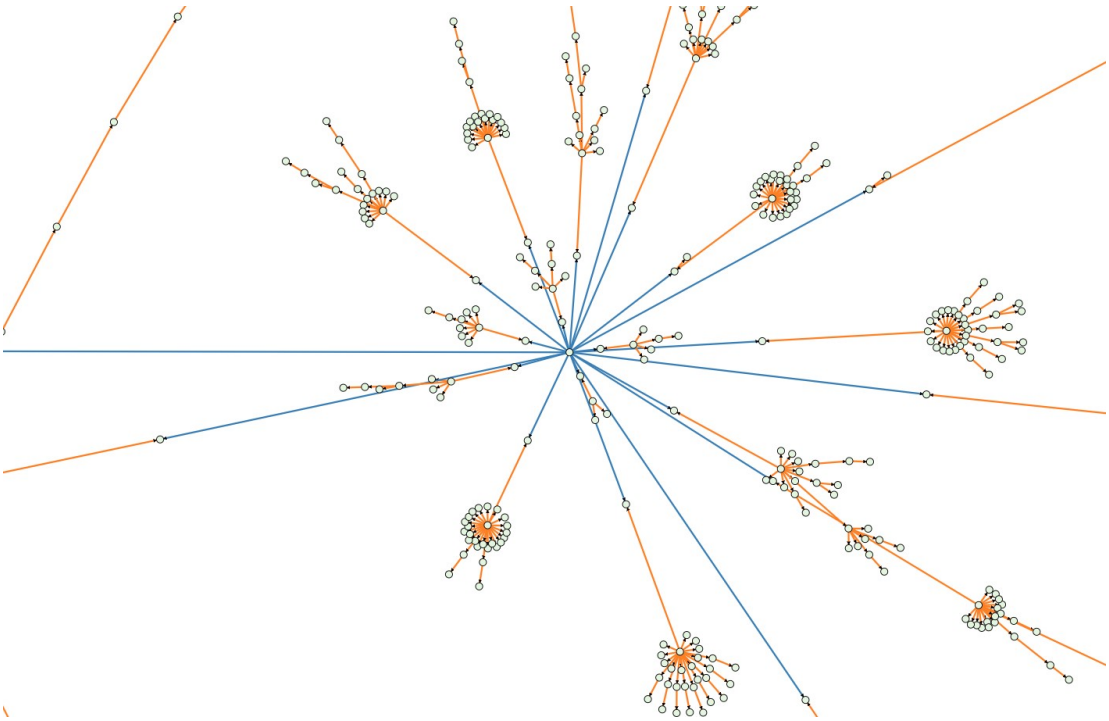


Figure 5.15: Closer look at the graph in Fig. 5.12.

be more influential. Finally, this example shows how an account with certain tweets can be invasive, cause disorder in the conversation, or take advantage of this viral event to get more views or followers.

5.4 Joe Biden Tweet

This time we will visualize the conversation obtained by a tweet from the president of the United States, Joe Biden. The Tweet that produces the analyzed conversation is shown in Fig. 5.17. The Tweet was published on October 14, 2022, and has approximately 22 thousand replies.

After using the tweet id (1581049730565832705), 31,83 nodes (tweets) and 31,921 links (interactions) are obtained. The result of the visualization of the tweet trajectory is shown in Fig. 5.18. This conversation has a large number of nodes which causes problems in the visualization tool. The main problem when visualizing a conversation with this amount of nodes and links is that the interactions start to run slowly. When displaying the nodes, the application suffers when rendering all the visual elements. Additionally, the nodes and links take a long time to be positioned according to the forces of the Forced direct algorithm.

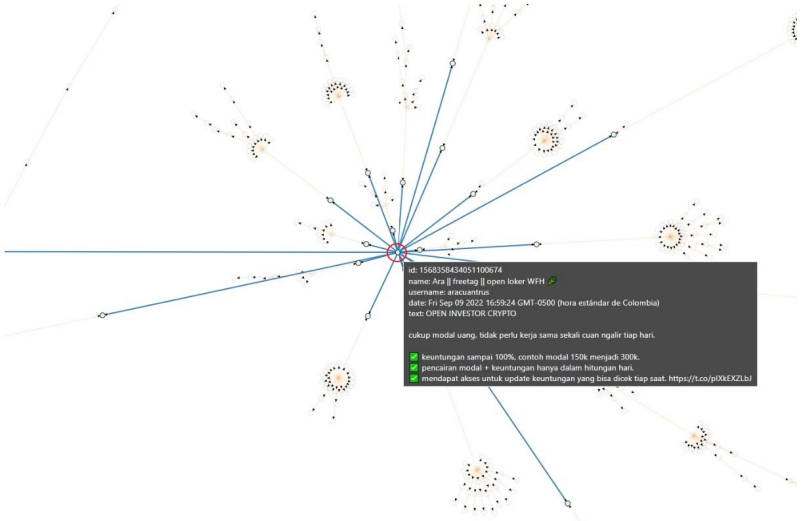
With many elements, the interactions configured are not immediate since the user modifies some parameters of the left side panel. The zoom is laggy, and navigation becomes very difficult. Although the response in all types of interactions is not immediate, they can still be used.

Fig. 5.19 shows that the highlighting interaction to show the tweet paths works, but tasks such as zooming, navigating the graph (when zooming), dragging and dropping with the nodes, and sidebar slicers, are almost impossible to perform because of the very slow response of the tool. Also, using the task manager tool in Windows to see the amount of ram used indicates that rendering this large amount of elements can be a very heavy task for any browser.

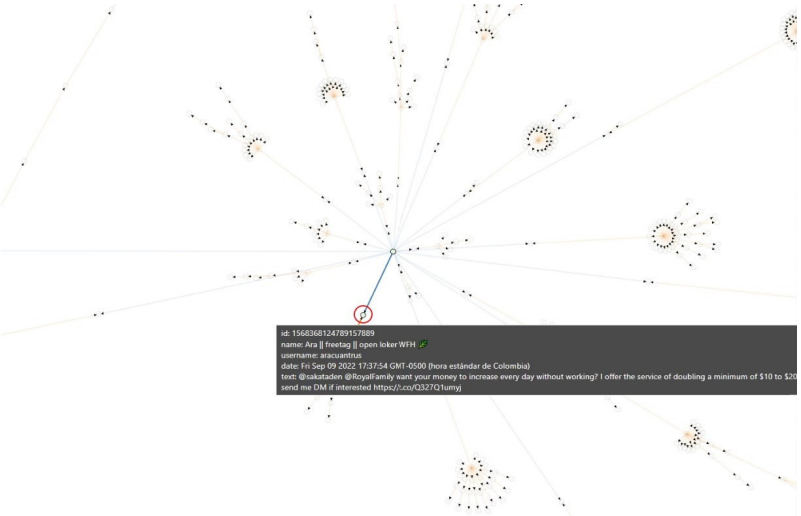
After modifying the force with which the nodes are repelled, it is necessary to wait for them to adjust their positions according to the force. As mentioned, the delay until the final position is achieved is evident because the positioning algorithm works iteratively. Fig. 5.20 shows that the trajectories are better distributed than in Fig. 5.18.

In order to find the number of nodes where the InfoVis tool starts to suffer serious performance problems, it has been decided to reduce the dataset size. With 20,446 nodes and 20,106 links, the result seen in Fig. 5.21 is obtained, but still, the tool has problems handling this amount of rendered elements. Finally, it is decided to reduce the number of nodes to 15,557 and 15,155 links, and the visualization of Fig. 5.22 is obtained, this time the interactions work in a more manageable and fluid way, but the tool still has difficulties with handling this amount of elements rendered by the web browser.

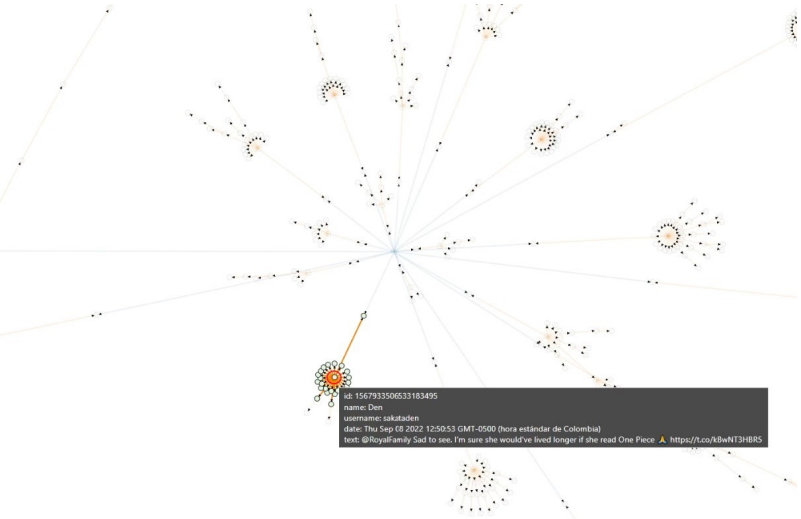
Zooming in on the graph in Fig. 5.22 reveals some remarkable interactions. These interactions are visualized in Fig. 5.23. These blue-colored interactions are of the quote type and occur because several users respond to Joe Biden's tweet by quoting a tweet that seems important. Fig. 5.24 shows more details about the tweet being quoted several times.



(a) Main tweet of user “aracuantrus”.



(b) Tweet quoting and replying made by the user “aracuantrus”.



(c) Tweet that is part of the conversation about Queen Elizabeth II.

Figure 5.16: Behavior of a user who apparently could be a troll.

Figure 5.17: Joe Biden tweet (id: 1581049730565832705).

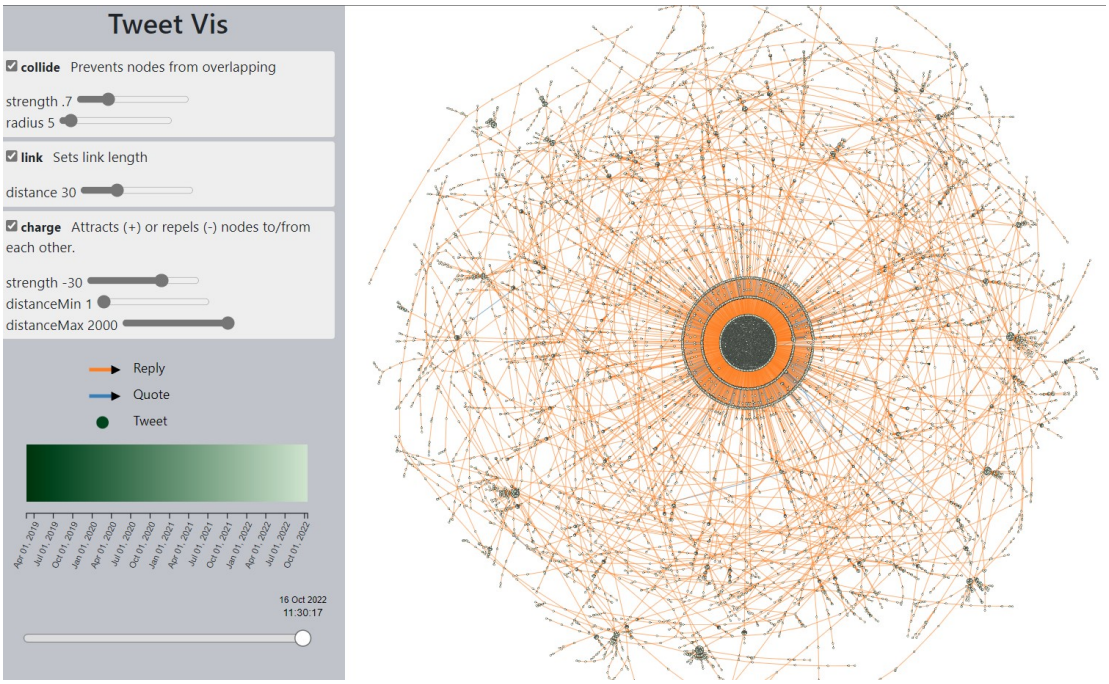


Figure 5.18: Joe Biden graph result with all tweets in the conversation.

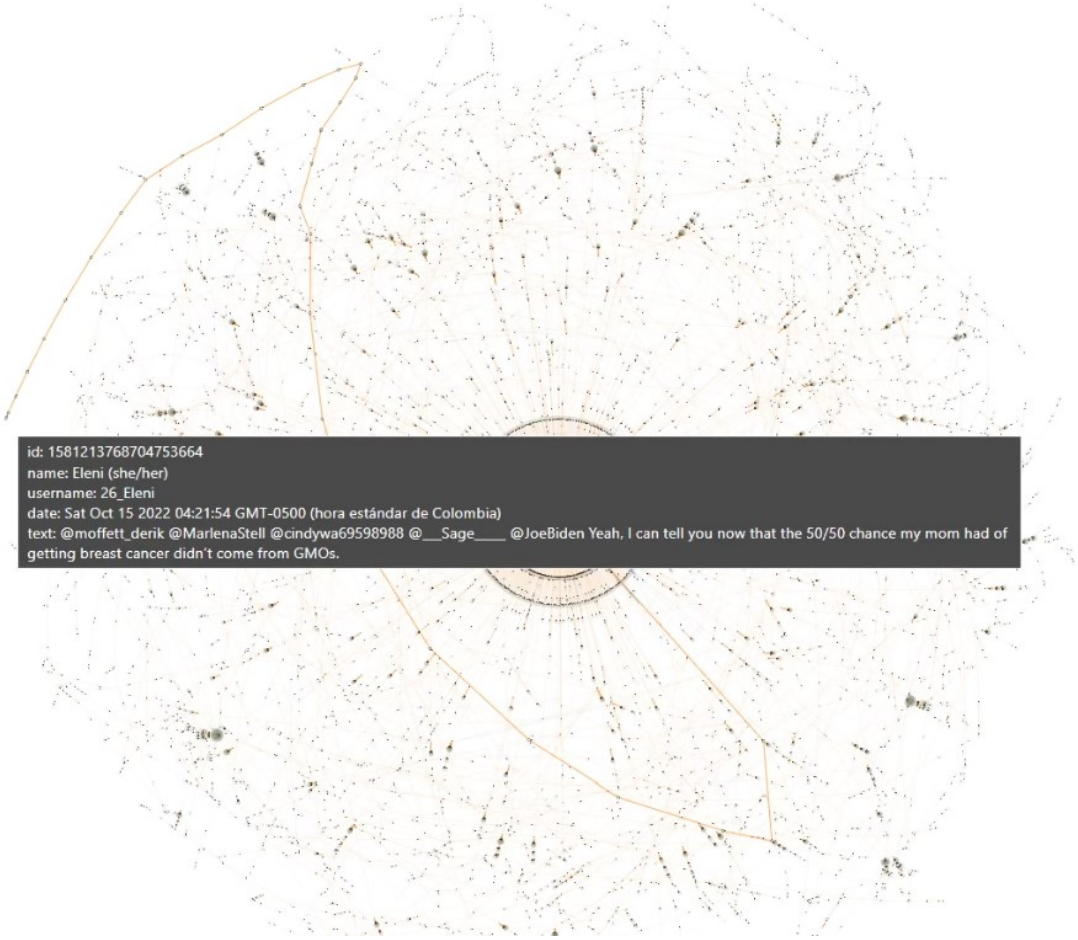


Figure 5.19: Graph result with one path highlighted

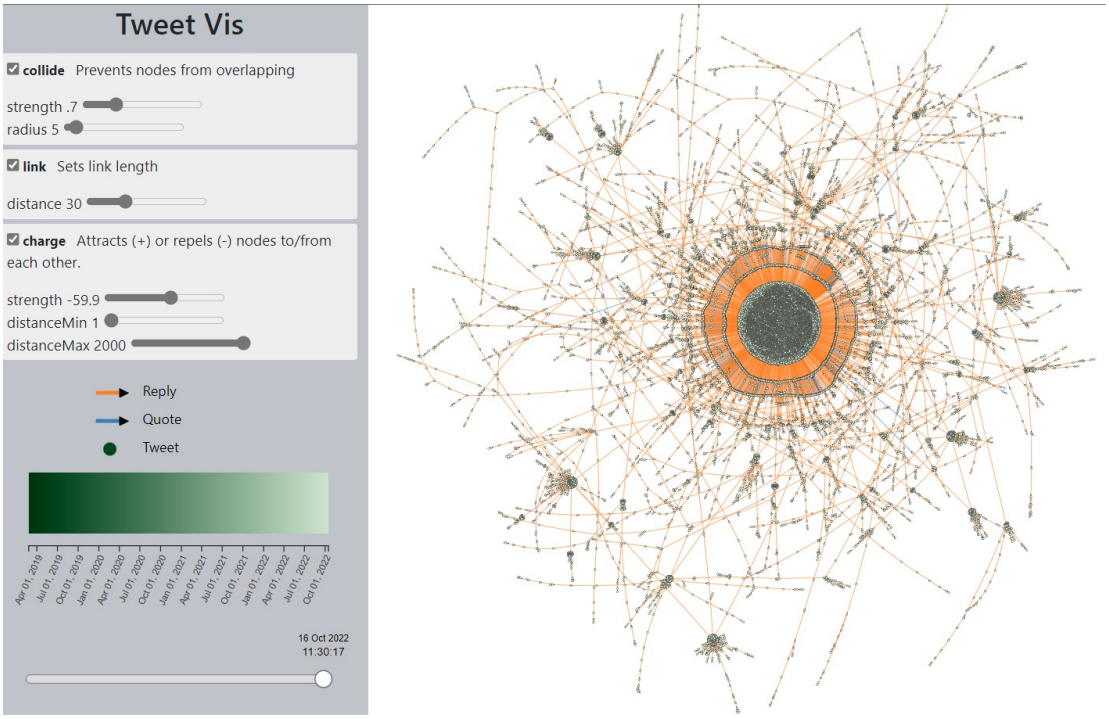


Figure 5.20: Complete graph after changed the strength to -59.9

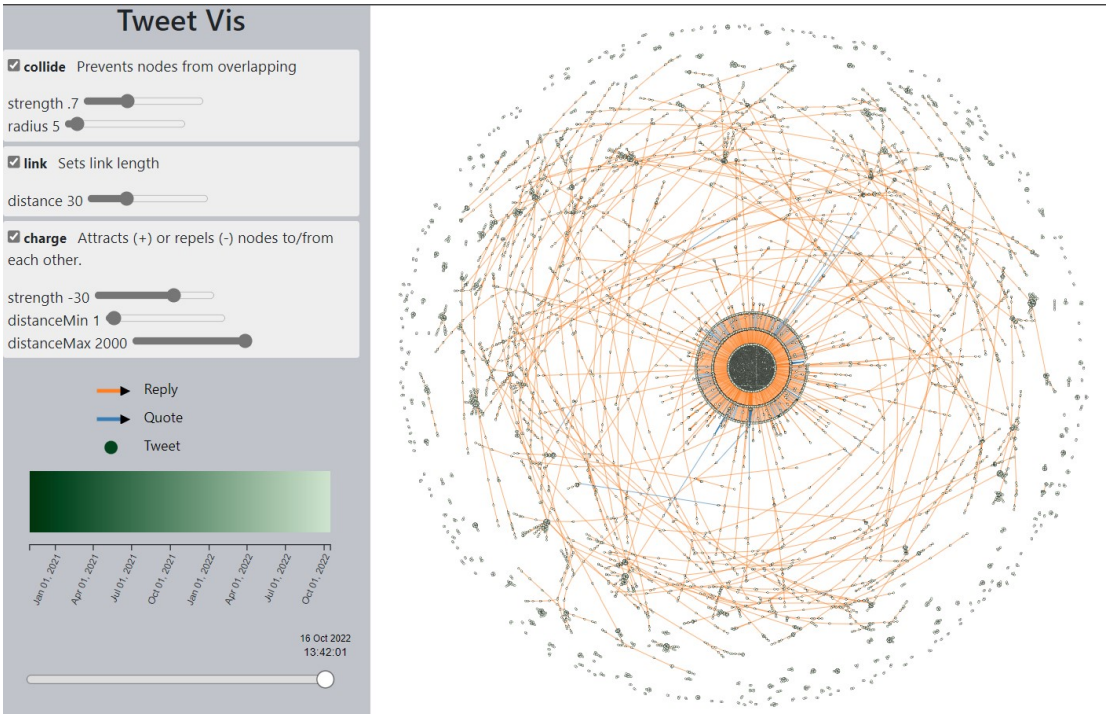


Figure 5.21: Graph produced with 20 mil nodes.

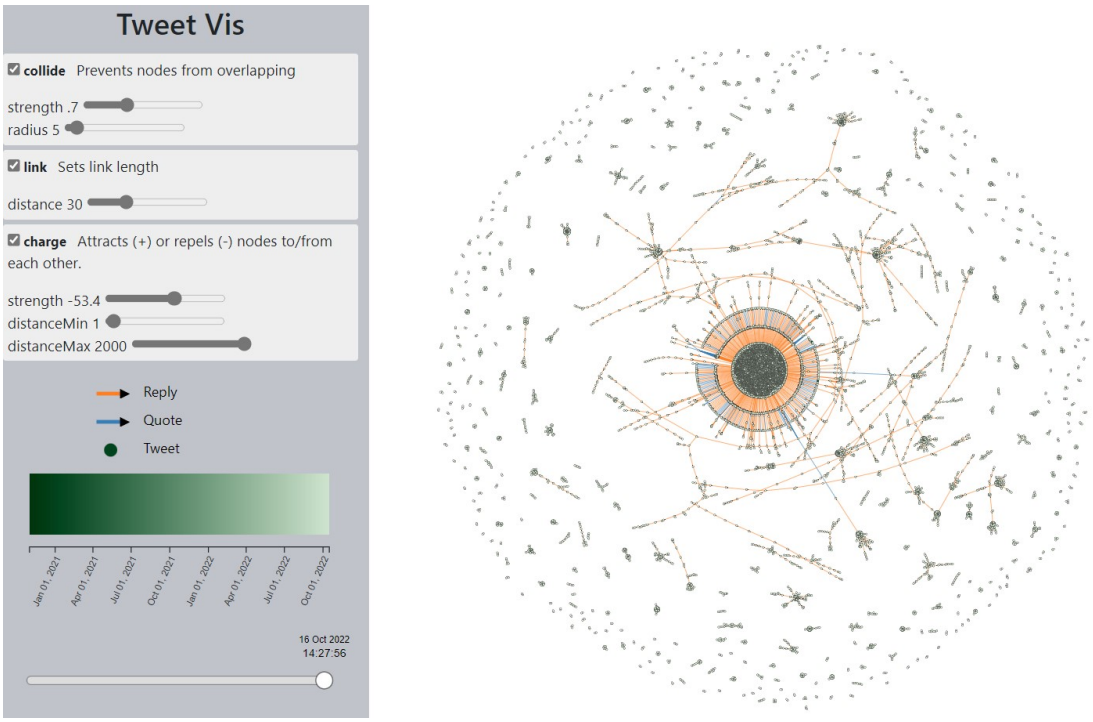


Figure 5.22: Graph produced with 15 mil nodes.

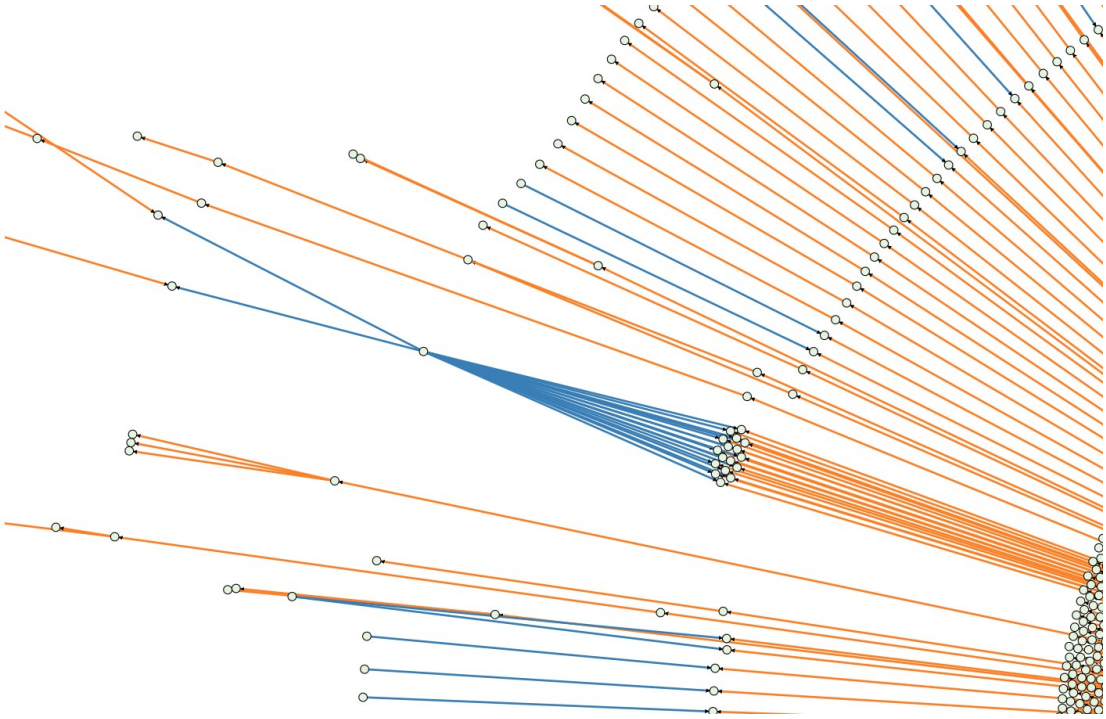


Figure 5.23: Zoom in of the graph shown in Fig. 5.22.

Figure 5.24: Details of the node shown in Fig. 5.23.

5.5 Discussion

In this section, it has been demonstrated how the tool, with its visual encodings and its interactions, behaves on different datasets. As already mentioned in the introduction, InfoVis is focused on the discovery and exploration of information, and this has been accomplished in this section.

Because Twitter is very important and many studies can be done, we have chosen to work with Twitter datasets. Unfortunately, the amount of Twitter data with geo-location is minimal, so it has been decided to work with a visualization focused on graphs, unlike several articles (Yin et al. [30], Brodkorb et al. [20], Boyandin et al. [17],Graser et al. [13]) analyzed previously. The papers that focused on making visualizations on maps had data with geographic locations, which highlights the importance that visualization development is highly dependent on the data source.

The developed visualization tool uses the force-direct placement algorithm for element positioning. The force-directed placement algorithm allows the user to modify the position of the nodes either by using drag-and-drop interaction or by modifying the forces with the left-side panel tools. The force-direct placement algorithm implemented with D3.js has proven useful in visualized conversations. In the work of Bliss et al. [32], they use an algorithm called Force Atlas 2, which is convenient for their work since the nodes in their reply network can have different levels of connection. Also, in the work of Bliss et al. [32], it is useful to have two nodes close together when they are highly connected. The nodes in the InfoVis tool developed are tweets, and there cannot be many links between tweets, so there would be no advantage in using an algorithm such as Force Atlas 2.

The tool panel allows you to modify the layout easily. For example, it is possible to relax the force that prevents the overlapping of nodes so that the nodes do not move too far

away. As an illustration, the tool panel was used to obtain Fig. 5.20. Initially, the collision force is increased as well as its radius so that the nodes are separated from each other, and then the repulsion between nodes is increased. Then the collide force parameters are returned to their default values, and the result is a graph where paths are more easily visualized than isolated nodes as in Fig. 5.18 and Fig. 5.22. Also, with the tool panel, the user can modify the distance of the links. The link modification can be useful for the paths that are formed not to extend too much on the display screen. You can also modify the strength with which the elements repel and attract each other. Increasing the repulsion between nodes can be useful to move node clusters away from isolated nodes, as in Fig. 5.12.

The force-direct placement algorithm adjusts the elements iteratively, which can be time-consuming, but modifying the parameters can speed up the iterative process. The positioning iterations can be accelerated by modifying certain forces, for example, by having more repulsion between the nodes. This delay that exists until the visualization elements are positioned according to forces can be seen as a disadvantage because of the time it takes, but this happens with very large datasets. Also, the final layout is usually very good since the algorithm in its measure avoids overlapping of nodes.

Although there are forces to prevent nodes from overlapping and the algorithm minimizes the crossing of links, there are still areas where it is difficult to visualize the nodes and their links. In Fig. 5.22, it is difficult to visualize the nodes and their links easily, but with the semantic zoom, this improves as in Fig. 5.24. In Fig. 5.24, it can still be easy to lose the path of the tweets, but this improves with the path highlighting as in Fig. 5.19. Despite there being several interactions when there are large numbers of nodes the links often overlap each other and can become somewhat confusing. One way that has not been implemented that could help avoid link overlap is to use arc-shaped links as in Hawelka et al. [29], Froio and Ganesh [27], and Bongsug [25]. As seen in the images of the visualizations of the mentioned articles, the arc-shaped links avoid confusion and overlapping links. Nevertheless, another way to avoid overlaps or between crossing nodes and links is by using 3D graphs as in the work of Mitrpanont et al. [146]. Work that focuses on 3D visualizations can be useful to avoid overlaps but also requires more computational power to render the elements.

As seen in the previously analyzed conversation graphs, many tweets are replicas of the main tweet, but after them, there are no more interactions. These nodes that are seen as highly dense clouds of nodes could be treated with an abstraction technique that only shows a few nodes instead of a large number of nodes, abstraction techniques such as the ones used in Graser et al.[13] and Brodkorb et al. [20].

A weakness of the positioning algorithm was evident when visualizing the conversations with the tool. The placement of the elements by the force-directed placement algorithm is not deterministic. Each time the algorithm was run, i.e., visualization, a different layout was produced. When running the tool, the user may not get the same positioning of elements, mainly because the algorithm randomly places the elements. The randomness leads to the inability to take advantage of the user's spatial memory since the network's layout may differ in different algorithm executions.

The articles analyzed within the state-of-the-art are not focused on studying or proposing a new positioning algorithm. In general, many papers already use implemented algorithms that minimize the amount of distracting elements such as cross-links and overlaps.

This work has also not focused on developing any positioning algorithm. Instead, the force-directed placement algorithm is chosen as the most appropriate one. The algorithm used provides ease of implementation, is easy to understand and explain at a conceptual level, and is adequate to reflect the structure of the networks.

One problem noted with the visualization tool was when using datasets of more than 10,000 nodes, in which case the rendering of the visual objects becomes a cumbersome task for the web browser. Small datasets produce readable designs quickly, but when handling thousands of nodes, the designs can be prone to visual clutter, and the time to a final layout increases. Additionally, the parameters that can be used to modify a network's layout can help improve the visualizations. However, the configurations must be modified to work well in both datasets.

Running visualizations with a large number of nodes is beyond the scope of this work. Although the work has not focused on handling huge amounts of nodes and links, we can think of solutions to this problem. One of the simplest ways would be to use canvas instead of SVG to render the visual elements. Canvas generally performs better than SVG, but the disadvantage is that event binding, and interactivity is not as easy as in SVG. It is known that canvas can handle the movement of many objects more easily, but this should be studied in the scope of this visualization tool. Canvas may have been a better choice than SVG to visualize the elements, but this alternative should be evaluated with the specific application. Another way the visualization tool's performance could be improved could be by using libraries that work with GPU. For example, you can use the Stardust library, which allows GPU rendering, and integrate it with D3.js for force-directed graph visualization.

State-of-art works have not yet taken advantage of the Twitter API to obtain conversations. In this work, we take advantage of the Twitter API to visualize the trajectories of conversations on Twitter. Within a Twitter conversation, several tweets are involved, and each of these tweets has some information. Some of the information in the Tweets has been encoded for visualization, and some are simply displayed for informational purposes. The time is coded with the color intensity (saturation), which allows differentiating older tweets from recent ones. However, when analyzing specific conversations, it was noticed that when the dates of the tweets are progressive, different intensities are seen in the nodes. On the other hand, when there are tweets that are temporally far away from the rest of the tweets, the color of the tweets does not vary, and their temporality is not easily noticed.

When there are tweets far away temporally from the rest of the tweets, there is no granularity in color intensity changes, even when using the temporal filter. The changes are minimal or practically null when tweets are close to each other temporally. On the other hand, the change in color intensity and the temporal filter is very recognizable from a tweet that is far away in time compared to the rest of the tweets. For example, the oldest tweet will be seen with a higher intensity than all the other tweets, while the rest will have a similar intensity. In the temporal filter, the first node will appear immediately, while the rest will take some time. These nodes that take time to appear will be generated immediately with a short sliding slider; this abrupt appearance of nodes does not allow us to appreciate their changes in time. The above described is a problem in the visualization tool and could be improved by simply removing from the dataset these tweets that are too far away in time compared to the rest of the tweets.

The saturation intensity chosen for color coding belongs to the magnitude channel and is, therefore, suitable for sorted data. The color intensity shows low accuracy for non-contiguous regions, but it is the best for the InfoVis tool development. Another alternative to encoding the temporality and according to Fig. 2.16, is the size of the objects. The saturation, together with the size, can interact strongly, and in this way, it would be easier to perceive the temporality, but the size of the object, in this case, could be misunderstood. The large size of an object can make certain nodes more important even though they are not. The size of a node would help encode temporality, but as seen in Guille and Favre [26], Davis et al. [31], and Bliss et al. [32], larger objects are generally perceived as having greater importance.

Another channel that could be even better to encode temporality is the position of the elements along with the size in length of the links. To achieve this, the force-directed placement algorithm should be modified so that the nodes' distances change with time. Modifying the algorithm would help a lot in interpreting the timing of the nodes, but due to the complex structures that can exist in a Twitter conversation, it would be a complex task that could be addressed in future work.

Finally, information is also encoded in the network formed by the tweets and the types of interactions. The user name and tweet text are not coded to a visual channel but provide important information, as seen in the previous section. The information that is not visually coded is, for example, the tweet's text. As seen in the conversation analysis, the tweet text can provide vital information to discover and discern what is happening with specific tweets and interactions.

Chapter 6

Conclusions and Future Work

This work proposed a visualization tool based on fundamental, theoretical, and practical aspects of information visualization design to analyze multidimensional Twitter data. The most important thing about the proposal is that unlike many other works on Twitter, this one addresses Twitter conversations and works based on the available data and with several well-studied visual coding techniques. A directed graph was applied to show the network of the data. The color hue channel was applied to encode the different relationships between tweets. The color saturation channel was used to differentiate the temporality in the tweets. Markers in the form of arrows were also used to indicate the direction of the tweets' trajectory. Additionally, several interactions were used to improve the use of the tool, such as drag-and-drop, hovers, semantic zoom, highlighting, a slider that allows the dynamic change of the graph depending on the time, and several controls that allow controlling the forces that govern the behavior of the layout. The techniques mentioned above and the visualization process are based on Tamara Munzner's framework for visualization generation and her nested model for visualization design.

A visualization tool was successfully developed that satisfies the task of analyzing the trajectories and relationships of Twitter conversations using the most appropriate techniques for the available data and that additionally can be adapted not only to Twitter conversations. As already mentioned, tweets do not usually have a latitude and longitude, so the chosen way to encode this is to use a graph. The primary coding for the trajectories and relationships was through a force-directed graph. The trajectory is visualized through the arrows on the network links, and the relationships are identified with different colors (either replica or quote). The most important to work with a force-directed graph in D3.js is the use of JSON files with nodes and links. Since most Twitter datasets are focused on sentiment analysis to retweets, it was necessary to use the Twitter API to process the information obtained. Since the tool works generically with a dataset of nodes and links, it is possible to visualize the graphs of any dataset with this same structure, and in this way, the tool can be adapted to other uses. With the tool, it is possible to explore and discover patterns or aspects that have to do with the network structure of a specific dataset.

A visualization tool was presented that pays attention to the temporal dimension based on the fundamental aspects for developing visualizations and complies with certain principles and design choices suitable for analyzing Twitter conversations. The temporal dimension, often not addressed in visualization tools, is one of the most remarkable aspects of

the developed visualization tool. In the same way, basic coding principles were applied to make the tool capable of showing how a Twitter conversation evolves. In this way, it allows any user to determine time windows where the generation of tweets is higher or lower or explore and visualize how conversations change over time.

A visualization tool was developed that can be easily shared and integrates interactions through its development in HTML, CSS, and JavaScript for use from a web browser. The tool can be easily shared with a link and can run easily in any web browser as the tools used for its development are widely used and known. In addition, JavaScript and D3.js add a series of interactions that facilitate the use of the tool. The most challenging part of this design choice is handling large amounts of nodes, which becomes a cumbersome task. For future work, we can study the use of canvas to improve the fluidity of the tool with large nodes and the possibility of using other tools that work with GPU for rendering.

As mentioned in the discussion section, some aspects can be improved and can be addressed in future work. The most notable is modifying the forced-direct-placement algorithm of D3.js to improve the visual coding, making the nodes closer in time to a central node at a closer distance and the nodes farther in time to be at a greater distance. An issue that can also be addressed in future work is reshaping links to arcs and using node aggregations to avoid overlaps and visual clutter. Another aspect that can be improved is when some tweets or nodes are far away in time from the rest. In this area, other types of scaling or eliminating nodes can be addressed if they do not have essential participation in the conversations.

Bibliography

[1] D. Cotrim and J. Campos, "Representação das características do movimento de objetos móveis em mapas estáticos," *Proceedings of the Brazilian Symposium on GeoInformatics*, 01 2007.

[2] Y.-C. Chang, C.-H. Ku, and D.-D. L. Nguyen, "Predicting aspect-based sentiment using deep learning and information visualization: The impact of covid-19 on the airline industry," *Information Management*, vol. 59, no. 2, p. 103587, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378720621001610

[3] J. Jung, "Code clouds: Qualitative geovisualization of geotweets: Code clouds," *The Canadian Geographer / Le Géographe canadien*, vol. 59, 09 2014.

[4] C. Musto, G. Semeraro, M. de Gemmis, and P. Lops, "Modeling community behavior through semantic analysis of social data: The italian hate map experience," 07 2016, pp. 307–308.

[5] M. Chavent, A. Vanel, A. Tek, B. Levy, S. Robert, B. Raffin, and M. Baaden, "Gpu-accelerated atom and dynamic bond visualization using hyperballs: A unified algorithm for balls, sticks, and hyperboloids," *Journal of computational chemistry*, vol. 32, pp. 2924–35, 10 2011.

[6] F. Ferstl, K. Bürger, and R. Westermann, "Streamline variability plots for characterizing the uncertainty in vector field ensembles," *IEEE transactions on visualization and computer graphics*, vol. 22, 09 2015.

[7] T. Munzner, *Visualization Analysis and Design*, ser. AK Peters Visualization Series. CRC Press, 2014. [Online]. Available: https://books.google.es/books?id=NfkYCwAAQBAJ

[8] ——, "A nested model for visualization design and validation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 921–928, 2009.

[9] N. Andrienko and G. Andrienko, *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer, 2006.

[10] Z. Feng and Y. Zhu, "A survey on trajectory data mining: Techniques and applications," *IEEE Access*, vol. 4, pp. 2056–2067, 2016.

[11] M. Imran, U. Qazi, and F. Ofli, "Tbcov: Two billion multilingual covid-19 tweets with sentiment, entity, geo, and gender labels," *Data*, vol. 7, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2306-5729/7/1/8

[12] S. Schöttler, Y. Yang, H. Pfister, and B. Bach, "Visualizing and interacting with geospatial networks: A survey and design space," 01 2021.

[13] A. Graser, J. Schmidt, F. Roth, and N. Brändle, "Untangling origin-destination flows in geographic information systems," *Information Visualization*, vol. 18, no. 1, pp. 153–172, 2019. [Online]. Available: https://doi.org/10.1177/1473871617738122

[14] K. N. S. Behara, A. Bhaskar, and E. Chung, "Geographical window based structural similarity index for origin-destination matrices comparison," *Journal of Intelligent Transportation Systems*, vol. 26, no. 1, pp. 46–67, 2022. [Online]. Available: https://doi.org/10.1080/15472450.2020.1795651

[15] J. Wood, J. Dykes, and A. Slingsby, "Visualisation of origins, destinations and flows with od maps," *Cartographic Journal, The*, vol. 47, pp. 117–129, 05 2010.

[16] Y. Yang, T. Dwyer, S. Goodwin, and K. Marriott, "Many-to-many geographically-embedded flow visualisation: An evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, pp. 411–420, 2017.

[17] I. Boyandin, E. Bertini, P. Bak, and D. Lalanne, "Flowstrates: An approach for visual exploration of temporal origin-destination data," *Computer Graphics Forum*, vol. 30, no. 3, pp. 971–980, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.01946.x

[18] M. T. Gastner and M. E. J. Newman, "Diffusion-based method for producing density-equalizing maps," *Proceedings of the National Academy of Sciences*, vol. 101, no. 20, pp. 7499–7504, 2004. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.0400280101

[19] Q. W. Bouts, T. Dwyer, J. Dykes, B. Speckmann, S. Goodwin, N. H. Riche, M. S. T. Carpendale, and A. Liebman, "Visual encoding of dissimilarity data via topology-preserving map deformation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 2200–2213, 2016.

[20] F. Brodkorb, A. Kuijper, G. Andrienko, N. Andrienko, and T. Landesberger, "Overview with details for exploring geo-located graphs on maps," *Information Visualization*, vol. 15, 08 2015.

[21] J. Heer and D. Boyd, "Vizster: visualizing online social networks," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, 2005, pp. 32–39.

[22] N. Henry, J.-D. Fekete, and M. J. McGuffin, "Nodetrix: a hybrid visualization of social networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1302–1309, 2007.

[23] B. Bach, E. Pietriga, I. Liccardi, and G. Legostaev, "Ontotrix: a hybrid visualization for populated ontologies," *Proceedings of the 20th international conference companion on World wide web*, 2011.

[24] G. J. Abel and N. Sander, "Quantifying global international migration flows," *Science*, vol. 343, no. 6178, pp. 1520–1522, 2014. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1248676

[25] B. Chae, "Insights from hashtag supplychain and twitter analytics: Considering twitter and twitter data for supply chain practice and research," *International Journal of Production Economics*, vol. 165, pp. 247–259, 07 2015.

[26] A. Guille and C. Favre, "Event detection, tracking and visualization in twitter: A mention-anomaly-based approach," *Springer Social Network Analysis and Mining*, vol. 5, pp. 18:1–18:18, 05 2015.

[27] C. Froio and B. Ganesh, "The transnationalisation of far-right discourse on twitter. issues and actors that cross borders in western european democracies," *European Societies*, vol. 21, 07 2018.

[28] T. Masaharu, Y. Onoue, H. Torii, S. Suda, K. Mori, Y. Nishikawa, A. Ozaki, and K. Uno, "Twitter use in scientific communication revealed by visualization of information spreading by influencers within half a year after the fukushima daiichi nuclear power plant accident," *PLOS ONE*, vol. 13, p. e0203594, 09 2018.

[29] B. Hawelka, I. Sitko, E. Beinat, S. Sobolevsky, P. Kazakopoulos, and C. Ratti, "Geolocated twitter as proxy for global mobility patterns," *Cartography and Geographic Information Science*, vol. 41, 11 2013.

[30] J. Yin, Y. Gao, Z. Du, and S. Wang, "Exploring multi-scale spatiotemporal twitter user mobility patterns with a visual-analytics approach," *ISPRS International Journal of Geo-Information*, vol. 5, no. 10, 2016. [Online]. Available: https://www.mdpi.com/2220-9964/5/10/187

[31] C. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "Botornot: A system to evaluate social bots," 04 2016, pp. 273–274.

[32] C. A. Bliss, I. M. Kloumann, K. D. Harris, C. M. Danforth, and P. S. Dodds, "Twitter reciprocal reply networks exhibit assortativity with respect to happiness," *Journal of Computational Science*, vol. 3, no. 5, pp. 388–397, 2012, advanced Computing Solutions for Health Care and Medicine. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187775031200049X

[33] S. Murray, *Interactive Data Visualization for the Web: An Introduction to Designing with D3*. O'Reilly Media, 2013.

[34] S. Schlosser, D. Toninelli, and M. Cameletti, "Comparing methods to collect and geolocate tweets in great britain," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 7, p. 44, 01 2021.

[35] D. Antonakaki, P. Fragopoulou, and S. Ioannidis, "A survey of twitter research: Data model, graph structure, sentiment analysis and attacks," *Expert Systems with Applications*, vol. 164, p. 114006, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742030779X

[36] Marketingcharts. Social networking eats up 3+ hours per day for the average american user. Accessed: 2022-05-10. [Online]. Available: https://www.marketingcharts.com/digital-26049

[37] "Data dictionary: Standard v1.1, howpublished = https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet, note = Accessed: 2022-02-10."

[38] M. Celik and A. S. Dokuz, "Discovering socio-spatio-temporal important locations of social media users," *Journal of Computational Science*, vol. 22, pp. 85–98, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877750317308165

[39] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Commun. ACM*, vol. 59, no. 7, p. 96–104, jun 2016. [Online]. Available: https://doi.org/10.1145/2818717

[40] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: Quantifying influence on twitter," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, ser. WSDM '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 65–74. [Online]. Available: https://doi.org/10.1145/1935826.1935845

[41] M. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, F. Menczer, and A. Flammini, "Political polarization on twitter," in *Proceedings of the international aaai conference on web and social media*, vol. 5, no. 1, 2011, pp. 89–96.

[42] R. Nishi, T. Takaguchi, K. Oka, T. Maehara, M. Toyoda, K.-i. Kawarabayashi, and N. Masuda, "Reply trees in twitter: data analysis and branching process models," *Social network analysis and mining*, vol. 6, no. 1, pp. 1–13, 2016.

[43] R. Spence, *Information Visualization: Design for Interaction*, 2nd ed. Pearson, 2007.

[44] S. Fadloun, *Visualisation dínformation.* Algérie: Ecole nationale Supérieure dÍnformatique, 2022.

[45] J.-D. Fekete, J. J. van Wijk, J. T. Stasko, and C. North, *The Value of Information Visualization.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–18. [Online]. Available: https://doi.org/10.1007/978-3-540-70956-5_1

[46] B. MIREL, "1 - what makes complex problem solving complex?" in *Interaction Design for Complex Problem Solving*, ser. Interactive Technologies, B. MIREL, Ed. Burlington: Morgan Kaufmann, 2004, pp. 3–29. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781558608313500023

[47] J. van Wijk, "The value of visualization," in *VIS 05. IEEE Visualization, 2005.*, 2005, pp. 79–86.

[48] B. H. McCormick, T. A. DeFanti, and M. D. Brown, "Visualization in scientific computing," in *Computer Graphics*, vol. 21(6), 1987.

[49] T. DeFanti, M. Brown, and B. McCormick, "Visualization: expanding scientific and engineering research opportunities," *Computer*, vol. 22, no. 8, pp. 12–16, 1989.

[50] R. Moorhead, C. Johnson, T. Munzner, H. Pfister, P. Rheingans, and T. Yoo, "Visualization research challenges - a report summary," *Computing in Science Engineering*, vol. 8, pp. 66 – 73, 08 2006.

[51] H. Paggi, J. Soriano, J. A. Lara, and E. Damiani, "Towards the definition of an information quality metric for information fusion models," *Computers Electrical Engineering*, vol. 89, p. 106907, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S004579062030759X

[52] R. Capurro and B. Hjørland, "The concept of information," 2003. [Online]. Available: http://hdl.handle.net/10150/105705

[53] J. Lamping and R. Rao, "Laying out and visualizing large trees using a hyperbolic space," ser. UIST '94. New York, NY, USA: Association for Computing Machinery, 1994, p. 13–14. [Online]. Available: https://doi.org/10.1145/192426.192430

[54] K. Andrews, "Visual exploration of large hierarchies with information pyramids," in *Proceedings Sixth International Conference on Information Visualisation*, 2002, pp. 793–798.

[55] K. Andrews, M. Osmić, and G. Schagerl, "Aggregated parallel coordinates: Integrating hierarchical dimensions into parallel coordinates visualisations," in *Proceedings of the 15th International Conference on Knowledge Technologies and Data-Driven Business*, ser. i-KNOW '15. New York, NY, USA: Association for Computing Machinery, 2015. [Online]. Available: https://doi.org/10.1145/2809563.2809588

[56] T. Müller, "Geovis—relativistic ray tracing in four-dimensional spacetimes," *Computer Physics Communications*, vol. 185, no. 8, pp. 2301–2308, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465514001362

[57] A. Joshi, M. Novaes, J. Machiavelli, S. Iyengar, R. Vogler, C. Johnson, and J. Zhang, "A human centered geovisualization framework to facilitate visual exploration of telehealth data: A case study," *Technology and health care : official journal of the European Society for Engineering and Medicine*, vol. 20, pp. 457–71, 11 2012.

[58] C. Rinner, "A geographic visualization approach to multi-criteria evaluation of urban quality of life," *International Journal of Geographical Information Science*, vol. 21, no. 8, pp. 907–919, 2007. [Online]. Available: https://doi.org/10.1080/13658810701349060

[59] B. Ki and S. Klasky, "Scivis," *Concurrency: Practice and Experience*, vol. 10, no. 11-13, pp. 1107–1115, 1998.

[60] C. Wang and J. Han, "Dl4scivis: A state-of-the-art survey on deep learning for scientific visualization," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022.

[61] R. Li and J. Chen, "Toward a deep understanding of what makes a scientific visualization memorable," in *2018 IEEE Scientific Visualization Conference (SciVis)*, 2018, pp. 26–31.

[62] A. Coltekin, I. Lokka, and M. Zahner, "On the usability and usefulness of 3d (geo)visualizations – a focus on virtual reality environments," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B2, pp. 387–392, 06 2016.

[63] M. Card, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[64] S. Lallé and C. Conati, "The role of user differences in customization: A case study in personalization for infovis-based content," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, ser. IUI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 329–339. [Online]. Available: https://doi.org/10.1145/3301275.3302283

[65] G. Judelman, "Aesthetics and inspiration for visualization design: bridging the gap between art and science," in *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.*, 2004, pp. 245–250.

[66] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1224–1231, 2007.

[67] Z. Pousman, J. Stasko, and M. Mateas, "Casual information visualization: Depictions of data in everyday life," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1145–1152, 2007.

[68] R. Buck, "Chapter two - motivation, emotion, cognition, and communication: Definitions and notes toward a grand theory," ser. Advances in Motivation Science, A. J. Elliot, Ed. Elsevier, 2019, vol. 6, pp. 27–69. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215091918300178

[69] M. Sorapure, "Text, image, data, interaction: Understanding information visualization," *Computers and Composition*, vol. 54, p. 102519, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S875546151830032X

[70] Z. Pousman, J. Stasko, and M. Mateas, "Casual information visualization: Depictions of data in everyday life," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1145–1152, 2007.

[71] T. Reichenbacher and O. Swienty, "Attention-guiding geovisualisation," 05 2007.

[72] M.-J. Kraak, "Geovisualization illustrated," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 57, pp. 390–399, 04 2003.

[73] K. Brodlie, L. Carpenter, R. Earnshaw, J. Gallop, R. Hubbold, A. Mumford, C. Osland, and P. Quarendon, *Scientific Visualization: Techniques and Applications*. Springer Berlin Heidelberg, 2012. [Online]. Available: https://books.google.es/books?id=RUKqCAAAQBAJ

[74] B. H. McCormick, T. A. DeFanti, and M. D. Brown, "Visualization in scientific computing," in *Computer Graphics*, vol. 21(6), 1987, pp. 15,21.

[75] L. Manovich, "What is visualisation?" *Visual Studies*, vol. 26, pp. 36–49, 03 2011.

[76] W. J. SCHROEDER and K. M. MARTIN, "30 - the visualization toolkit," in *Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds. Burlington: Butterworth-Heinemann, 2005, pp. 593–614. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123875822500320

[77] J. Kim, Y. Jung, D. D. Feng, and M. J. Fulham, "Chapter seventeen - biomedical image visualization and display technologies," in *Biomedical Information Technology (Second Edition)*, second edition ed., ser. Biomedical Engineering, D. D. Feng, Ed. Academic Press, 2020, pp. 561–583. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128160343000171

[78] R. Haber and D. McNabb, "Visualization idioms: A conceptual model for visualization systems," *Visualization in Scientific Computing*, pp. 74–93.

[79] H. Schumann and W. Müller, *Visualisierung: Grundlagen und allgemeine methoden*. Springer-Verlag, 2013.

[80] E. Chi, "A taxonomy of visualization techniques using the data state reference model," in *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, 2000, pp. 69–75.

[81] E. H.-H. Chi and J. Riedl, "An operator interaction framework for visualization systems," in *Proceedings IEEE Symposium on Information Visualization (Cat. No.98TB100258)*, 1998, pp. 63–70.

[82] S. dos Santos and K. Brodlie, "Gaining understanding of multivariate and multidimensional data through visualization," *Computers Graphics*, vol. 28, no. 3, pp. 311–325, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849304000251

[83] C. Tominski, "Event-based visualization for user-centered visual analysis," Ph.D. dissertation, 01 2006.

[84] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, *Visual Analytics: Definition, Process, and Challenges*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 154–175. [Online]. Available: https://doi.org/10.1007/978-3-540-70956-5_7

[85] J. Kohlhammer, D. Keim, M. Pohl, G. Santucci, and G. Andrienko, "Solving problems with visual analytics," *Procedia Computer Science*, vol. 7, p. 117–120, 12 2011.

[86] P. C. Wong and J. J. Thomas, "Visual analytics," *IEEE computer graphics and applications*, vol. 24 5, pp. 20–1, 2004.

[87] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, *Mastering The Information Age – Solving Problems with Visual Analytics*, 01 2010.

[88] E. Dimara and C. Perin, "What is interaction for data visualization?" *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 119–129, 2020.

[89] J. Thomas and K. Cook, "A visual analytics agenda," *Computer Graphics and Applications, IEEE*, vol. 26, pp. 10– 13, 02 2006.

[90] S. L. O. VISUALIZATION, "Interaction for visualization."

[91] G. W. Furnas and B. B. Bederson, "Space-scale diagrams: Understanding multiscale interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '95. USA: ACM Press/Addison-Wesley Publishing Co., 1995, p. 234–241. [Online]. Available: https://doi.org/10.1145/223904.223934

[92] B. B. Bederson, "The promise of zoomable user interfaces," *Behaviour & Information Technology*, vol. 30, no. 6, pp. 853–866, 2011. [Online]. Available: https://doi.org/10.1080/0144929X.2011.586724

[93] M. Tory and T. Moller, "Evaluating visualizations: do expert reviews work?" *IEEE Computer Graphics and Applications*, vol. 25, no. 5, pp. 8–11, 2005.

[94] T. Zuk, L. Schlesier, P. Neumann, M. Hancock, and S. Carpendale, "Heuristics for information visualization evaluation," 01 2006, pp. 1–6.

[95] C. Ware, H. Purchase, L. Colpoys, and M. McGill, "Cognitive measurements of graph aesthetics," *Information Visualization*, vol. 1, pp. 103–110, 06 2002.

[96] E. Bertini and D. Lalanne, "Surveying the complementary role of automatic data analysis and visualization in knowledge discovery," ser. VAKD '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 12–20. [Online]. Available: https://doi.org/10.1145/1562849.1562851

[97] G. J. Klir, *Architecture of systems problem solving*. Springer Science & Business Media, 2013.

[98] J.-D. Fekete, "The infovis toolkit," in *IEEE Symposium on Information Visualization*, 2004, pp. 167–174.

[99] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "Dnn-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPACIAL '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2996913.2997016

[100] O. Schabenberger and C. Gotway, *Statistical Methods for Spatial Data Analysis*, 1st ed. Chapman and Hall/CRC, 2005.

[101] E. Pebesma, "spacetime: Spatio-temporal data in r," *Journal of Statistical Software*, vol. 51, no. 7, p. 1–30, 2012. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v051i07

[102] D. Peuquet, "It's about time: A conceptual framework for the representation of temporal dynamics in geographic information systems," *Annals of the American Association of Geographers*, vol. 84, no. 3, pp. 441–461, 9 1994.

[103] G. Andrienko, N. Andrienko, P. Bak, D. Keim, and S. Wrobel, *Visual Analytics of Movement*. Springer, 2013.

[104] N. Andrienko, G. Andrienko, N. Pelekis, and S. Spaccapietra, *Basic Concepts of Movement Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 15–38. [Online]. Available: https://doi.org/10.1007/978-3-540-75177-9_2

[105] F. Barbieri, L. E. Anke, and J. Camacho-Collados, "Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond," 2021. [Online]. Available: https://arxiv.org/abs/2104.12250

[106] R. Villa-Cox, S. Kumar, M. Babcock, and K. M. Carley, "Stance in replies and quotes (srq): A new dataset for learning stance in twitter conversations," 2020. [Online]. Available: https://arxiv.org/abs/2006.00691

[107] S. Fadloun, Y. Morakeb, E. Cuenca, and K. Choutri, "Trajectoryvis: a visual approach to explore movement trajectories," *Social Network Analysis and Mining*, vol. 12, no. 1, pp. 1–17, 2022.

[108] D. Selassie, B. Heller, and J. Heer, "Divided edge bundling for directional network data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2354–2363, 2011.

[109] D. Zielasko, B. Weyers, B. Hentschel, and T. Kuhlen, "Interactive 3d force-directed edge bundling," *Computer Graphics Forum*, vol. 35, pp. 51–60, 06 2016.

[110] 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. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01450.x

[111] 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, pp. 2364–73, 12 2011.

[112] A. Voorhees, "A general theory of traffic movement: The 1955 ite past presidents' award paper," *Transportation*, vol. 40, 11 2013.

[113] L. Wilkinson and M. Friendly, "The history of the cluster heat map," *The American Statistician*, vol. 63, pp. 179–184, 05 2009.

[114] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff, "Boundary labeling: Models and efficient algorithms for rectangular maps," *Computational Geometry*, vol. 36, no. 3, pp. 215–236, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925772106000447

[115] M. Gastner and M. Newman, "Diffusion-based method for producing density equalizing maps. proc. natl. acad. sci. usa 101, 7499-7504," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, pp. 7499–504, 06 2004.

[116] S. Sun, "An optimized rubber-sheet algorithm for continuous area cartograms," *Professional Geographer - PROF GEOGR*, vol. 65, 01 2012.

[117] N. Ahmed and H. Miller, "Time-space transformations of geographic space for exploring, analyzing and visualizing transportation systems," *Journal of Transport Geography*, vol. 15, pp. 2–17, 01 2007.

[118] E. Shimizu and R. Inoue, "A new algorithm for distance cartogram construction," *International Journal of Geographical Information Science*, vol. 23, pp. 1453–1470, 11 2009.

[119] J. Böttger, U. Brandes, O. Deussen, and H. Ziezold, "Map warping for the annotation of metro maps," *IEEE Computer Graphics and Applications*, vol. 28, no. 5, pp. 56–65, 2008.

[120] S.-S. Lin, C.-H. Lin, Y.-J. Hu, and T.-Y. Lee, "Drawing road networks with mental maps," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, pp. 1241–1252, 09 2014.

[121] B. Jenny, "Adaptive composite map projections," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, pp. 2575–2582, 12 2012.

[122] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[123] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, no. 1, pp. 7–15, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020019089901026

[124] S. Tabassum, F. S. F. Pereira, S. Fernandes, and J. Gama, "Social network analysis: An overview," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 5, p. e1256, 2018. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1256

[125] J. R. Enos and R. R. Nilchiani, "Understanding the importance of expanding the definition of interoperability through social network analysis," *Systems Engineering*, vol. 23, no. 2, pp. 139–153, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21500

[126] S. Kolli and D. Khajeheian, *How Actors of Social Networks Affect Differently on the Others? Addressing the Critique of Equal Importance on Actor-Network Theory by Use of Social Network Analysis.* Singapore: Springer Singapore, 2020, pp. 211–230. [Online]. Available: https://doi.org/10.1007/978-981-15-7066-7_12

[127] M. Ghoniem, J.-D. Fekete, and P. Castagliola, "On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis," *Information Visualization Journal*, vol. 4, 05 2005.

[128] N. Henry and J.-d. Fekete, "Matrixexplorer: a dual-representation system to explore social networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 677–684, 2006.

[129] S. Hennemann, "Information-rich visualisation of dense geographical networks," *Journal of Maps*, vol. 9, pp. 68–75, 03 2013.

[130] E. Gansner and Y. Koren, "Improved circular layouts," 09 2006.

[131] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[132] H.-J. Kim, Y. K. Jeong, Y. Kim, K. Kang, and M. Song, "Topic-based content and sentiment analysis of ebola virus on twitter and in the news," *Journal of Information Science*, vol. 42, 10 2015.

[133] J. Brummette, M. DiStaso, M. Vafeiadis, and M. Messner, "Read all about it: The politicization of "fake news" on twitter," *Journalism Mass Communication Quarterly*, vol. 95, p. 107769901876990, 05 2018.

[134] A. Molla, Y. Biadgie, and K.-A. Sohn, "Network-based visualization of opinion mining and sentiment analysis on twitter," in *2014 International Conference on IT Convergence and Security (ICITCS)*, 2014, pp. 1–4.

[135] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics Theory and Experiment*, vol. 2008, 04 2008.

[136] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Commun. ACM*, vol. 59, no. 7, p. 96–104, jun 2016. [Online]. Available: https://doi.org/10.1145/2818717

[137] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software," *PloS one*, vol. 9, p. e98679, 06 2014.

[138] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," 03 2009.

[139] M. Rodriguez and P. Neubauer, "Constructions from dots and lines," *Bulletin of the American Society for Information Science and Technology*, vol. 36, 08 2010.

[140] R. Tamassia, "Advances in the theory and practice of graph drawing," *Theoretical Computer Science*, vol. 217, no. 2, pp. 235–254, 1999, oRDAL'96. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397598002722

[141] I. A. Lisitsyn and V. N. Kasyanov, "Higres — visualization system for clustered graphs and graph algorithms," in *Graph Drawing*, J. Kratochvíyl, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 82–89.

[142] H. Hosobe, "A high-dimensional approach to interactive graph visualization," in *Proceedings of the 2004 ACM Symposium on Applied Computing*, ser. SAC '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 1253–1257. [Online]. Available: https://doi.org/10.1145/967900.968155

[143] R. Tamassia, *Handbook of graph drawing and visualization*. CRC press, 2013.

[144] N. Zhu, *Data Visualization with D3.js Cookbook*. Packt Publishing, 2013. [Online]. Available: https://books.google.es/books?id=YYukAQAAQBAJ

[145] P. Cook, *D3 Start to Finish*. O'Reilly Media, 2022.

[146] J. Mitrpanont, W. Sawangphol, W. Thongrattana, S. Suthinuntasook, S. Silapadapong, and K. Kitkhachonkunlaphat, "Icdwiz: Visualizing icd-11 using 3d force-directed graph," in *Recent Challenges in Intelligent Information and Database Systems*, T.-P. Hong, K. Wojtkiewicz, R. Chawuthai, and P. Sitek, Eds. Singapore: Springer Singapore, 2021, pp. 321–334.

Appendices

Listing 1: Twitter API JSON response sample

```
{
    "data": [
        {
            "id": "1570477577432694785",
            "text": "@RoyalFamily Rest in peace QUEEN
              ELIZABETH II",
            "author_id": "1564165997430538240",
            "referenced_tweets": [
                {
                    "type": "replied_to",
                    "id": "1570117203583696896"
                }
            ],
            "conversation_id": "1570117203583696896",
            "created_at": "2022-09-15T18:20:07.000Z"
        },
        {
            "id": "1570477470213611521",
            "text": "@RoyalFamily https://t.co/Cv9pKjeuKv",
            "author_id": "2797734059",
            "referenced_tweets": [
                {
                    "type": "quoted",
                    "id": "1570413149651869696"
                },
                {
                    "type": "replied_to",
                    "id": "1570117203583696896"
                }
            ],
            "conversation_id": "1570117203583696896",
            "created_at": "2022-09-15T18:19:41.000Z"
        }
    ],
    "includes": {
        "users": [
```

```
            {
                "id": "1564165997430538240",
                "name": "Michael Iheakachi",
                "username": "MichaelyoungL"
            },
            {
                "id": "2797734059",
                "name": "Aman Pnjr",
                "username": "ahmadi223"
            }
        ],
        "tweets": [
            {
                "id": "1570117203583696896",
                "text": "https://t.co/U5ph5hcVdg",
                "author_id": "36042554",
                "conversation_id": "1570117203583696896",
                "created_at": "2022-09-14T18:28:07.000Z"
            },
            {
                "id": "1570413149651869696",
                "text": "Taliban killing yesterday https://
                  t.co/FMU47UzW9Y",
                "author_id": "2797734059",
                "conversation_id": "1570413149651869696",
                "created_at": "2022-09-15T14:04:06.000Z"
            }
        ]
    },
    "meta": {
        "newest_id": "1570477741673447425",
        "oldest_id": "1570474128486072321",
        "result_count": 10,
        "next_token": "b26v89c19zqg8o3fpzbjo31yst8l0cecs3c7
          u6l3dziil"
    }
}
```

Listing 2: Node-Link JSON sample

```
{
  "nodes": [
    {
      "id": "1563573550027460608",
      "author_id": "300390462",
      "name": "Guillermo Lasso",
      "username": "LassoGuillermo",
      "time": "2022-08-27T17:05:58.000Z",
      "text": "Mi solidaridad a la familia del @EjercitoECU
          y a la familia de los militares que cumpliendo su
         deber fallecieron. Dios los tenga en su gloria.
         https://t.co/sBHIqPcUaK",
      "conversation_id": "1563573550027460608"
    },
    {
      "id": "1567844545546620929",
      "author_id": "1402444288882008064",
      "name": "Jorge",
      "username": "Jorge79851353",
      "time": "2022-09-08T11:57:23.000Z",
      "text": "@LassoGuillermo @EjercitoECU https://t.co/
         nAj3K6fmvA",
      "conversation_id": "1563573550027460608"
    },
    {
      "id": "1566818035754745862",
      "author_id": "1440346695158550535",
      "name": "Guillermo Eduardo Romero Ruiz",
      "username": "Guiller05757360",
      "time": "2022-09-05T15:58:24.000Z",
      "text": "@LassoGuillermo @EjercitoECU Este gobierno
         nadie le cree el dolor de los dem s es un gobierno
          corrupto",
      "conversation_id": "1563573550027460608"
```

```
    }
  ],
  "links": [
    {
      "source": "1563573550027460608",
      "target": "1567844545546620929",
      "type_tw": "replied_to"
    },
    {
      "source": "1563573550027460608",
      "target": "1566818035754745862",
      "type_tw": "replied_to"
    },
    {
      "source": "1566430332240404480",
      "target": "1566469413175529475",
      "type_tw": "quoted"
    },
    {
      "source": "1563573550027460608",
      "target": "1566469413175529475",
      "type_tw": "replied_to"
    },
    {
      "source": "1563573550027460608",
      "target": "1565144542151692289",
      "type_tw": "replied_to"
    }
  ]
}
```