UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Study of Visual Clutter in Geographic Node-Link Diagrams

Alberto Debiasi

Advisor

Raffaele De Amicis

Fondazione Graphitech

March 2016

Abstract

An increasing amount of geographic data is now freely available on the Internet, and this number is expected to increase as monitoring systems and sensors are becoming a ubiquitous part of our environment. A substantial subset is structured as networks (or graphs). Notable examples are import/export of goods, networks of climate stations, flight connections, migration flows and Internet traffic. To fully comprehend such networks, it is needed to know both their geographic and relational information. For this purpose, the most common visual representation is the node-link diagram, where vertices are depicted as points and the edges connecting them are drawn as lines. This Thesis focuses on an instance of node-link diagrams, where nodes are over-imposed on a map and fixed according to their geographical information. The major issue of this visualization is the visual clutter of nodes or edges, defined in literature as "the state in which excess items, or their representation or organization, lead to a degradation of performance at some task". In particular, nodes and links may cause occlusion and ambiguity in the graph representation. Such problems characterize a cluttered diagram because they reduce the potential usefulness of the visualization. The goal of this thesis is to advance the state of the art in Graph Visualization with respect to visual clutter. On the theoretical perspective, our goal is to acquire a deep understanding of existing approaches for visualizing geographic networks and for reducing visual clutter. Initially, we provide a classification of geographic node-link diagrams and a survey on techniques to reduce the visual clutter on such visualizations. Afterwards, we present a schematization of techniques that helps the reader to decide, given a task and a geographical node-link diagram, which are the candidate solutions that help to reduce the visual clutter. On the practical perspective, our goal is the development of visualization and interaction techniques to overcome various issues of the state-of-the-art approaches. On the one hand, we present an interactive lens that faces the issues of links organization over a map. On the other hand, we describe a deformation-based technique that reveals nodes and links otherwise hidden behind the globe surface. Finally, we introduce a method that automatically generates flow map layouts starting from multivariate geographical datasets.

Keywords
Graph Visualization, Visual Clutter, Graph Drawing Techniques, Node-Link Diagram, Geographic Networks.

Acknowledgements

This Thesis is the result of my research work at Fondazione GraphiTech for the past four years. I would like to express my gratitude to all those who helped me in conducting this Thesis. A special thank you goes to my supervisor Raffaele De Amicis for his steady support and encouragement, giving me the freedom to choose the direction of my research work. I also thank my (past and current) colleagues in Fondazione GraphiTech who have contributed to create a great and inspiring working environment. In particular, I thank Bruno Simões that actively contributed as author in most of the papers I wrote. He also supported me with helpful comments in the late afternoons in GraphiTech.

I would like to express my deepest appreciation to Jüergen Döellner for giving me the opportunity to spend three months at the Hasso Plattner Institute as visiting student. In that period, I have learned more about GPU programming than I have in my entire life. In particular, I thank Stefan Buschmann, Willy Scheibel, and Daniel Limberger for their great technical support.

I owe sincere thankfulness to all the anonymous reviewers for their suggestions to improve my work. I also thank Hans-Jörg Schulz, Ben Shneiderman, Christian Tominski, Emmanuel Pietriga, Ulrik Brandes, and Felix Brodkorb who kindly responded to my requests for feedback via email. Many thanks are also due to my committee members Alfredo Liverani, Jürgen Döllner, and Michele Fiorentino for their valuable time and constructive feedback on my thesis.

Lastly, and most importantly, I want to thank my father Diego, my mother Giovanna, my brother Stefano and my beloved girlfriend Elisa for their continual understanding and encouragement when it was most required. This thesis could not have been written without your support.

Contents

Chapter 1

Introduction

The eye... the window of the soul,
is the principal means by which the central sense
can most completely and abundantly appreciate
the infinite works of nature.
–Leonardo Da Vinci

This chapter gives an overview of geographic networks in the Information Visualization domain, introduces the motivations and the challenges, and summarizes the contributions of this thesis.

1.1 Geographic Networks

In the world of today many things which take place in reality are represented as data. In 2012, it was estimated by the IDC (International Data Corporation) that the world has generated 1.8 zettabytes of data and the amount is expected to increase at a rate of 59 percent annually. According to that analysis, the digital universe will reach 40 zettabytes by 2020 [68].

A substantial amount of data, collected by scientific and governmental institutions, private companies, and individuals includes geographic referencing[1]. Examples of such data occur in everyday events: a credit card purchase transactions has associated the location of the purchaser and the purchase, telephone records have associated the addresses of the caller and the callee and the locations of cell phone zones, census data include addresses. This work focuses on *geographic networks*, defined as "networks with associated geographic information[2]" [33]. Notable examples are import/export of goods (also named supply chain networks), networks of climate stations, flight connections, migration flows

[1] The estimated percentage of data with geographical reference is among the 80% and 95% of the total [78, 77, 194].

[2] In this thesis, we consider synonyms the terms *graph* and *network* and the terms *geographic*, *geo-located* and *geo-referenced.*

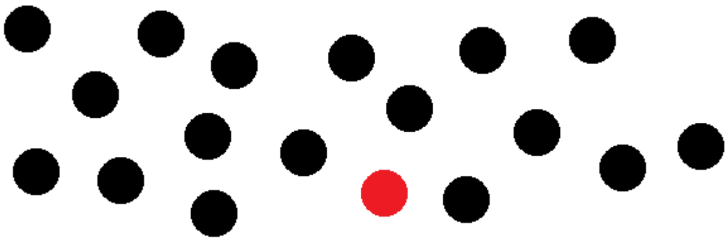


Figure 1.1: Example of preattentive processing - the red circle ”pops out” of the display.

and Internet traffic. To fully comprehend such networks it is needed to know both their geographic and relational information.

In many fields, the rate at which data is collected and stored exceeds the human ability for decisions making. This phenomena has been termed *information glut* [138] and *information overload* [96]. Overall, the study on the data representation allows us to amplify the cognition to extract knowledge.

1.2 Information Visualization

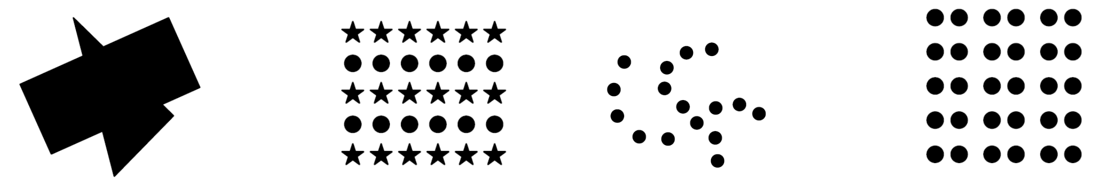
Information Visualization is defined as “the use of computer-supported, interactive, visual representations of abstract data to amplify cognition” [29]. In this definition, *abstract data* include both numerical and non-numerical data, such as geographic information and text. *Cognition* is the ability to understand the interpreted images, making conclusions mainly based on prior learning.

Compared to any of our senses, the vision is the most appropriate to carrying information to the brain because it has the highest bandwidth (i.e. the amount of data in a range of time)[3]. This motivates the fact that “seeing” is a synonym of “understanding” in English. As stated by Ware [181], the effectiveness of the vision for the perception of shapes and features is mainly explained in two psychological theories: *Preattentive processing theory* [169] and *Gestalt theory* [100].

Preattentive processing theory, at the low level, explains that visual features such as closure, color, curvature, line length or line orientation are preattentively processed, see Figure 1.1. Such processes are extremely important when the main task is the visual search.

Gestalt theory, at a higher cognitive level, describes the brain principles aimed to understand an image [181]. Visual perception is considered a complex process where human beings tend to perceive simple geometric forms. This suggests that the structure

[3] Audition has a bandwidth around 100 b/s, meanwhile vision has 100 Mb/s [181].



(a) The Law of Simplicity (b) The Law of Similarity (c) The Law of Continuity (d) The Law of Proximity

Figure 1.2: The image shows a subset of Gestalt Laws.

composing a visual display is more important than the elements, that is summarized as "the whole is greater than the sum of its parts" [181]. Gestalt theory relies on a set of laws (or principles) that can facilitate the reasoning processes in visual displays. In fact, such processes are strongly influenced by visual characteristics, such as joining lines or bright colors, see Figure 1.2.

Visualization, defined as the graphical representation of data or concepts [97], helps users to carry out different classes of cognitive processes [27, 98, 148]:

- *Exploratory.* The user is not aware on what he/she is looking for. In other terms, the person may not have a question in mind or a specific goal. Instead, he/she simply examines the data to make new discoveries, or simply to learn more about it [63, 159]. The development of human-computer interface equipments and advanced graphics cards in the mid-1990s, has further facilitated exploratory tasks of global and local data features [181].

- *Analytical.* The user is aware on what is looking for in the data, searching for confirmation. The hypotheses verification may be accomplished by automatic techniques from machine learning or statistics [63, 98].

- *Descriptive.* The user knows the phenomenon observed in the data, but he needs an effective visual representation to transfer knowledge. This process may involve different target audiences such as decision makers, domain experts, or the general public [159].

1.2.1 InfoVis Pipeline

Usually, the designer of a new visualization analyses the dataset available and the type of information the viewer wants to extract from, or convey with, the display [96]. The data can have heterogeneous sources and complex structure. As mentioned above, the viewer may use the visualization for exploration (looking for patterns or anomalies), to confirm

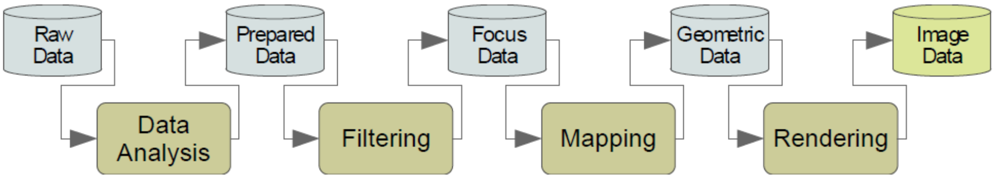


Figure 1.3: The visualization pipeline [52].

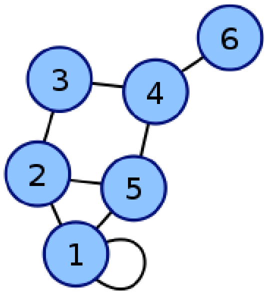
a hypothesis (either conjectured or the result of quantitative analysis), or to present the results of analysis to an audience.

As shown in Figure 1.3, to visualize a dataset, the latter is processed through the *visualization pipeline* [180]:

- In the *data analysis step*, processes such as correction of erroneous measurements, smooth filter, and interpolation of missing values are used to prepare the data for visualization. This step can be described as computer-centered, since it involves only little to no interaction with users [52].
- The *filtering step* performs a selection of data to be visualized. The selected portions are denoted as focus data. Usually, this step is user-centered.
- In the *mapping step*, focus data are mapped to geometric primitives (e.g., points, lines) and their visual attributes (e.g. size, position, color). This step is of high interest to the designer because it is the most important to enhance expressiveness and effectiveness in the visualization.
- In the *rendering step*, geometric data are transformed to image data. Here, adequate rendering techniques are used to create the final visual representations (e.g. images or animations).

While early visualizations were static objects, printed on paper or on other fixed media, modern visualizations are the results of very dynamic processes, allowing the user to control virtually the entire procedure, from data selection and mapping control to color manipulation and view refinement. Overall, the *interaction* is the element that brings different advantages on modern visualizations for the following reasons [120]:

- When dealing with huge datasets, the limitations of both our visual system and displays preclude just showing everything at once; interaction allows the user to update the view.
- A single static view can show only one aspect of the dataset; interaction enables the support of many possible queries, changing the display.



(a) Node-link diagram

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

(b) Adjacency matrix

Figure 1.4: Two visual representations of the same graph.

- Interaction can present different data representations, supporting the understanding of the connections between these alternatives.

The effectiveness of a given visualization mainly depends on the user. Different users, with different backgrounds, perceptual abilities, and preferences, will have differing opinions on each visualization. Moreover, the user's task will affect the usefulness of the visualization. Even a change in the data being visualized can have implications on the resulting visualization. Thus, it is critical to enable users to customize, modify, and interactively refine visualizations until they feel they have achieved their goal, such as extracting a complete and accurate description of the data contents or presenting a clear depiction of patterns that they want to convey.

1.2.2 Graph Drawing and Graph Visualization

The processes and methods for converting a graph to its geometric form are called *graph drawings* [11]. The development of computer technology made it possible to automatically draw graphs. The most common visual representation is the *node-link diagram*, where vertices are depicted as points and the links connecting them are drawn as lines [49], see Figure 1.4(a). Apart from representing graphs as node-link diagrams, we could also use a more compact visualization such as the *adjacency matrix*, where each cell is used to indicate whether the nodes, identified respectively by the row and column, are connected or not. This representation was initially introduced by Bertin [14] and popularized later by Becker et al. [13]. It is very effective for dense graphs and it reduces visual clutter to minimum, see Figure 1.4(b). As drawback, this technique does not facilitate the solving of path-related tasks.

Graph Visualization and *Graph Drawing* are disciplines relied on the visualization of graphs. Graph Drawing discipline studies the readability of graphs, in particular when they are represented as node-link diagrams. In such cases, the readability issue is expressed in terms of *aesthetic criteria*, formulated as optimization goals for the graph drawings [48]. Some of the most important criteria are: minimize number of edge crossings, minimize edge length, minimize number of edge bends, minimize node overlap, minimize drawing area, and maximize symmetry [162]. Although these criteria are generally accepted, they present some limitations. First, they cannot be simultaneously optimized. Second, they are not enough to design an appropriate layout mainly because it is usually task and data dependent [177]. Moreover, such criteria are not satisfiable in large graphs.

Graph Visualization discipline, that is a branch of Information Visualization, is mainly characterized by the problem of scalability [81]. The latter involves three different kinds of limitations [120]: computational capacity, human perceptual and cognitive capacity, and display capacity. An example of limited computational capacity is when large datasets do not fit into the computer memory. Regarding our perception system and cognitive capacity, the main limitation is that we store only little information internally in visual working memory. Finally, the main limitation of the screen is its resolution that is not enough to show all the desired information simultaneously.

When involved operations are the collection, the organization, the manipulation of relational data and the presentation of the results, rarely network visualization is a complete application by itself. In most of cases, it is a feature of a complex system. In summary, the main challenge is to find applications where network visualization creates enough value, either as part of system or as a standalone application, that allow the user to gain insight into relational data [157].

1.3 The Motivation

Geographic networks differ from abstract networks because they cannot be fully comprehended without reference to their spatial component. Examples of such networks are road maps [84], migration flows [129], Internet traffics [33] and commercial trading [113]. For such datasets, *geographic node-link diagram*, i.e. node-link diagram that encodes both relational information and geographic information, can be an appropriate visualization.

In such context, as stated by Peter Rodgers [141], graph drawings are considered important in different cases:

- **Overlap of links.** To distinguish close links, graph drawings may provide techniques to reroute them in the most understandable way. The visualization of train networks [21] is a concrete example.
- **Undesired spatial arrangement of nodes.** There are cases, such as underground railway maps, where the spatial information is important, but is not desirable to lay-

out the nodes according to their exact position. In these cases, graph drawing may provide techniques to rearrange nodes, preserving at the same time their relationships.

- **Unimportant geographical information.** For some geographic networks, the physical location of vertices is not important to extract useful information. In such cases, desirable solutions may provide visualizations based on the topological relationship between items, rather than based on their geographic position [115]. For example, in economic data visualization, countries with strong connections are spatially grouped together.
- **Lack of geographical information.** If vertices do not have a clear physical location (such as for networks of corporations), graph drawing techniques are needed to layout the network. For example, visualizations of Internet typically are not based on spatial information, because it is difficult to extract from web sites precise location information [31, 50].
- **Multivariate datasets.** Graph visualizations are sometimes used together with geographic visualizations, defining the so called *linked multiple views* [137]. On one side, they allow the exploration of spatial information. On the other side, they provide the relational representation of the data.

This thesis focuses on an instance of geographic node-link diagrams, where nodes are over-imposed on a map and fixed according to their geographical information [34]. The major issue of this representation is the *visual clutter* of nodes or edges, defined in literature as "the state in which excess items, or their representation or organization, lead to a degradation of performance at some task" [142]. In particular, nodes and links may cause occlusion and ambiguity in the graph representation. Such problems characterize a cluttered diagram because they reduce the potential usefulness of the visualization [108, 155, 59].

1.4 Contributions and Outline

To move a step forward the solving of visual clutter in geographic node-link diagrams, this thesis provides theoretical and practical contributions. The following are the theoretical contributions:

- A classification of geographic node-link diagrams is presented in Chapter 2. The criterion used in this taxonomy is the way the nodes are spatially organized in the presentation space. This work helps to choose the most appropriate visualization according to the geographic attributes that characterize the dataset in input.

- The aforementioned diagrams tend to suffer from visual clutter. Hence, Chapter 3 describes a survey of the main visualization and interactive techniques designed to face such undesired state. Furthermore, this chapter surveys the main techniques used to support the visual exploration of geographic networks.
- Clutter reduction techniques may focus on different readability problems. Hence, Chapter 4 classifies the effects of the visual clutter (defined as problems) on geographic node-link diagrams. Then, each task group has associated the classes of problems that characterize it. The aim of this work is to provide an answer to the following research question: *Given a task and a cluttered geographic node-link diagram, which are the appropriated techniques to reduce the visual clutter.*

The following are the practical contributions:

- *3DArcLens*, an interactive lens that faces the issues of links arranged over a map, is described in Chapter 5. The aim of *3DArcLens* is to provide an answer to the following research question: *Given a geographic node-link diagram, how can we reduce the effect of clutter in a user-specific area, revealing information about the nodes, the links and the geographic surface, without moving the camera view and with no information loss.* It reduces the visual clutter acting on the shape of the links that are inside the lens. *3DArcLens* is designed to facilitate the visual exploration of geographic networks.
- *GeoPeels*, a deformation-based technique that faces the occlusion of nodes and links by the globe, is described in Chapter 6. *GeoPeels* provides an answer to the following research question: *Given a geographic node-link diagram on a globe, how can we reveal nodes and links of interest that are occluded by the globe shape, without moving the camera view and preserving the geographical context of the items.* It solves the occlusion problem of nodes and links deforming the globe surface. *GeoPeels* is designed to facilitate the visual exploration of global networks.
- An algorithm for the automatic generation of Flow Maps layouts is presented in Chapter 7. It reduces the visual clutter grouping together the flow lines. It is based on a force-directed algorithm. Additionally, it supports for an optional supervision of the graph layout. An extension of such algorithm, designed for the mapping of multi-variate data in a static visualization, is described in Chapter 8.

1.5 Related Publications

The major parts of the chapters outlined above are based on the following published papers:

[45] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “Schematization of Clutter Reduction Techniques in Geographic Node-Link Diagrams using Task-based Criteria.” In *Proceedings of the IVAPP - 7th International Conference on Information Visualization Theory and Applications*, 2016.

Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “Schematization of Node-Link Diagrams and Drawing Techniques for Geo-Referenced Networks.” *Cyberworlds, 2015 International Conference* on. IEEE, pages 34-41, 2015.
They serve as core material for Chapter 4.

[42] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “3DArcLens: Interactive Network Analysis on Geographic Surfaces.” In *Proceedings of the IVAPP - 6th International Conference on Information Visualization Theory and Applications* , pages 291-299, 2015.

[37] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “3DArcLens: A Technique for the Exploration of Geographical Networks.” *Visual Analytics Science and Technology (VAST)*, pages 245-246, 2014.
They serve as core material for Chapter 5.

[44] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “GeoPeels: Deformation-Based Technique for Exploration of Geo-Referenced Networks.” In *Proceedings of the WSGC - 23nd International Conference in Central Europe on Computer Graphics*, pages 53-62, 2015.

[43] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “Deformation-based Lens for the Exploration of Geo-Networks.” In *Posters in the EuroVis - Eurographics/IEEE Conference on Visualization*, 2015.
They serve as core material for Chapter 6.

[41] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “Supervised force directed algorithm for the generation of flow maps.” In *Proceedings of the WSCG - 22nd International Conference in Central Europe on Computer Graphics*, pages 193-202, 2014.

[39] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “Force directed flow map layout.” In *5th International Conference on Information Visualization Theory and Applications*, pages 170-177. SCITEPRESS, 2014.
They serve as core material for Chapter 7.

[38] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “Depiction of Multivariate Data through Flow Maps.” In *Proceedings of the SOMAP - Symposium on*

Service-Oriented Mapping, 2014.

[40] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. “A Novel Method for the Depiction of Multivariate Data through Flow Maps.” In *Poster in the IEEE Scientific Visualization Conference, SciVis*, 2014.

They serve as core material for Chapter 8.

The following published papers are not included in this thesis:

Alberto Debiasi, Federico Prandi, Giuseppe Conti, Raffaele De Amicis, and Radovan Stojanovic.“Visual Analytics Tool For Urban Traffic Simulation” In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, pages 51-56, 2013.

Bruno Simões, Michele Bianchi, Alberto Debiasi, and Raffaele De Amicis. “Serious Games For Large-Scale Image Sensing”. In *Proceedings of the 5th EAI International Conference on Serious Games, Interaction and Simulation*, 2015.

Chapter 2

Geographic Node-Link Diagrams

There is a magic in graphs.
The profile of a curve reveals in a flash a whole situation -
the life history of an epidemic, a panic, or an era of prosperity.
The curve informs the mind, awakens the imagination, convinces.
–Henry D. Hubbard

2.1 Introduction

We define *geographic node-link diagram* as “node-link diagram that visually encodes both relational information and geographic information”. It includes the well-known visualization where nodes are located on top of a map. However, other solutions presented in literature preserve such requirements. For example, approaches where the order of nodes or their label map to geographic information.

On the one hand, the aim of this chapter is to provide a comprehensive introduction to the visualization of geographic networks. Section 2.2 gives the main definitions used in this thesis, in particular those related to datasets characterized by both geographic and relational attributes. Section 2.3 introduces representative examples of geographic networks that can benefits from node-link representation.

On the other hand, the contribution of this chapter relies on a classification-based approach to describe geographic node-link diagrams, see Section 2.4. For this purpose, we extend the taxonomy of Schulz and Schumann [149], in which the general node-link diagrams are classified according to the degree of freedom for node layouts (i.e. Free, Styled and Fixed) within the presentation space. Then, Section 2.5 presents the main instances of geographic visualizations for the considered scenarios. Finally, Section 2.6 ends this chapter with a brief recap.

2.2 Main definitions

The *graph* is an abstract structure used to represent relational information. It can be defined as ordered pair composed to a set of vertices and a set of edges, where each edge is related to two vertices [49]. Graphs are often classified into *directed* and *undirected* [81]. For a directed graph (resp. undirected), the edge vertices are ordered (resp. unordered). A *path* is a sequence of connected vertices. A *cycle* is a closed path that has the first vertex equal to the last vertex. A *tree* is a connected undirected graph without cycles [49]. A tree is called *rooted* (or hierarchy) when one vertex is distinguished as a so-called root node [177].

Vertices and edges may have attributes associated. An *attribute* is some specific property that can be measured, observed, or logged [120]. As shown in Figure 2.1, the major distinction of attributes is between *categorical* and *ordered* [120, 28]:

- Categorical: It does not have an implicit ordering. Categories can only distinguish whether two things are the same or different.
- Ordered: All ordered data does have an implicit ordering, as opposed to unordered categorical data. This type can be further subdivided.
  - Ordinal data: It is not possible to perform arithmetic comparison among data, but there is a well-defined ordering.
  - Quantitative data: It is a measurement of magnitude that supports arithmetic comparison. Both integers and real numbers are quantitative data.

In some domains, the quantitative category is further split into *interval* versus *ratio* data [154]; this distinction is typically not useful when designing a visual encoding, so we do not consider it.

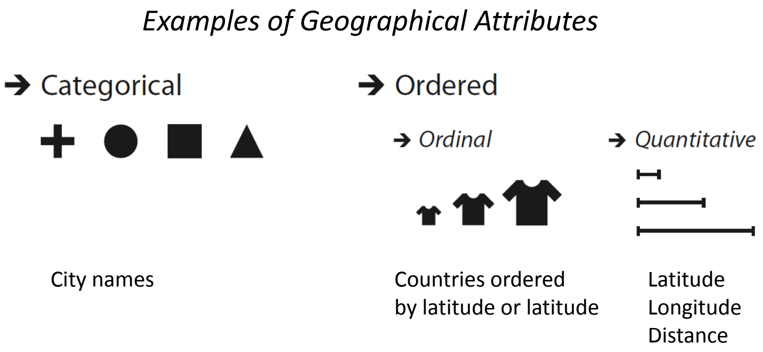


Figure 2.1: Categorical and ordered geographical attributes.

We define *geographical attribute* simply as an instance of general attribute, thus it can be categorical, or ordered. In Figure 2.1 the image from [120] was specifically redesigned to describe geographical attributes. Furthermore, we define *global networks* as networks with geographical attributes not restricted to a region of the Earth. In other words, they span the entire Earth.

2.3 Representative Scenarios

From the information visualization community, there are different scenarios studied where data is structured as geographic network. The following are representative examples.

Learn about subway. Citizen or tourist with no familiarity with a metro needs indications to reach his/her destination as faster as possible. This scenario is extracted from [185].

Geographic undirected network. Vertices are metro stations and edges (undirected) are metro lines connecting the stations. The number of vertices and edges are in the order of tens to hundreds. The geographical attribute of vertices is the spatial location of stations and the geographical attributes of edges are subway tracks positions.

Learn about migrants flows. Citizens want to be aware of migration flows among countries. Such information are sometimes retrieved via newspapers or Internet. This scenario is extracted from [161, 18, 73].

Geographic rooted directed trees. Vertices are countries and edges (directed) represent migrants. The geographical attribute of vertices is the related country. Other attributes associated to migrants are age, ethnicity, occupation. Each tree represents the (in-coming/out-coming) flows from a specific country (the root). An instance of this dataset is visualized in Chapter 7 and 8.

Monitor network infrastructure. Maintainers need to accurately understand the large-scale geographic structure of the network infrastructure. This scenario is extracted from [118, 33].

Geographic undirected network. Vertices are routers and edges (undirected) represent the direct connections between routers. The geographical attribute of vertices is the physical location of routers. An instance of this dataset is visualized in Chapter 5 and 6.

Analyze climate networks. Climate scientists want to investigate the information of interest contained in the spatial correlation of climate data. This scenario is extracted from [165].

Geographic undirected network. The graph represents the structure of statistical relationships between observations. Vertices represent grid points or measurement stations, like precipitation or temperature. Edges (undirected) represent statistical associations between two measurements. The vertex attributes are *degree centrality*, i.e. the number of its first neighbors, and *betweenness centrality*, i.e. vertex centrality measurement based on shortest path lengths. The edge attributes are *edge betweenness*. The geographical attribute of vertices is the position of the stations.

Analyze airplane trajectories and airport connections. Airplane trajectories recorded by the air traffic authorities enable to analyze traffic or devise new ways of managing airspace. Such data can be used for different purposes, for example it can alter how epidemic disease spreads. This scenario is extracted from [83, 150].

Geographic directed network. The graph represents the air traffic. Vertices are cities and edges are direct flights. The edges are derived from actual flight paths; since flights start and end in airports, edge endpoints are spatially grouped. An instance of this dataset is visualized in Chapter 6.

Analyze mobility data. Decision makers and urban planner want to understand the movement behavior of people and its dynamics. Data on the movement of people can be obtained from traffic sensors, public transportation usage records or reconstructed from tracking devices such as mobile phone and GPS receivers. This scenario is extracted from [176, 74].

Geographic directed network. The graph represents traffic flows. Vertices are geographical regions and edges are the flows between them. Time-based attributes of the flows create dynamic geographic graphs. It is known only the starting and ending locations of the trips, and not the exact routes of people movements.

Analyze friendship connections. Social scientists want to analyze friendships or connections between teams. Their aim is to gain knowledge about social and spatial structures of the network in focus. This scenario is extracted from [175].

Geographic undirected network. Vertices are persons and edges are the friendships. The geographical attribute of vertices is the home location of the persons.

2.4 Classification on Node Arrangement

The criterion used in this taxonomy is the way the nodes are spatially organized in the presentation space. We distinguish the following node arrangements: *fixed*, *styled*, *free*, and *hybrid*.

2.4.1 Fixed Node Arrangement

This class contains the diagrams with nodes fixed in a *geographic space*, that is the space of the visualization used to give a geographical reference to the visual features, mapping their position expressed in latitude and longitude to the coordinate axes of the drawing area.

These graph visualizations are recommended to effectively show the latitude and longitude (quantitative attributes) of nodes. The techniques described in the next chapters focus on Fixed node arrangement Visualizations. A classification of 2D and 3D visualizations of spatial data is described in [53]. The authors focused on the dimensional aspects of two presentation spaces: the attribute space and the reference space. As opposite, we propose two alternative categorizations for this class of visualizations:

- *Flat Maps vs. Globes*
- *Geometric vs. Semantic Distortion*

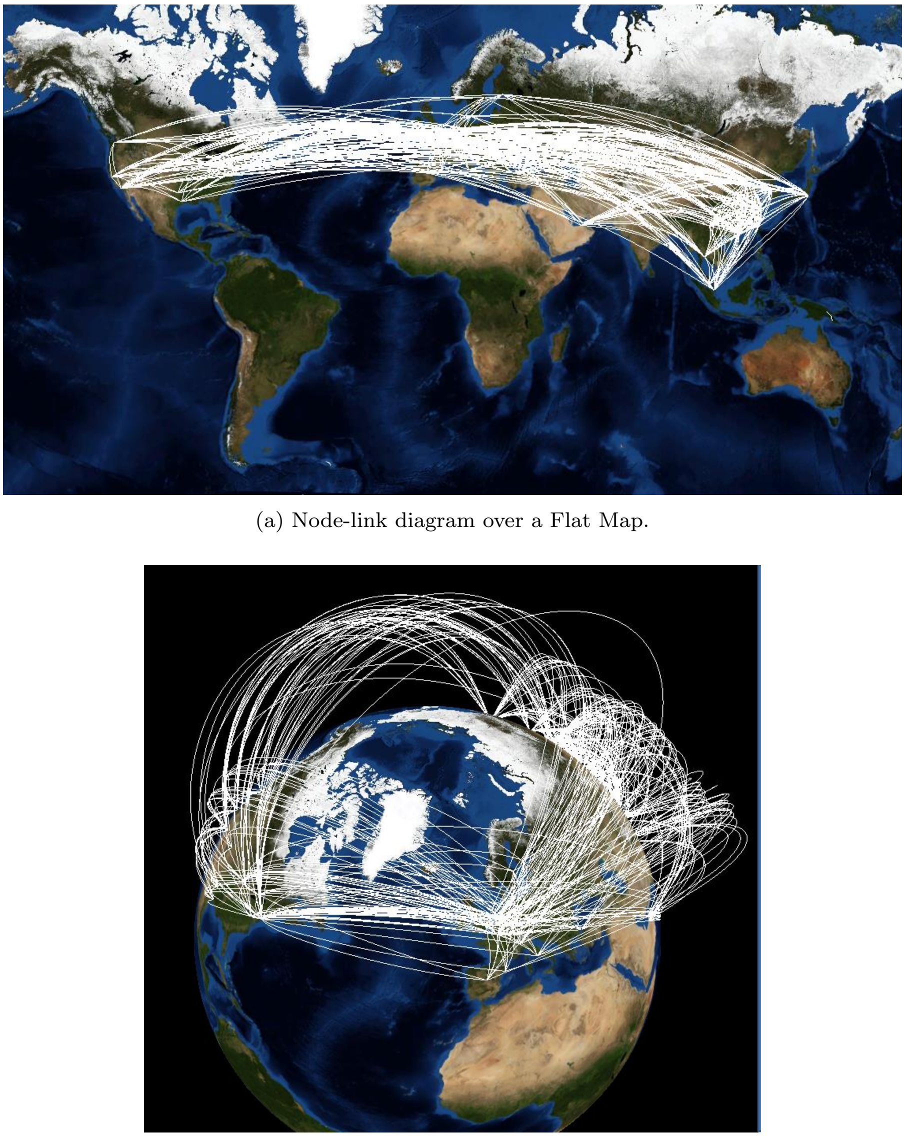
Flat maps vs. Globes. Geographic networks are usually organized in a flat map, see Figure 2.2(a). The latter involves the visualizations where two axes of the geographical space are demanded to map latitude and longitude. As drawback, in case of global networks, nodes that are geographically close may reside at the opposite sides of the flat map. Consequently, their links would cross the entire map giving a misleading impression of the actual geographic distance between nodes [26]. One possible solution is a 2D map projected over an imaginary infinite plan. However, such strategy may cause redundant information related to the nodes and the arcs.

Globes reflect the actual shape of our world and are usually used to represent global networks, , see Figure 2.2(b). In globes there are three axes used to map latitude and longitude. Up to now, the term “globe” is referred as the *virtual hyperglobe* [86], i.e. an immaterial digital globe projected over a 2D screen. As opposite, we do not consider the *tactile hyperglobes* and the *hologlobes* that are respectively a material and an immaterial globe both located in the real space.

Compared to flat maps, globes enable continuous navigation, preserving the spatial coherence of nodes and links [53]. With respect to general 3D visualizations, globes allow the user little chance of becoming disoriented [13, 57].

As drawback, projecting a 3D representation onto a 2D plane make distances hard to estimate [125], because of perspective foreshortening [168]. Another limitation is the occlusion of nodes and links caused by the globe shape. In fact, at most one hemisphere is visible at a time. It can be perceived as an advantage; the globe works as an explicit visual filter over the nodes, reducing the clutter [118].

Geometric vs. Semantic Distortion. A geographic distortion is *geometric* if it is caused only by the chosen map projection. A *map projection* is a systematic transformation of the coordinates, expressed in latitude and longitude, of points on the surface of a sphere or ellipsoid into points on a plane [152]. For virtual globes, the geometric distortion is the creation process of the 3D geometry. This section describes only the main representative projections. Each of them has advantages and limitations.



(a) Node-link diagram over a Flat Map.

(b) Node-link diagram over a Globe.

Figure 2.2: Examples of fixed node layout.

The widely-used *Mercator* projection, which in principles was designed to display compass bearings for sea travel, preserves angles (i.e. angles on the map match angles on the ground). In fact, the routes of ships, that are composed to lines of constant direction (*rhumb lines*), are shown as straight lines. Meanwhile, the shortest distances between two points (*great circle paths*) are shown as curved lines and the area distortions are significant towards the polar regions. Maps that preserve angles are useful also for displaying the flow of atmospheric or oceanic currents. In the majority of Web-based mapping programs, such as Google Maps and OpenStreetMap, the projection used is the *Web Mer-*

cator. It is a variant of the Mercator projection that uses spherical formulas to simplify the computation.

The *Azimuthal orthographic* projection is mainly used for illustration purposes, since it shows the Earth as seen from space infinitely far away. The availability of high-quality imagery of the Earth and other planets as seen from orbit, has caused a recent revival of interest for this projection. NASA World Wind and Google Earth use a similar projection called the *general perspective* where the point of view is a finite distance, enabling a wide variety of interactive pan and zoom operations.

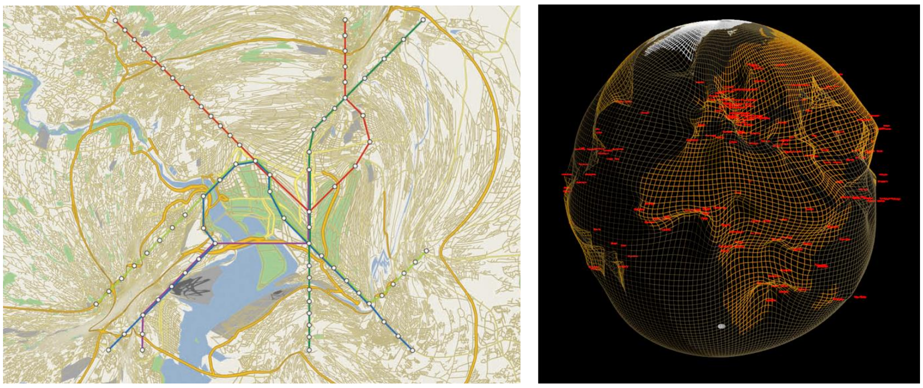
The *Lambert azimuthal equal-area* projection preserves areas and at the same time maintains a true direction from the center. It is recommended to the European Commission for statistical analysis, such as the study of air pollution, ice, and population density. It can used for example to create choropleth maps (showing shaded regions normalized by area) or dot maps (showing dots used to represent the density).

In *Azimuthal equidistant* projection, the distances and the bearing from the center of the map to any point are correct. Thus, this projection is useful to analyze the distance between nodes of a network. It is often used to show distances of air-routes, placing the airport of interest in the center of the map. Similarly, the *Two-Point Equidistant projection* preserves true scale from two specified points on the projection to any point on the map. This projection could be used to show the distance of a ship from the origin and destination of a journey.

In *gnomonic projection*, distortions of distance, shape, area, and direction are severe. As advantage, all great circles are shown as straight lines. As all azimuthal maps, only from the center of the map the direction of the shortest route (azimuth) is shown by a straight line. The projection is useful for defining routes of air travel, because the shortest route between any two locations is a always a straight line.

Van Wijk presented *Myriahedral projections* [173], that rely on methods for mapping the earth on a polyhedron with a large number of faces. The polyhedron is then cut open and unfolded. The result are maps that conserve areas and that are (almost) conformal. As drawback, they are characterized by a large number of discontinuities. For a comprehensive survey on map projection see [111].

A geographic distortion is *semantic* if information on nodes defines the geographic space. For example, the nodes position is updated such that the distance between each node is similar. This effect is more evident if the map is depicted as background, see Figure 2.3(a). Semantic geographic distortion can even be applied on globe. For example, in [6], the nodes connected with each other are rearranged using a force-directed technique such that they appear closer. The globe is consequently deformed to preserve the geographical context of the nodes, see Figure 2.3(b).



(a) Semantic distorted Flat Map. Source: [15]. (b) Semantic distorted Globe. Source: [6].

Figure 2.3: Examples of visualizations where the nodes position affects the geographical space.

2.4.2 Styled Node Arrangement

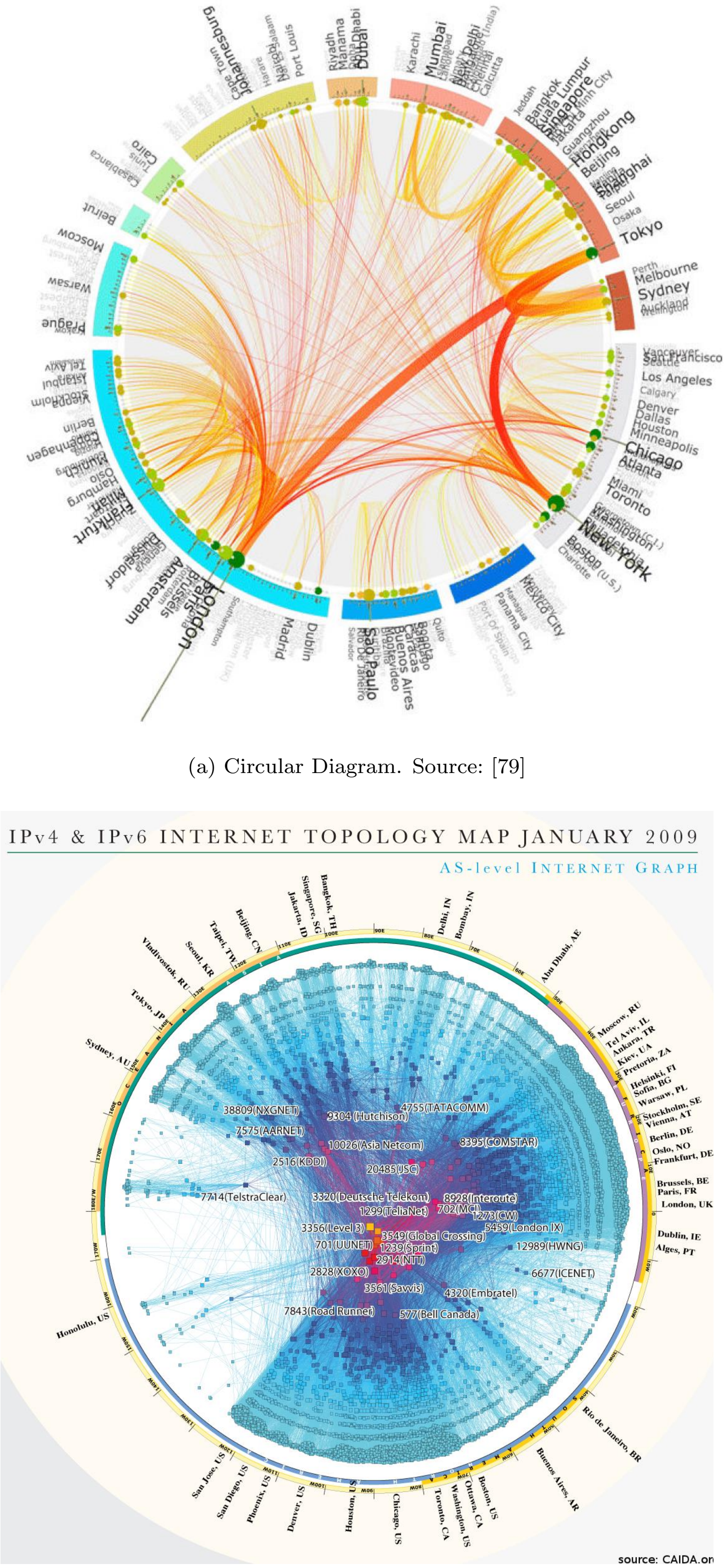
In this class, node arrangement is confined to a certain predefined scheme. They are considered valid alternatives to diagrams with fixed node arrangement if the latter are susceptible to visual clutter. They do not preserve the entire geographical context, but the order of vertices by one or more attributes, such as latitude and longitude. With *Circular Layout* [79] or *ArcDiagram* [182] it is possible to preserve the spatial closeness of nodes. However, labels are needed to extract the geographic location of nodes. For example, in Figure 2.4(a) the nodes representing the countries are ordered along a circle according to their geographic closeness. In terms of readability, no node is occluded by other nodes or links.

Figure 2.4(b) represents a sample of IPv4 and IPv6 Internet topology registered in 2009 [2]. The visualization shows both the interconnectivity of nodes of the Internet routing system and their spatial structure. The location of each node is specified in polar coordinates (i.e. radius and angle). The radius encodes the traffic degree of the node and the angle depends on its longitude. Hence, nodes with nearby/same degree and longitude will overlap.

With those visualizations, the style of the graph representation is known at priori. In this way it is possible to estimate the screen space needed to display the graph and preselect effective graph drawing techniques.

2.4.3 Free Node Arrangement

In this class, nodes are not restricted in space. These graph visualizations are recommended to encode geographical attributes such as city or county names (categorical attributes). The variables used are labels or colors. Thus, all the graph visualization tech-



(a) Circular Diagram. Source: [79]

(b) CAIDA Visualization. Source: [2]

Figure 2.4: Examples of styled node arrangement.

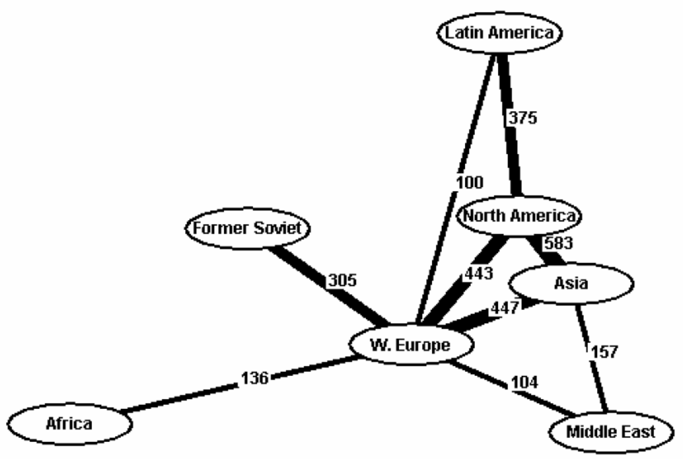


Figure 2.5: Example of Free node arrangement created with a force-directed graph drawing algorithm. Source: [141].

niques that improve the readability re-arranging the nodes can be used. Such visualizations give less attention to the location of items and focus more on their relationship [141]. For example, economic data might be depicted by spatially grouping countries that are strongly connected. Figure 2.5 shows a graph drawn using a force-directed technique. The edges represent the main trade in goods between regions of the world in 2001.

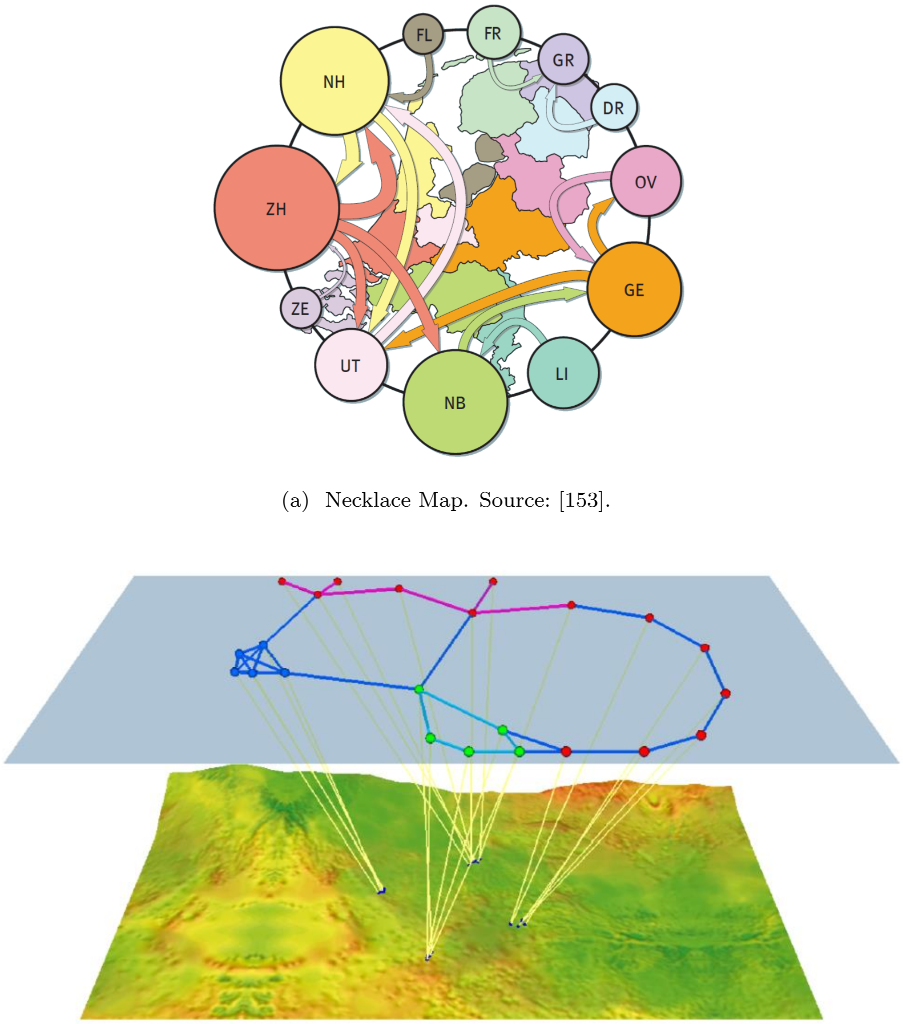
2.4.4 Hybrid Node Arrangement

In this class, the visualizations combine the above mentioned classes gaining their advantages. The introduction of additional items or the use of visual variables are needed to associate one layout with the other. For example, *Necklace Map* [153] mixes a circular layout (Styled Node arrangement) with a colored background map (Fixed Node arrangement), see Figure 2.6(a). It combines the advantage of having no overlapping nodes with the effectiveness of having the geographic space. The color is used to connect the two layouts. In this case the nodes represent areas and not points. In Figure 2.6(b), a force-directed layout (Free Node arrangement) is mixed with a background map (Fixed Node arrangement) [36]. In this case, additional links are used to connect the two layouts.

2.5 Representative Visualizations

Data transformation processes the raw data into a form suitable for visualization. This section introduces the main visualizations that are mentioned in this thesis. Each visualization is used to represent one or more of the graphs described in the Section 2.3.

For the metro scenario, the network is often represented as a *Metro Map Layout*, where



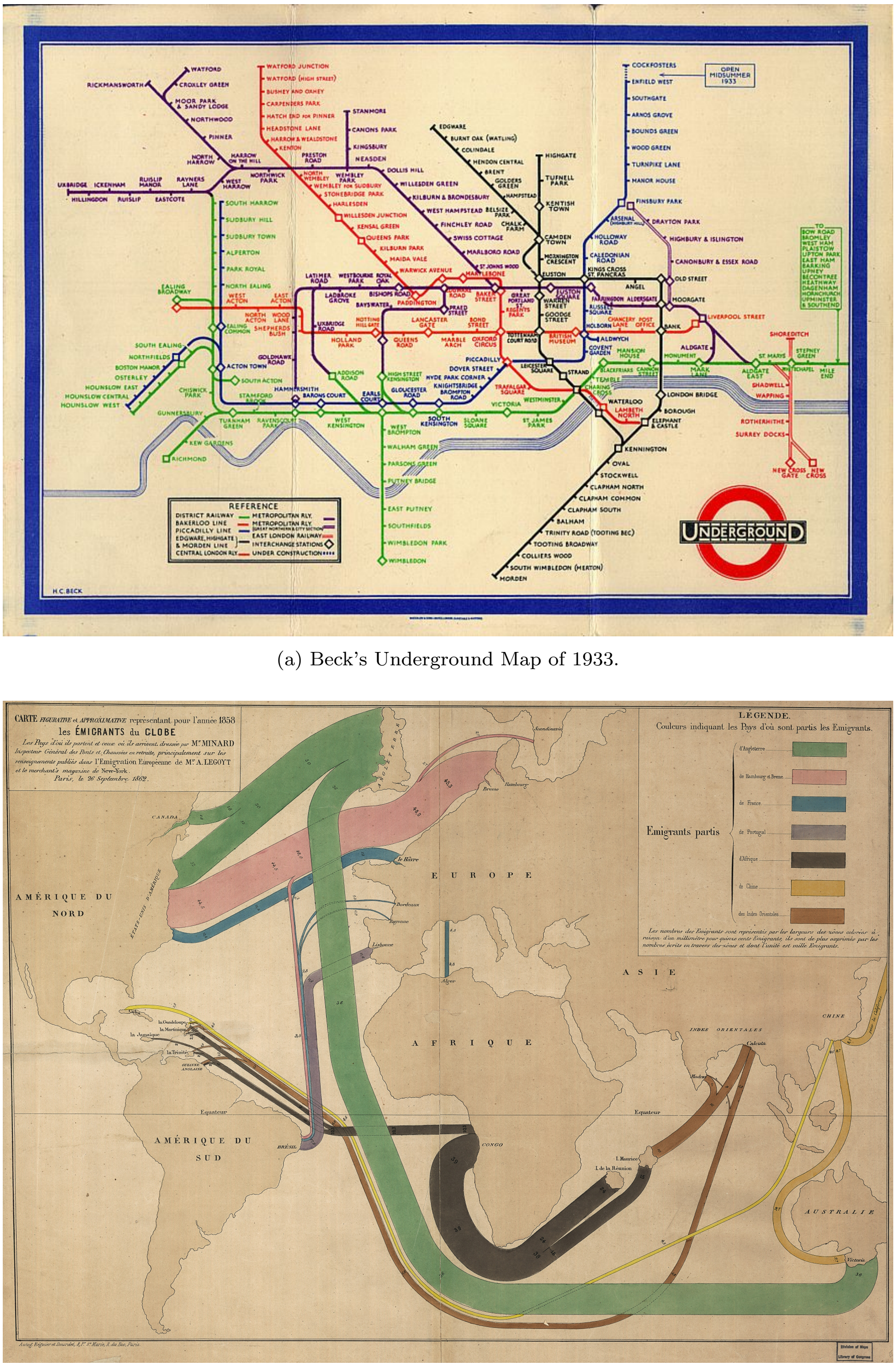
(a) Necklace Map. Source: [153].

(b) 2.5D interface. Source: [36].

Figure 2.6: Examples of hybrid node arrangement.

each metro line has a color associated. They are usually straightened and restricted to verticals, horizontals, and diagonals at 45°. Metro Map Layouts use very different scales for downtown versus suburban areas. In this way distances between adjacent stations are more uniform [84]. The first metro map was designed in 1933 by an engineering draftsman called Harry Beck for the London Underground [184], see Figure 2.7(a).

To illustrate migrants flows, the network can be represented as a *Flow Map Layout*, where links sharing the same destination (also named *flow lines*) are aggregated and their thickness represents the sum of the moving migrants (also named *flow magnitude*). The first flow map representation was created by Henry Harness in 1837 [139] and then popularized by Charles Minard around 1850 [140]. Minard is considered a pioneer in the



(a) Beck's Underground Map of 1933.

(b) Minard's Map of World Migration of 1858.

Figure 2.7: Examples of popular hand drawn maps. They are instances of geographic node-link diagrams.

use of graphics in engineering and statistics, mainly due to his map on the subject of Napoleon's disastrous Russian campaign of 1812 [112]. Figure 2.7(b) shows the emigrants of the World in 1858. The color of the flow lines identifies their origin.

In the scenario of network infrastructure, nodes are usually located over a map or a globe. In this way, it is easy to detect redundant connections, or to find badly placed connections that might cause distribution inefficiencies by wasting bandwidth. Such representation is also used in the context of climate networks, airport connections, mobility data, and social networks because in all the cases the geographic context is important. However, if they are susceptible to visual clutter, other visualizations are recommended.

2.6 Conclusion

This chapter wants to understand and investigate the main aspects related to the exploration and the depiction of geographical graphs through node-link diagrams. Our contribution relies on the classification of geographic node-link diagram based on the criterion used to arrange nodes on the presentation space. This solution facilitates the choice of an appropriate visualization for a given geographic graph.

Chapter 3

Related Work

If I have seen further than others,
it is by standing upon the shoulders of giants.
–Isaac Newton

There are different approaches that support the visual exploration of geographic networks. This chapter discusses the main techniques surveyed in [81, 177]. Section 3.1 describes the techniques that facilitate the exploration when the presentation space has too much detail or is too large to fit in regular displays. Meanwhile, Section 3.2 focuses on techniques to reduce the visual clutter that is the main aspect of this Thesis.

3.1 Graph Visualization Techniques

Common techniques. One of the most used technique in the context of node-link diagram is the *panning and zooming.* Panning allows the user to move the viewing frame whether there are items off-screen. Zooming in and out enables respectively to explore more in details a portion of the dataset and to view the complete data.

There are also techniques designed to provide awareness of off-screen items, using additional features on the windows edges [75, 12, 191]. However, they do not support for context awareness. The navigation from two connected nodes can be facilitated using *Link sliding* [116]. It allows the user to slide a cursor along the link towards the destination node.

Multi views. In *Overview & Detail*, two distinct visualizations are used to provide respectively an overview of the entire graph (i.e. the bird's eye view) and a detailed view of a subgraph of interest [32]. Brodkorb et al. [22] presented a method to visualize details of small dense areas on the screen, keeping the overall geographic context. It uses the notion of insets, which were used by Ghani et al. [69] to allow off-screen nodes to be visible. *Flowstrates* [18] is designed for the exploration of dynamic directed networks. It uses a map for the origins, another one for the destinations, and a view in the middle to show temporal information. *PolyZoom* [91] enables the exploration of multiscale 2D spaces. In particular, it allows the user to shape areas of increasing magnification into different views. As drawback, in multi views the user needs to mentally combine the different views, increasing his/her cognitive load [72].

Geometric distortion. With zooming, the user loses the overview of the area outside the zoom frame. To avoid such drawback, *focus+context* (or *detail in context*) techniques are designed to focus on a subset of the dataset while showing the context of the entire data. With respect to overview & detail, focus+context embeds detailed information within the context view, with no spatial and temporal separation [32]. Geometric distortion can be applied locally or globally over the entire view.

In the first case, different approaches based on lens metaphor are designed. *Fisheye lens* [107] is able to deform the presentation space to visualize both the overview of the space and a focused regions, enabling a smooth transition between the two areas. Pietriga et al. [130] presented *Sigma Lenses*, a set of interactive lenses that replaces standard optical (space-based) distortions with dynamic translucence (time-based) to transition between focus and context areas. *JellyLenses* [131] are two focus+context techniques that dynamically adapt to the shape of the features of interest, by optimizing the regions in focus, and distorting the area around the region of interest. As opposite, the *Undistort Lens* [23] is an interactive technique that allows the user to get access to an undistorted representation of the interest area within a distorted visualization.

Compared to Overview & Detail techniques, distortion based lenses are less accurate [85]. Moreover, they require the interaction by the user. For large datasets, the visual exploration can be tedious and may require the user to reduce the number of items to visualize [22]. In the case of global distortion, the user interaction is not required. A representative example is *Melange* proposed by Elmqvist et al. [60]. It is a space folding technique such that multiple focus areas became visible at the same time. Böttger et al. [16] described a focus+context method based on a global deformation. It enables the interactive exploration of the Earth's surface from its smallest details to the whole planet.

Semantic distortion. An example of semantic distortion is the Metro Map Layout, where the nodes are not arranged exactly using a map projection. As opposite, their position is updated such that the distance between each node is similar [15]. In this particular case, distortion is not used to improve the visualization of an area of interest, but rather to provide a connection between a geographically accurate map and a metro map. Semantic geographic distortion can even be applied on globe. In [6], the nodes connected with each other are rearranged using a force-directed technique such that they appear closer. The globe is consequently deformed to preserve the geographical context of the nodes.

Furnas [66] presented *Generalized Fisheye View*, a focus+context interface which shows items in the presentation spaces based on a *degree of interest (DOI) function.* The DOI function defines which information has to be displayed in the focus. In context of node-link diagrams, different focus+context lenses exist to improve the readability of the visualization. Sarkar and Brown [145] extended the fisheye views to distort each node of a graph according to the distance between its position and a user's focal point.

Attributes-based techniques. Such techniques facilitate the visual exploration changing one or more of the visual attributes, such as the position, the color or the shape of the items. For example, it is possible to visually distinguish paths of interest from the context, applying to them a unique color that differentiates from background items [116]. *Bring Neighbors Lens* [164] dynamically relocates the nodes connected to the node inside the lens area, such that they all appear visible. Similarly, *Bring & Go* moves the neighbors of a particular node around it and enables the user to navigate to one of these.

3.2 Clutter reduction Techniques

Visual clutter is a common issue in geographic node-link diagrams for small as well as large graphs [117]. In this section, the main solutions are introduced in the context of Information Visualization perspective. Section 3.2.1 describes the techniques used to improve a node-link diagram statically. Section 3.2.2 introduces the interactive techniques.

3.2.1 Static Techniques

Items reduction. A common approach to avoid clutter is through *content filters* [186], i.e. the reduction of the number of items to visualize. The downside of this approach is that we also lose information about purged or less relevant links and nodes [66]. *Clustering* the nodes is an approach often used for huge datasets. Nodes can be clustered according to their associated attributes or according to the graph structure. In the latter, nodes strongly connected can be grouped together. In hierarchical methods [4], a pyramid of simplified versions of a graph is used to browse the data in different level of details.

Color-based approaches. The color of links can be used in different way to face the issues that lead to a clutter layout such as ambiguities in the layout and occlusion of items. Map the orientation of links to the color reduces the ambiguity between crossing links [35]. However, it is not effective on links with small crossing angle, because they have similar colors. Bearing this is mind, Hu et al. [87] and Jianu et al. [92] color each link differently. Meanwhile, in [164], the color of a link is interpolated between the colors of its incident nodes. In both solutions it is easier to perceive the nodes connected to the links. Using transparency [143], short breaks are introduced in full links where crossings occurs. The perception of the whole link is based on the gestalt principle of closure. In this way, ambiguity related to graph topology is reduced. Transparency is also used to reveal hidden elements. In [127, 123], transparency is applied in the middle of links. Similarly, with *half-lines* [13] only the first half of each link is drawn to depict directed edges. With this strategy, the amount of screen space used to show links is reduced. As drawback, it makes hard to determine where links end. In [127], transparency is applied in the same way but on curves instead of straight lines.

Nodes relocation. The optimization of screen space can lead to an improvement of the readability of the visualization. For example, *force-directed algorithms* [94] can be used to rearrange nodes' positions. In a force-directed algorithm, the nodes are represented as particles of a physical system with forces acting between them. At each iteration, the energy of the system changes. The algorithm halts when local minimum of the energy is found. The combination of attractive forces on adjacent nodes, and repulsive forces on all nodes, was first introduced by Eades et al. [55]. A few years later, similar methods were presented as an extension to this idea. As notable example, Kamada and Kawai [93] introduced the idea of using only spring forces between all pairs of nodes, with ideal spring lengths equal to the nodes' graph-theoretic distance.

Links relocation. In *edge bundling*, the links of a graph are bundled together if certain conditions are met. Holten et al. [83] presented a force-directed algorithm in which links are modeled as flexible springs that can attract each other while node positions remain fixed. This algorithm is extended to separate opposite-direction bundles, emphasizing the structure of the graph [150]. Cui et al. [35] described a mesh-based link-clustering method for graphs. Control mesh generation techniques were used to capture the underlying edge patterns and to generate informative and less cluttered layouts. Ersoy et al. [62] created bundled layouts of general graphs, using skeletons as guidelines to bundle similar links. Pupyrev et al. [133] proposed a link routing algorithm based on ordered bundles. With this technique, the links are placed in parallel channels to avoid overlap. The peculiarity of the edge bundling is that it is used to understand the overall patterns but not the single links of the graph because the connections become harder to read.

Approaches aimed to generate flow maps are considered clutter reduction techniques. In [129], flow maps are automatically generated aggregating links and making them curves. The authors defined a binary hierarchical clustering to formulate the layout in a simple and recursive manner. Then, in a post creation phase, users are given the possibility to alter the shape of the flow lines by moving their control points. Verbeek et al. [174] introduced a method to create flow maps using directed angle-restricted *Steiner trees* of minimum length, also known as *spiral trees.* They are designed to avoid crossing nodes, as well as user-specified obstacles. In order to straighten and smoothen the tree, a cost function is defined. At each iteration, a new position of the intermediate nodes is computed and the cost function is calculated. Then, they apply the steepest descent to minimize the global cost function, which is the sum of the cost function for each intermediate node. The layout is updated according to the magnitude of the flows. Nocaj et al. [123] proposed an approach based on a new drawing style for the generation of flow maps called *confluent spiral* drawings. Confluent spirals consist of smooth drawings of each bundle in which edge directions are represented with smooth appearance. A sequence of spiral segments is used to represent an edge between the origin and the target node, and the vortex of each spiral corresponds to a target node. When an obstacle overlaps a branching point, it is branched out in an earlier or later phase of the parent spiral to miss that obstacle. An exhaustive survey on edge bungling and flow map techniques is provided by Zhou et al. [192].

Rendering. Recent approaches face the problem of estimating the number of links in dense regions. The below proposed solutions are often used in conjunction with edge bundling techniques. In [83, 103, 67], the authors calculate the amount of overlapping items for each pixel of the display. That information is used to map pixels colors with a gradient color scale. In [35, 127], instead of the pixels colors, the opacity of each link is used to map to the density of overlapping items. The use of luminance and saturation as additional visual variables was also proposed [158]. Finally, in [102, 103], the overdraw densities are rendered making dense areas higher than sparse ones. Hence, it preserves links and pixels colors.

Alternative Visualizations. To emphasize the density of links and nodes it is possible to move to another visualization, for example transforming the diagram in a continuous representation. *Graph splatting* convolves nodes with a Gaussian filter [172] into a intensity or height map. Dense nodes regions, which cause clutter, stand out as high-value splats. The same approach can be used to visualize the density of links [104]. As drawback, splatting produces simplified views, making harder the traceability of links. Although these visualizations are able to reveal high-level patterns, they do not permit to distinguish nodes and links. This paragraph includes only alternative visualizations that are created starting from node-link diagrams.

3.2.2 Interactive Techniques

In Information Visualization the role of interaction is crucial, as highlighted in the work of Yi et al. [190]. This section discusses the main interaction techniques used in context of graph visualization to face visual clutter.

Common Techniques. *Dynamic Queries* are one of the most used technique in the context of node-link diagram. They can be used for example to filter out nodes with low connectivity using a slider.

Interactive Lenses. Wong et al. [187] proposed *EdgeLens*; a technique that iteratively curves graph edges away from the point of focus. In this way, the relationship between nodes and edges are disambiguated without information loss. Bearing this in mind, Schmidt et al. [147] used a multi-touch interaction in conjunction with some edge displacement techniques. *Semantic lenses* can be used to explore crowded areas (e.g. overlapping bundles). *MoleView* [90] is a semantic lens that selects a set of data elements located inside the lens' shape and having an attribute value defined by the user. In case of visualization of graphs, the lens moves the control points that compose the links around the lens. *Local Edge Lens* [164] is an interactive technique that draws in a local focus region only edges that connect certain nodes within the lens scope. It reduces the occlusion of element caused by the links. Tominsky et al. [167] provided an exhaustive overview of interactive lenses in Information Visualization domain.

Alternative Interactive Techniques. In [195], a fast hardware-assisted layout technique is used to visually analyze large real world graphs to detect patterns. It uses information of nodes and links density to avoid visual clutter.

Chapter 4

Schematization of Clutter Reduction Techniques in Geographic Node-Link Diagrams using Task-based Criteria

To construct a useful graphic,
we must know what has come before
and what is going to follow.
–Jacques Bertin.

4.1 Overview

Visual clutter is a hot topic in the study of node-link diagrams as it negatively affects usability, aesthetics and data interpretation. As mentioned in Chapter 2, the organization of items, i.e. the way nodes and links are positioned in the screen space, and the number of visible items, are two aspects among many that leads to visual clutter. In previous work, described in Chapter 3, different techniques were proposed to reduce the clutter that depends on the organization of nodes and links and on the number of visible items. However, a schematization of such techniques by task was never considered. Approaching the problem by task would be more efficient since visual clutter, by definition, depends on the task to be performed.

In this chapter, we propose a solution to visual clutter driven by the type of task. In particular, the aim of our work is to get an answer to the following question: *Given a task and a cluttered geographic node-link diagram, which are the appropriated techniques to reduce the visual clutter.*

Our solution crystallizes on a schematization of techniques derived from a sequence of steps. Initially, we have classified tasks into a limited number of task groups. For each tasks group, we have identified and analyzed issues leading to a performance degradation. The final outcome consists on a list of good candidate techniques for each task group. The selected techniques are the results of a survey that takes into account only approaches that act on the position of nodes and links.

Part of the work in this chapter was previously published in [45].

4.2 Introduction

The node-link diagram is a powerful tool for the visualization of relationships between entities. In this work, we focus on an instance of geographic node-link diagram where the nodes are fixed in a geographic space. The latter is the space of the visualization used to provide a geographical reference to the visual features, mapping their position expressed in latitude and longitude to the coordinate axes of the drawing area.

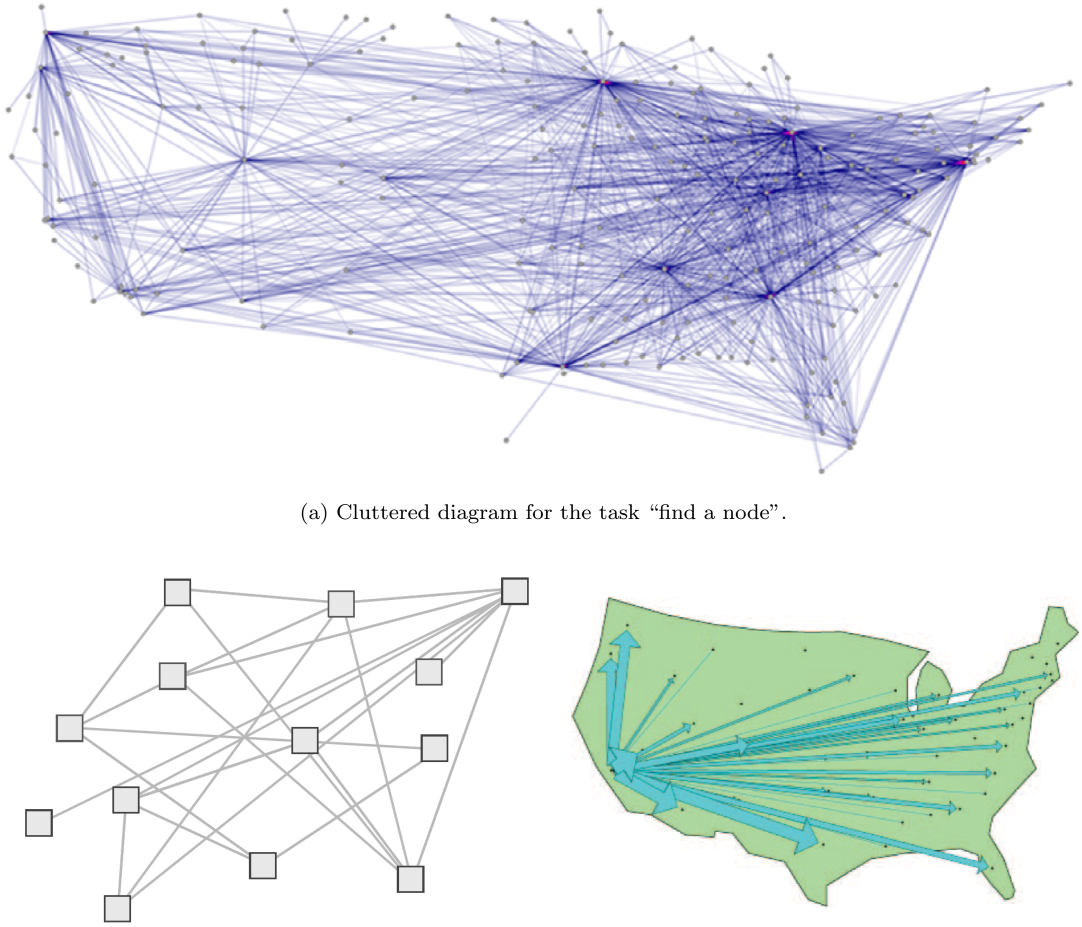
Node-link diagram often suffers from visual clutter [108, 155, 59] affecting usability, aesthetics and data interpretation. In particular, this undesired state is difficult to handle in the geographical node-link diagrams because the relocation of the nodes may cause the loss of geographical information.

In previous studies visual clutter is defined as: “the state in which excess items, or their representation or organization, lead to performance a degradation of performance at some task” [142]. Hence, visual clutter depends on the task, as shown in Figure 4.1. In exploratory and analytical scenarios, if the task is “find a node”, only the diagram in Figure 4.1(a) should be considered cluttered [83, 62, 89] because many nodes are occluded causing loss of important information. If the task is “given a node, find the connected nodes”, also the diagram in Figure 4.1(b) should be considered cluttered [187, 186, 147] because the overlap of nodes over links cause ambiguities. In descriptive scenario, if the task is considered of high-level, such as “understand the story described by the data”, then the diagram in Figure 4.1(c) should also be considered cluttered by the viewer [129, 174, 41] as overlapping links lead to a degradation of aesthetic.

In this work, we focus on visualizations characterized to excess items, and to inappropriate organization of items, i.e. when the position of nodes and links in the display lead to a performance degradation, for which we propose a solution that is driven by task. As opposite, we do not examine cluttered visualizations that depend on the graphical representation of items, for example when the color of the nodes is the same of the background map, or when size of nodes is too small or too large taking into account the display size and the number of items. However, solutions to such problem can be obtained by applying the general guidelines for graphic design [170].

Our work answers to the following research question: *Given a task and a cluttered geographic node-link diagram, which are the appropriated techniques to reduce the visual clutter.*

Our solution crystallizes on a schematization of techniques that is the result of a



(a) Cluttered diagram for the task “find a node”.

(b) Cluttered diagram for the task “given a node, find the connected nodes”.

(c) Cluttered diagram for the task “understand the story described by the data”.

Figure 4.1: Examples of different visual clutter in geographic node-link diagrams.

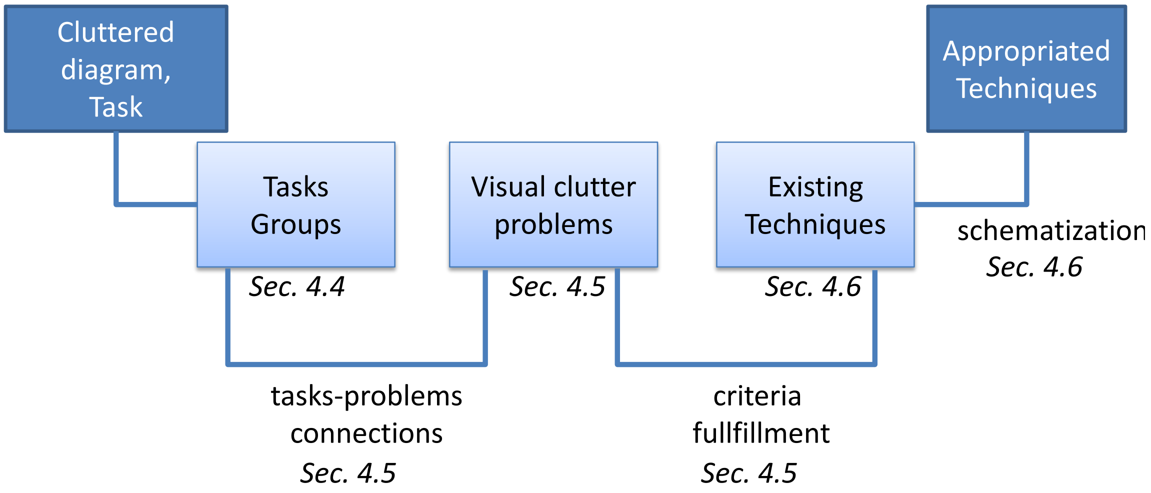


Figure 4.2: The steps that compose our schematization.

sequence of steps, as shown in Figure 4.2. We start diving the tasks into task groups. For exploratory and analytical scenarios, we use the task taxonomy for graph visualization proposed by Lee et al. [105]. For descriptive scenarios, we define a tasks group related to the aesthetic of the visualization. Then, we identify and analyze the main problems that lead to a degradation of tasks performance, i.e. *uninterpretable representation*, *occlusion*, *ambiguity*, and *unaesthetic representation*. Moreover, for each group we associate the problems that characterize it. Finally, we provide a list of good candidate techniques for each task group according to the problems they are able to solve.

The selected techniques are the results of a survey that takes into account only approaches that act on the position of nodes and links. We restrict the analysis to this subset mainly because the position is the most used variable to reduce clutter in geographical node-link diagrams. We do not consider force-directed techniques on nodes because they do not preserve spatial information.

This chapter is structured as follows. First we summarize, in Section 4.3, previous works on visual clutter in geographic node-link diagrams. Then, we define in Section 4.4 our classification of tasks by groups. In Section 4.5, we identify the main visual clutter problems. In Section 4.6, we itemize the main clutter reduction techniques according to the clutter problems they are able to solve. Lastly, an overview of the results obtained and of future work is presented in Section 4.7.

4.3 Existing Surveys

In literature, different surveys on clutter reduction techniques are presented. However, they do not consider the different problems related to visual clutter (e.g. occlusion and ambiguity), nor they take into account the task.

In [155, 108], visual analytics techniques are presented and a section is dedicated to the clutter reduction methods in large graph layouts. However, no hints are made with respect to the aforementioned aspects. The visual clutter and the related solutions are also mentioned in different categorizations of node-link diagrams [76, 46], but not as main aspects.

Zhou et al. [192] provided a survey on edge bundling techniques. They described the algorithms according to the way they generate the final graph layout, i.e. they distinguished *cost-based*, *geometry-based* and *image-based* techniques. As opposite, we classify clutter reduction techniques including edge bundling solutions, conforming to the problems they are designed to solve, by looking at the final layout they generate. In the survey [167], interactive lenses are categorized by data types and tasks. Although many clutter reduction techniques are included, they are classified based on the proprieties of the solutions, rather than on the problems to solve.

A previous survey focused on cluttered visualizations caused by huge amount of data. The authors classified clutter reduction methods defining eight criteria based on their experience and on the study of the related literature, i.e. "avoids overlap, keeps spatial information, can be localized, is scalable, is adjustable, can show point/line attribute, can discriminate points/lines, and can see overlap density" [59]. The main difference between our work and theirs is that their criteria focus on the techniques. As opposite, we provide criteria derived from an analysis of cluttered layouts and tasks. Moreover, our work focuses also on inappropriate organization of nodes and links in node-link diagrams and not on clutter solely caused by huge data.

4.4 Tasks Groups for Geographic Node-Link Diagrams

We distinguished different tasks groups in the context of geo-referenced networks, to better understand visual clutter. At high level of abstraction, we apply the general division for information visualization tasks [98] described in Chapter 1.

Exploratory and Analytical tasks: Visual exploration allows the possibility to get new insight. In case of analytical scenario, the user knows always the task, being it implicit or explicit. We used the tasks groups identified in the taxonomy for graph visualization proposed by Lee et al. [105] as follows:

- *Graph low-level tasks*: The tasks are related to the topology of the graph, i.e. "given a node, find adjacent nodes" and "given a link, find incident nodes".
- *General low-level tasks*: The user examines each item of the visualization to make new discoveries. The tasks are: "find item", "retrieve value", "filter item", etc. The element "item" can be a node, a link, an annotation on the map, a label associated to a node, etc.
- *Overview tasks*: Some tasks require only an overview of the graph such as "find clusters of nodes", "find patterns and outliers".

Such tasks groups allow the composition of high-level tasks such as "follow a give path", "count the number of clusters", etc.

Descriptive scenario: The designer knows the phenomenon observed in the data, but he needs an effective visual representation to transfer knowledge to the viewers. From the viewer perspective, the main goal is to make sense (i.e. associative thinking) of a story visually described by the data. In literature, common examples of node-link diagram used for such tasks are flow maps. In this scenario, we define the task group *associative visualization tasks*.

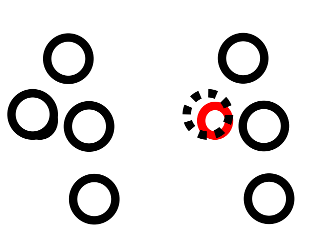
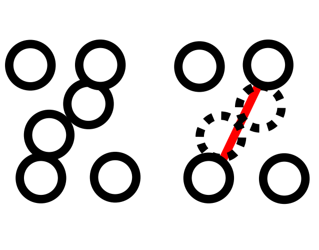
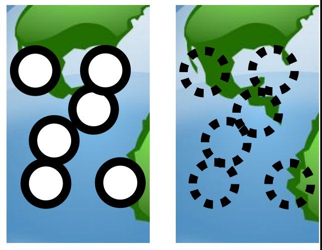
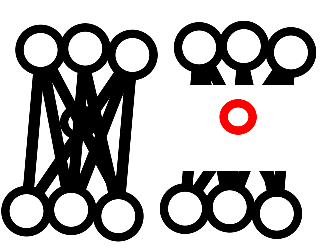
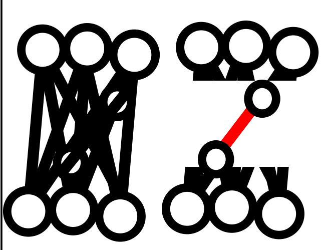
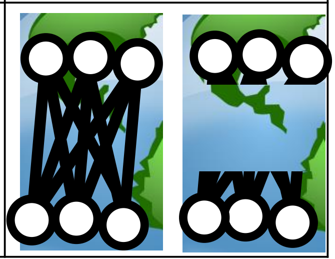
4.5 Visual Clutter Problems

Focusing on the organization of items, clutter is mainly the result of the *overlapping* (also known as overplotting or overdrawing) of those items, i.e. items rendered on top (or near) of each other. In this work, the considered items are nodes, links (straight lines or curves), and map portions of the geographical surface. We define the symbology *(item1, item2)* to indicate that *"item1 is rendered on top of item2"*. We do not consider the overlapping by itself as a problem, but as the cause of the following issues: *occlusion*, *ambiguity*, and *unaesthetic representation*. Analogously, we define the *uninterpretable representation* as a possible consequence to excess items.

4.5.1 Problem of Uninterpretability

The readability of node-link diagrams deteriorates when the size of the graph excesses in number. The latter increases the possibility to have both badly organized nodes and regions with dense links [70]. The problem of uninterpretable representation occurs when items become impossible to identify, making the visualization useless as an hairball-like visualization. This problem involves the entire graph representation and not the single overlapping of two items. Figure 4.1(a) shows an example of node-link diagram that does not provide a useful representation of the network.

Table 4.1: Different cases of occlusion. For each case, we apply an approach that acts on items opacity to solve the problem.

		Item on back		
		Node	Link	Geo-Surface
Item on top	Node			
	Link			

The following criteria are used to valuate the candidate solutions for the problem of uninterpretability.

- ✓: if the technique improves the readability of the visualization reducing the number of visible items.

- -: if the technique does not provide any evidence that it improves the readability.

4.5.2 Problem of Occlusion

The occlusion problem is a well-studied topic in Information Visualization [61, 59, 58]. Occlusion of items results in loss of information. Table 4.1 shows information loss of geographic node-link diagram taking into account all possible combinations between items. We do not consider the occlusion caused to the background map, e.g. when nodes are located over a virtual globe, because it is not the aim of this work. This problem is accentuated with thick items, or with a large number of items. Figure 4.1(a) presents a situation where many links occlude nodes and other links.

Here some items are hidden behind other items:

- *(node,node);(node,link);(node,geo-surface)*: Nodes located is the same spatial location or near with each other may cause their occlusion. In the same way, nodes can occlude also links or the background map. Such kind of occlusion is not easy to solve because, in the geographic node-link diagrams in analysis, the position of nodes encodes geographical information.

- *(link,node);(link,link);(link,geo-surface)*: Links obscure nodes, links, and information in the background such as map labels and map features.

The following criteria are used to valuate the candidate solutions for the occlusion problem.

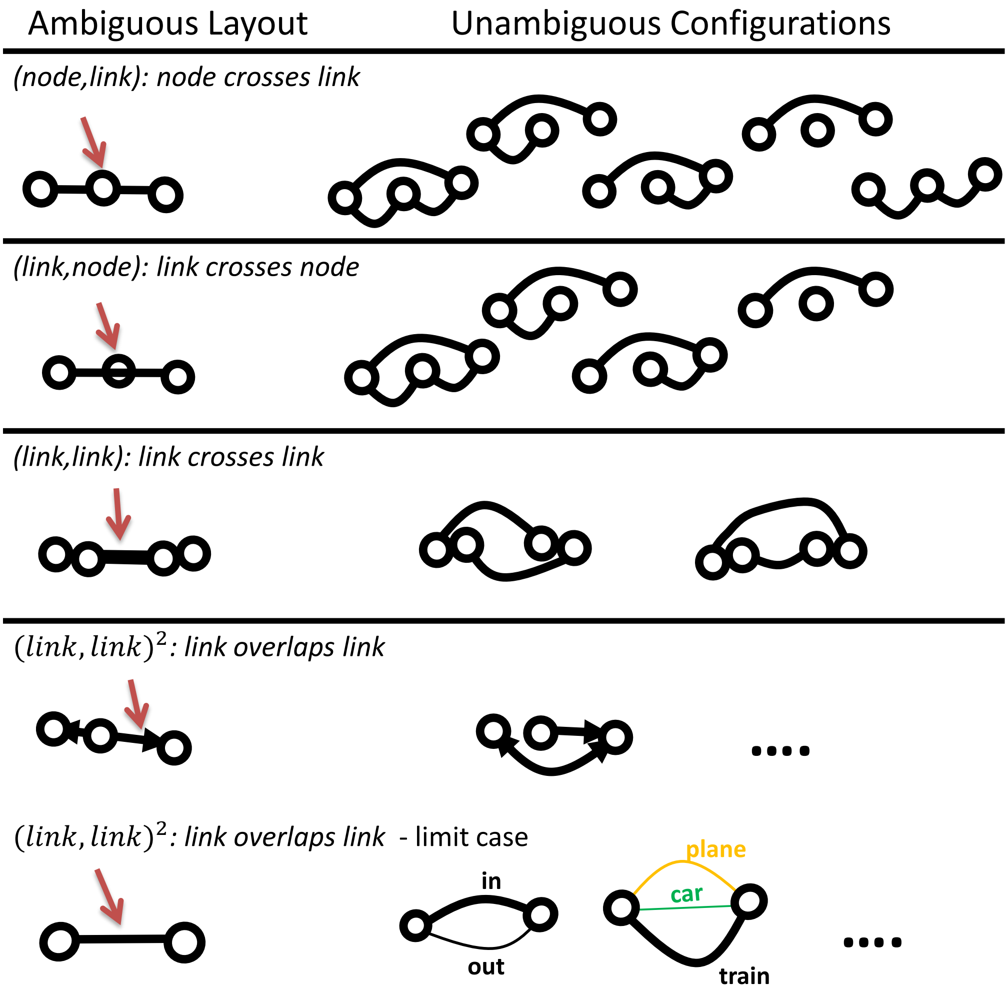
- ✓: If the technique acts on the entire visualization, it reveals all the hidden items. For local regions, the technique reveals the involved hidden items.
- ✓*: if the technique may reveals hidden items, however, this condition is not always guaranteed.
- -: if the technique does not provide any evidence that it reveals hidden items.
- ✗: if the technique increases the number of hidden items. In such situation, the goal of the technique is another.

4.5.3 Problem of Ambiguity

The ambiguity in node-link diagrams is another topic of interest [54]. It can impede the comprehension of network structure [178]. Figure 4.1(b) has different ambiguities in the graph topology. Table 4.2 shows the ambiguity problem of node-link diagram taking into account different cases:

- *(node,link);(link,node)*: It can be difficult to assess, if a link crosses a node, or vice-versa, whether this link is incident to this node or it is merely crossing it. This makes it hard to identify the actual endpoints of links. If nodes are rendered on bottom of links, links may obscure the content of the nodes. In the opposite case, the unambiguous layout candidates are more in numbers. In literature, such cases are respectively named *Edge Bridge* and *Edge Tunnel* [54].
- *(link,link)*: Link crossings are an important factor in readability [134]. We consider crossing links also when they are very close to each other and are almost parallel. When eyes try to follow a link to its destination, small crossing angles between this link and other links create multiple paths along the direction of the eye movement, either slowing down the eye movement, or taking eyes to the wrong path [87]. However, if two links share one node, this ambiguity does not occur.
- *(link,link)*^{2}: In this case, links overlap each other. Hence, it can be considered an occlusion criteria. This happen if they have one or both incident nodes in common.

Table 4.2: Ambiguity cases and their possible unambiguous configurations.



Ambiguous Layout	Unambiguous Configurations
(node,link): node crosses link	
(link,node): link crosses node	
(link,link): link crosses link	
$(link, link)^2$*: link overlaps link*	
$(link, link)^2$*: link overlaps link* - limit case	

....
in
out
plane
car
train
....

For example, if directed links are represented as arrows, they completely overlap if they share the same nodes. This case includes the *Low Angular Resolution* [54] i.e. the problem related to the minimum or average angle formed by all the links incident to an individual node.

The following criteria are used to valuate the candidate solutions for the ambiguity problem.

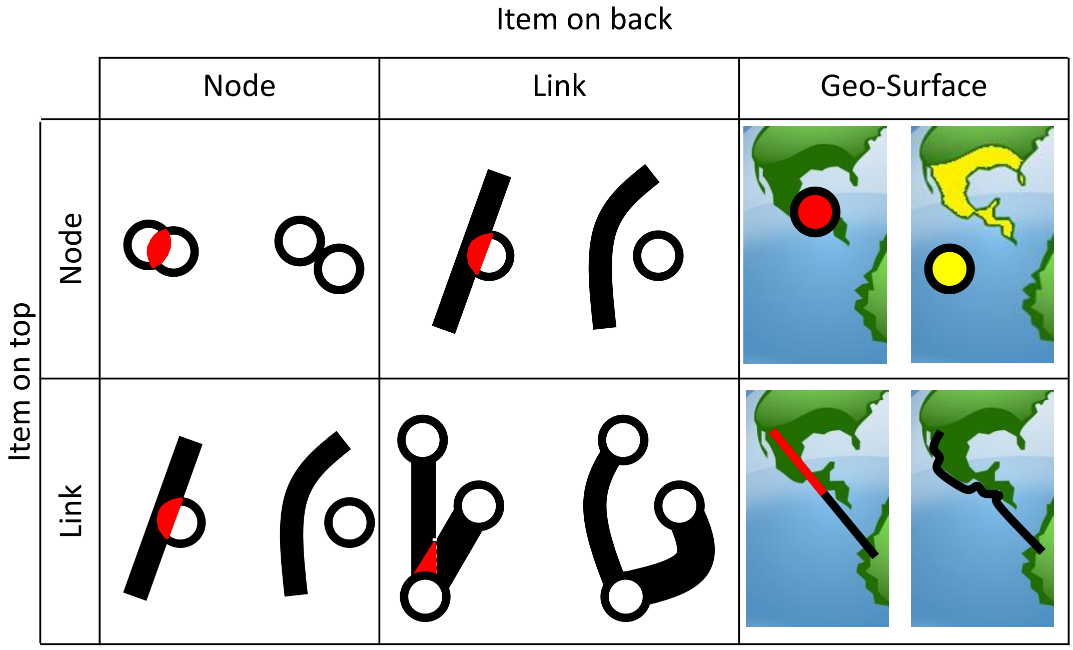
- ✓: If the technique acts on the entire visualization, it disambiguates all the ambiguities for the specific case. For local regions, the technique disambiguates only the involved region. For *(link,link)*, it is not always possible to avoid crossing links. Thus, in that situation the technique has to increase the space among links, or improve their crossing angles.
- ✓*: if the technique may disambiguates some instances, however, this condition is not always guaranteed.
- -: if the technique does not provide any evidence that it disambiguates the visualization.
- ✗: if the technique increases the ambiguity in the visualization. In such situation, the goal of the technique is another.

4.5.4 Problem of Unaesthetic

As shown in Table 4.3, partial overlapping causes a decrement in visual quality of the layout. This is supported by the fact that, in literature, a situation with no overlapping of items, is considered not only a readability criterion but also an aesthetic criterion [129, 174, 41]. Compared with the occlusion problem, the problem of unaesthetic representation does not cause necessarily loss of information. However, there is a lack in literature of studies related to which visual properties make a visualization visually appealing from the viewer perspective [30]. Figure 4.1(c) has overlapping of links that affects negatively the aesthetic of the visualization. The following cases are relevant to describe the effect of clutter:

- *(link,node);(node,link)*: Links can cut directly across a node (or vice-versa) interfering with the visual quality of the layout.
- *(link,link);(node,node)*: It is possible to have two links or two nodes that are close to or overlap each other.
- *(link,geo-surface);(node,geo-surface)*: When links or nodes overlap a region, they may decrease the layout aesthetic.

Table 4.3: Partial overlapping of items (colored in red) causes a decrement of layout aesthetic. For each case, an approach acting on items position is applied to solve the problem.



The following criteria are used to valuate the candidate solutions for the problem of unaesthetic.

- ✓: if the technique improves the aesthetic of the visualization avoiding the overlap of items.
- ✓*: if the technique improves the aesthetic of the visualization reducing the number of overlapping items.
- **-**: if the technique does not provide any evidence that it improves the aesthetic acting on the overlapping of items.

In this section, we also identify the problems that characterize each tasks group. Figure 4.3 summarizes the exploratory and analytical scenarios.

In "Overview tasks", visual clutter is interconnected to the interpretation of the visual representation. In these tasks, it is not essential to solve the problem of occlusion, ambiguity and unaesthetic.

In "General low-level tasks", visual clutter involves also the occlusion of items. It becomes difficult to understand information encoded in visual variables of occluded elements.

In "Graph low-level tasks" ambiguity is an important aspect to take care of. In this task group, we are not only searching for hidden elements, but also we are trying to have

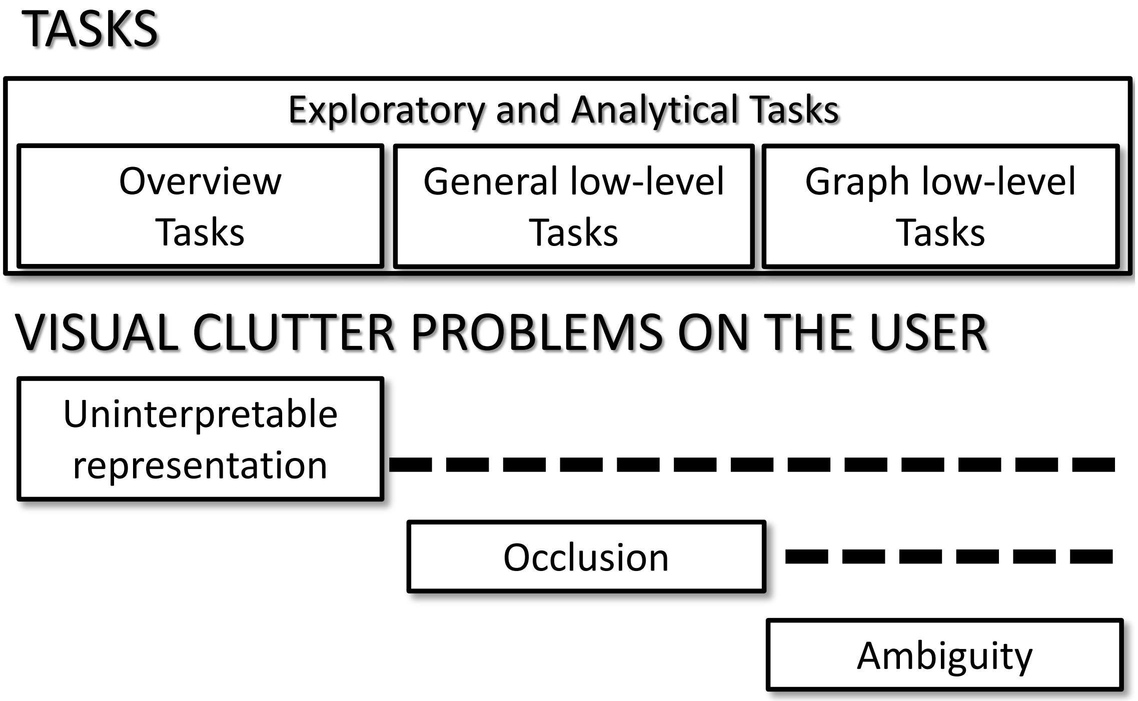

TASKS
Exploratory and Analytical Tasks
Overview Tasks
General low-level Tasks
Graph low-level Tasks
VISUAL CLUTTER PROBLEMS ON THE USER
Uninterpretable representation
Occlusion
Ambiguity

Figure 4.3: In exploratory and analytical scenarios, each task group is affected by different problems.

a clearer (unambiguous) layout.

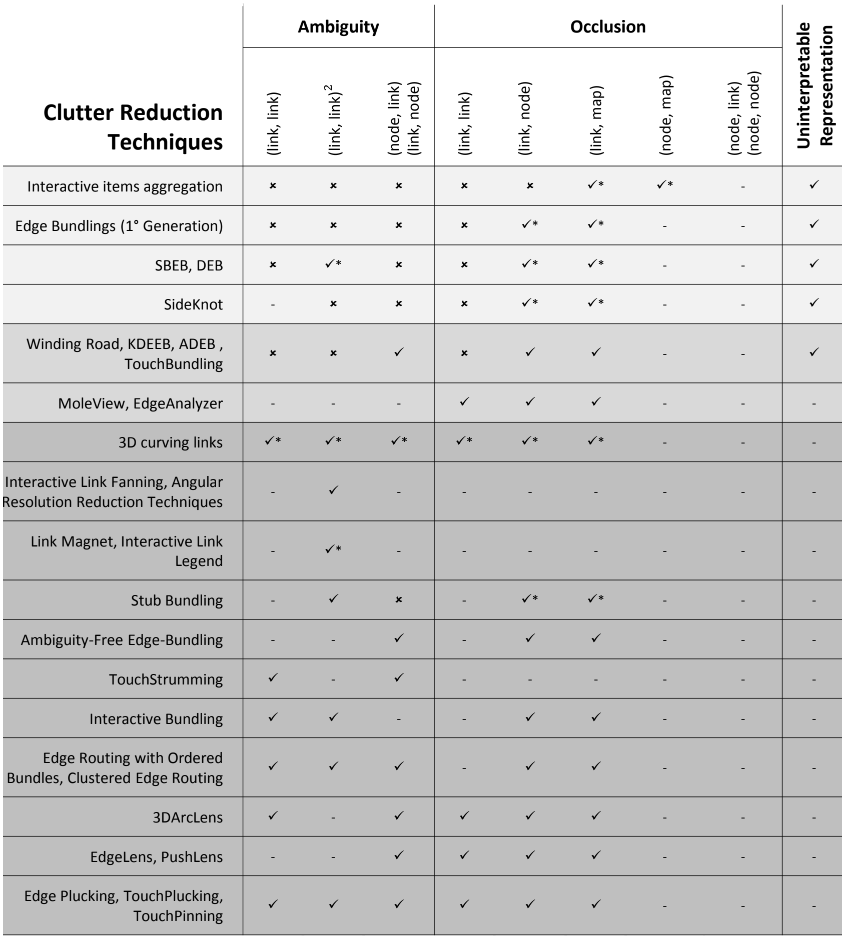
The aesthetic is not a priority in exploratory and analytical tasks. As opposite, in "Associative visualization tasks", among others, there is the problem of unaesthetic representation. Due to the layout, some partial overlap between elements may occurs, causing a decrement in visual quality of the layout.

4.6 Applied Schematization

In this section, we described only approaches that act on the position of nodes and links[1]. We restricted the analysis to this subset mainly because the position is the most used variable to reduce clutter in geographical node-link diagrams. We did not consider force-directed techniques on nodes because they did not preserve spatial information. We investigate works published in *Graph Drawing*, *IEEE Vis*, *IEEE Pacific Vis*, *IEEE Computer Graphics Forum*, *IEEE Visualization and Computer Graphics*, and *VISIGRAPP*. Moreover, we included some work that appeared in the bibliography of the paper in analysis. We surveyed 42 techniques in total: 29 global techniques, including 18 edge bundling

[1] Most of the techniques described in this section were also introduced in Chapter 3.

Table 4.4: Techniques to reduce ambiguity/occlusion/uninterpretable representation in exploratory and analytical scenarios. From top to bottom raws, the color identifies the task group: Overview Tasks, General Low-level Tasks, and Graph Low-level Tasks.



	Ambiguity			Occlusion					
Clutter Reduction Techniques	(link, link)	(link, link)^2	(node, link) (link, node)	(link, link)	(link, node)	(link, map)	(node, map)	(node, link) (node, node)	**Uninterpretable Representation**
Interactive items aggregation	×	×	×	×	×	✓*	✓*	-	✓
Edge Bundlings (1° Generation)	×	×	×	×	✓*	✓*	-	-	✓
SBEB, DEB	×	✓*	×	×	✓*	✓*	-	-	✓
SideKnot	-	×	×	×	✓*	✓*	-	-	✓
Winding Road, KDEEB, ADEB , TouchBundling	×	×	✓	×	✓	✓	-	-	✓
MoleView, EdgeAnalyzer	-	-	-	✓	✓	✓	-	-	-
3D curving links	✓*	✓*	✓*	✓*	✓*	✓*	-	-	-
Interactive Link Fanning, Angular Resolution Reduction Techniques	-	✓	-	-	-	-	-	-	-
Link Magnet, Interactive Link Legend	-	✓*	-	-	-	-	-	-	-
Stub Bundling	-	✓	×	-	✓*	✓*	-	-	-
Ambiguity-Free Edge-Bundling	-	-	✓	-	✓	✓	-	-	-
TouchStrumming	✓	-	✓	-	-	-	-	-	-
Interactive Bundling	✓	✓	-	-	✓	✓	-	-	-
Edge Routing with Ordered Bundles, Clustered Edge Routing	✓	✓	✓	-	✓	✓	-	-	-
3DArcLens	✓	-	✓	✓	✓	✓	-	-	-
EdgeLens, PushLens	-	-	✓	✓	✓	✓	-	-	-
Edge Plucking, TouchPlucking, TouchPinning	✓	✓	✓	✓	✓	✓	-	-	-

techniques and 4 flow maps algorithms, and 13 techniques focusing on a restricted area of the screen. Each technique was compared with respect to our criteria.

4.6.1 Exploratory and Analytical Scenarios

As shown in Table 4.4, for each task group, we proposed a list of candidate solutions for cluttered diagrams as follows: techniques for overview tasks are the ones solely able to solve the uninterpretable representation; for general low-level tasks, the chosen techniques solve or reduce mainly the occlusion issue; techniques for graph low-level tasks solved ambiguity and in some cases also occlusion issue.

Overview Tasks. In [195], a fast hardware-assisted layout technique was used to visually analyze large real world graphs to detect patterns. It used information of nodes and links density to reduce dynamically the number of visible items, allowing map portions to be revealed and reducing visual clutter. In particular, it aggregated nodes and links according to their spatial proximity. As drawback, it increased the overlapping of nodes and links, making the ambiguity and occlusion issues more frequent.

Edge bundling is one of the main techniques that makes a layout more easy to interpret. In edge bundling, links of a graph are bundled together according to defined conditions. In the first generation of such algorithms [135, 193, 158, 82, 35, 83, 67, 102], without considering the combination with other techniques, the main benefit was merely the reduction of the number of visible items. Moreover, they reduced the space dedicated to the links, increasing the possibility to make nodes and map portions visible. As opposite, the aggregation of links leaded to a more ambiguous visualization.

Edge Bundling techniques were also designed to manage multivariate networks. *Divided Edge Bundling (DEB)* [150] was able to separate opposite-direction bundles, emphasizing the graph structure. *Skeleton-Based Edge Bundling (SBEB)* [62] bundled similar links according to directions and also further variables associated to links. The ambiguity of links sharing the same node was solved only for links with a different attribute, e.g. links with opposite direction.

SideKnot [127] focused on ambiguity caused to overlapping links, aggregating only links that share the same node. Although it did not reduce the ambiguity, with respect to the aforementioned bundling techniques, it did not even increase that problem.

General low-level tasks. *Winding Roads* [103], *Graph Bundling by Kernel Density Estimation (KDEEB)* [89], and *Attribute-Driven Edge Bundling (ADEB)* [128], compared to previous Edge Bundling approaches, were able to avoid overlapping of links with all nodes and with map portions decided at priori. In this way, they removed the ambiguity and occlusion issues related to the overlapping of links with nodes. Similarly, *Touch-Bundling* [147] is a multi touch technique that aggregates edges but interactively.

MoleView [90] is a semantic lens that revealed a set of data elements located within the lens' radius and having an attribute value defined by the user. In case of node-link diagrams, the lens moved around the lens the control points of links. *EdgeAnalyzer* [126] is able to select specific sub-groups of links on dense areas. Both techniques were able to reveal nodes, links and map portions occluded by the links.

Graph low-level tasks. Links can be drawn as curves in a 3D space to reduce ambiguity issues, snapping nodes over a geographical surface to preserve the spatial context [33, 121]. Curved links allowed more screen space compared to their straight counterparts and potentially reduced visual clutter [189]. However, 3D curving links required the user to navigate over the visual space. Hence, the effectiveness of such technique depended on the camera view position. The ambiguity related to links connected to the same nodes was solved in 2D [21, 20] and in 3D [19] visualizations increasing the angle between incidents links of every node. Analogously, *Interactive Link Fanning* [136] created space between links incident to a selected node, to show labels or arrowheads for individual links.

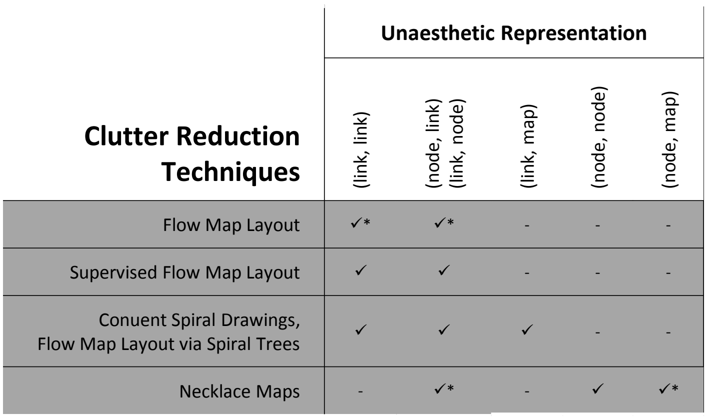
With *Link Magnet* [136], when a visual object like a circle representing a data attribute was dragged, links were attracted according to the attribute value of the link. Meanwhile, with *Interactive Link Legends* [136], link curvature encoded semantic information such as different types of links in multivariate graphs. The above mentioned approaches partially solved the ambiguity related to links sharing the same node, because they required additional information of the links.

Edge bundling techniques can be designed to reduce ambiguities. *Stub Bundling* [123] bundled links that shared nodes, and put them in parallel to facilitate the display of additional data attributes by varying width or color. Hence, it solved the *(link,link)*[2] ambiguity problem. *Ambiguity-Free Edge-Bundling (AFEB)* [110], respect to the previous technique, was also able to reroutes links that passed over nodes reducing the *(node,link)* ambiguity. *TouchStrumming* [147] is able to vibrate a link of interest. In this way, the ambiguities related to crossing links, and overlapping of nodes and links is solved. *Interactive Bundling* [136] was an interactive technique that generated crossing-minimal bundles that were routed to distinguish them. Moreover, changing the focus area, the user can also choose which map portion reveal. Finally, *Edge Routing with Ordered Bundles* [133] was able to fulfill most of our criteria. Here, links were placed in parallel channels to avoid overlaps. Additionally, *Clustered Edge Routing* [17] allowed the user to define the links to cluster. The last two approaches could theoretically be used to create flow maps, however, the result has to be evaluated in terms of aesthetic.

Wong et al. [187] proposed *EdgeLens*; a technique that iteratively curved graph links away from the point of focus. This consented to disambiguate the relationship between nodes and links without losing information. An analogous multi-touch technique was *PushLens* [147]. *3DArcLens* [42] extended the functionalities of EdgeLens, distinguishing the distorted links around the lens. As drawback, the distorted links may crossed with the surrounding lines causing further ambiguity[2]. With *Edge Plucking*, the user can drag groups of links away to clarify cluttered zones and specify links or nodes to be left unmoved [186]. However, Edge Plucking required a certain amount of manual effort. Bearing this in mind, Schmidt et al. [147] designed (but not implemented) multi-touch interaction techniques based on link displacement. With such techniques, the user had freedom to modify the layout of every links to reveal items or solve ambiguities. In particular, *TouchPlucking* is a multi touch variant of Edge Plucking; *TouchPinning* is able to alter the trajectory of selected links as long as the user desire.

[2] For more information, see Chapter 5.

Table 4.5: Techniques to reduce unaesthetic representation in geographic node-link diagram for Associative Visualization Tasks.



	Unaesthetic Representation				
Clutter Reduction Techniques	(link, link)	(node, link) (link, node)	(link, map)	(node, node)	(node, map)
Flow Map Layout	✓*	✓*	-	-	-
Supervised Flow Map Layout	✓	✓	-	-	-
Conuent Spiral Drawings, Flow Map Layout via Spiral Trees	✓	✓	✓	-	-
Necklace Maps	-	✓*	-	✓	✓*

4.6.2 Descriptive Scenario

In Associative visualization tasks, the techniques focus on the problem of unaesthetic representation as shown in Table 4.5. In literature, they are called *presentation-oriented techniques* [101].

Associative visualization tasks. In *Necklace Map* [153], nodes were rearranged in circular layouts to remove the overlapping between them. However, the spatial context of nodes was preserved for two reasons: the nodes were moved not too far from their original position and the color of nodes was used to associate them with their original location on the background map.

Phan et al. [129] developed a method to generate a flow map layout in a recursive and simple manner. The overlapping between links and nodes is only partially solved because undesired crossings may occur. However, in a post processing phase, user has the possibility to modify flow lines by moving their control points. Debiasi et al. [41] presented a method to generate flow map layouts using a force directed approach to remove the *(link-node),(node-node)* overlapping[3].

A second generation of flow map algorithms was developed to satisfy all the criteria related to the partial overlapping of links. Verbeek et al. [174] introduced a method using spiral trees, i.e. links are logarithmic spirals, implemented as cubic Hermite splines. A different approach called *Confluent Spirals* [123] consisted of smooth drawings in which link direction are represented by increasing curvature.

[3] For more information, see Chapter 7.

4.7 Conclusions

The scope of this work is not the creation of a rank of techniques, but a schematization that helps the reader to decide, given a task, which are the candidate solutions that help to reduce the clutter in a geographical node-link diagram.

This work provides a list of criteria to classify the effects of the visual clutter on geographic node-link diagrams on different scenarios. Moreover, it offers guidelines for the design of novel techniques, helping the researchers to focus on a well-defined list of criteria to fulfill.

Looking at Table 4.4, it is possible to notice the high heterogeneity of edge bundling techniques in exploratory and analytical scenarios. One the one hand, they can be used to simplify the visualization in terms of visible items. On the other hand, they can be used to solve both ambiguity and occlusion issues. However, no static edge bundling technique is able to solve the occlusion of a hidden portion of a map not defined at priori. In that case, the use of an interactive approach is recommended. Moreover, although eye tracking studies revealed that small angles can slow down and trigger extra eye movements [88], no edge bundling technique tried to maximize the angle between crossing links. Instead, they put the bundled links in parallel and increase the space between them.

Furthermore, some techniques are able to solve both ambiguity and occlusion issues. Thus, they can be used to facilitate both general low level tasks and graph low level tasks. As opposite, there are techniques that, improving the interpretation of the visualization, cause an increment of the ambiguity and occlusion issues.

Additionally, only approaches that put in parallel links with no restriction on graph topology, are able to solve the ambiguities of crossing links. As opposite, approaches that put in parallel only links that share nodes, are able to solve the ambiguity of overlapping links. In exploratory and analytical scenario, no technique is able to solve the occlusion of items caused by nodes without losing their geographical context or removing them from the visualization.

In descriptive scenario, only one approach (Necklace Map) satisfies the partial overlapping criteria related to overlapping nodes. The reason is the difficulties in rearranging the nodes without losing their geographic information.

Respect to the organization of items, this work focuses on the overlapping of items. However, other aspects may cause visual clutter, such as the abnormal link lengths [178] and the asymmetry of the items [54].

As future work, we plan to extend this schematization into a classification of techniques, based on their intent, e.g. “put in parallel”, “aggregate”, “push away”. Furthermore, we could include techniques that act on other visual variables such as color, final image rendering, etc.

Chapter 5

3DArcLens: Interactive Network Analysis on Geographic Surfaces

If a picture is worth a thousand words,
then an interface is worth a
thousand pictures.
–Ben Shneiderman.

5.1 Overview

Geographic datasets such as international telecommunications traffic, financial flows, trading patterns, and national migration patterns describe the movement of entities between geographical locations. In spatial relations analyses, the exact route of the connections is not important. Hence, one of the most preferred methods for their depiction is a graph representation with data nodes layered over a geographical surface (such as a flat map or a virtual globe). However, as mentioned in Chapter 2, a large number of links and their organization can produce visual clutter causing occluded items and ambiguities in the graph representation.

In this chapter, we present a novel focus+context technique for 2D and 3D virtual environments that interactively distorts and filters links, revealing underneath information about: nodes, links and geographical surface. Moreover, it reduces the ambiguity caused by the overlapping of links with other links and nodes. In our case studies, we observed that such technique is an advantage for tasks that include the exploration of geographical networks. Part of the work in this chapter was previously published in [37, 42][1].

[1] As additional material, a video showing a demo of the presented technique was included in the related published papers. It is available online at `https://www.youtube.com/watch?v=AzORB5sdoxc&list=PLlB_rfTJJP4S6ErhbjToHOJHe89wc5WZg&index=1`

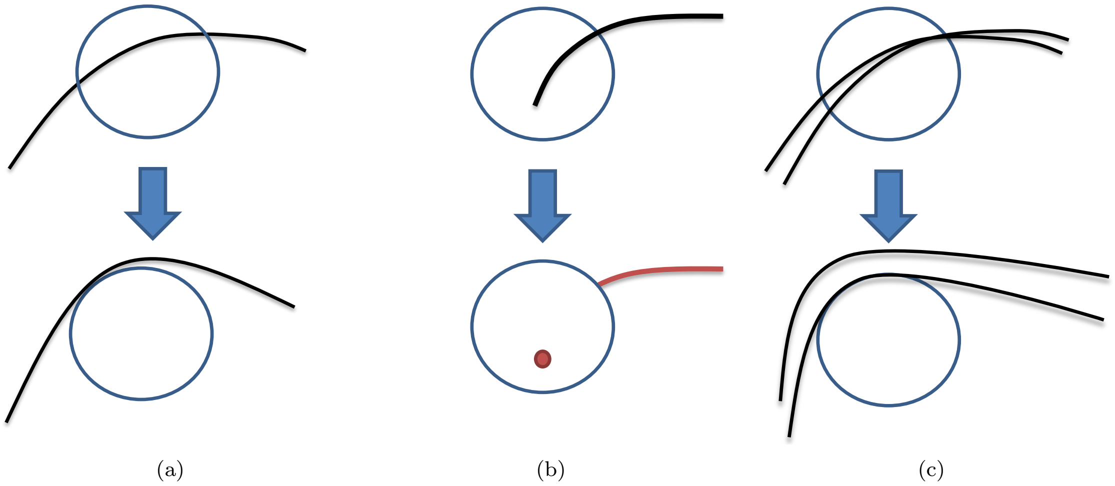


Figure 5.1: Combined strategies that characterize 3DArcLens.

5.2 Introduction

A huge amount of geographical data is available nowadays. Many of these datasets describe the relations of spatial entities, represented in form of graph. Examples of such datasets are international telecommunications traffic, financial flows, trading patterns, and national migration patterns [161]. Representations based on geographical maps can be useful to the study of these datasets, e.g. allow users to reason about geographic patterns of the spatial entities.

As mentioned in Chapter 2, the main visualization used to express both relational and geographical information of the data is the node-link diagram, where nodes are located on top of a flat map or a globe. In both solutions, the depiction of considerable amount of links over an area often causes occlusion, either on the nodes and links or on the layered geographical map. The latter is particular important due to the geographical information encoded into the dataset. For example, if the dataset to be analyzed describes tourist flows then it can be relevant to visualize the layered map that contains the major touristic attractions in the world. Moreover, the overlap of links with nodes and crossing links often cause ambiguity in the graph representation.

Although the user can pan, tilt and zoom the camera, in some situations this in not enough to reveal the hidden information and to solve ambiguities. In addition, the required time to compute these interactions might not be acceptable. Moreover, performing the operations in 3D space can be disorienting for the user [95]. Since the nodes encode geographical information, techniques that rearrange node positions, such as graph layout approaches [94, 55] are not recommended.

In this chapter, we introduce a novel focus+context technique called *3DArcLens* that interactively reduces the occlusion and the ambiguity problems without removing links or moving nodes. We define a lens based technique that acts on the links depending on their position and incident nodes. In particular, if the interactive lens intersects only with the link and not with its incident nodes, then the link is replaced with an aesthetically pleasing curve placed around the lens to be easily traced, see Figure 5.1(a). If the link has an incident node inside the lens, the node is visualized as a sphere with an unique color, meanwhile the segment inside the lens is not drawn and the rest is colored conforming to the node's color, see Figure 5.1(b). Due to the fact that the links that depict relational data do not represent real routes, changing their shape or not drawing a part of them does not affect the information they represent. Combined, these two methods allow the visualization of elements inside the lens (map, nodes and links) to be clutter free. Moreover, they reduce the ambiguity among nodes and links. Additionally, deformed links are placed on different distance from the lens to reduce the ambiguity among crossing links, see Figure 5.1(c). This interactive lens is specifically designed to support the identification of elements of geographical networks as well as the exploration of the layered map with 2D and 3D views preserving their geographic context, i.e. flat map and virtual globe.

This chapter is structured as follows. Section 5.3 introduces the problem of large number of links drawn in a limited geographical surface. Section 5.4 describes our theoretical framework. Some case studies are performed in Section 5.5. In Section 5.6, a comparison with similar techniques is made, and a discussion on the limitation of the proposed technique is done. We conclude with some remarks about the usability of the technique in Section 5.7.

5.3 Arc congestions over a geographic surface

The undesired situation addressed by this work is when a large number of links is drawn in a limited geographical surface. On the one hand, it may cause occlusion of nodes, links and background map. On the other hand, it may cause ambiguity in the final 2D image, e.g. some link may cross over a node making the false impression that the two items are connected.

Formally, there is a 3D (resp. 2D) space U defined by a Cartesian space $(x, y, z) \in \mathbb{R}^3$ (resp. $(x, y) \in \mathbb{R}^2$). A geographic surface S is the surface of a volume, e.g. a virtual globe or a flat plane, within U. Nodes in the set N are points located over S. An arc a is a link representation composed by a sequence of points $p_1, p_2, ..., p_n \in U$ such that the first and the last points represent nodes (i.e. $p_1, p_n \in N$).

We assume that, given a geographic visualization, the user is interested in analyzing at the time only a subset (*focus*) of the entire data (*context*) according to a geographic area. Consequently, the arcs, nodes, and surface of interest are formally defined below. S' is the *area of interest* over S and N' contains the *nodes of interest* such that are inside S'. Furthermore, we define the arcs in focus according to their connected nodes. As shown in Figure 5.2, the *arcs of interest* are the ones with one endpoint in N'. The *arcs of high interest* are the ones with both endpoints in N'. The *undesired arcs* are the arcs that *occlude* S' and are not *arcs of interest* neither *arcs of high interest*. Finally, the *context arcs* are the arcs in background (i.e. the arcs in the context).

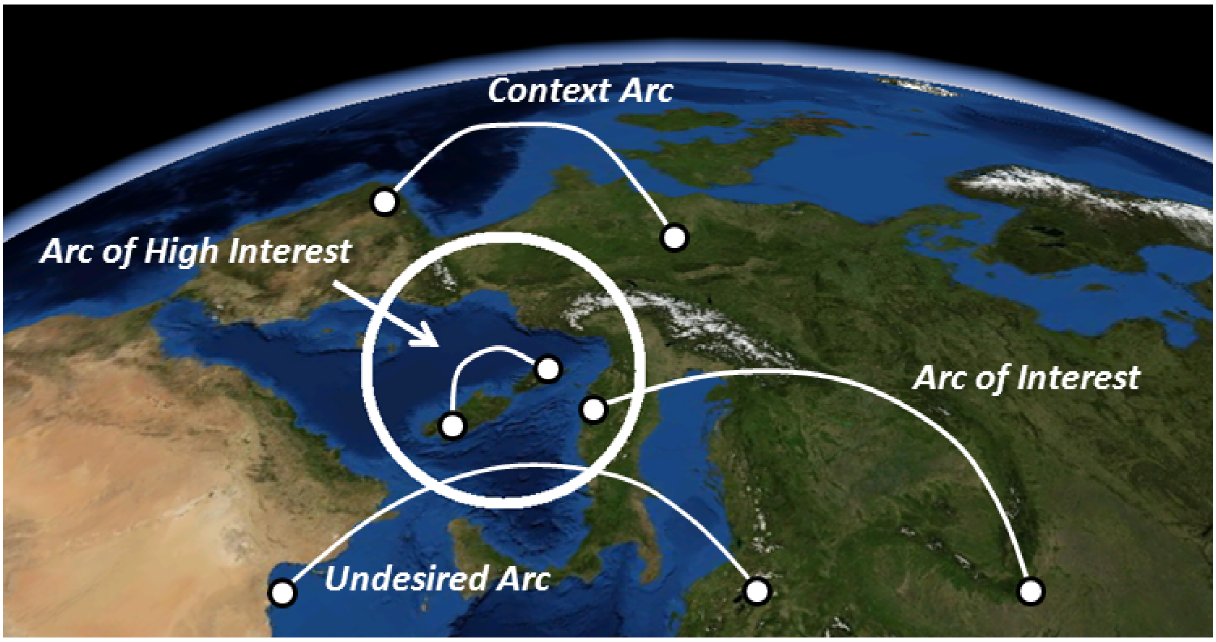
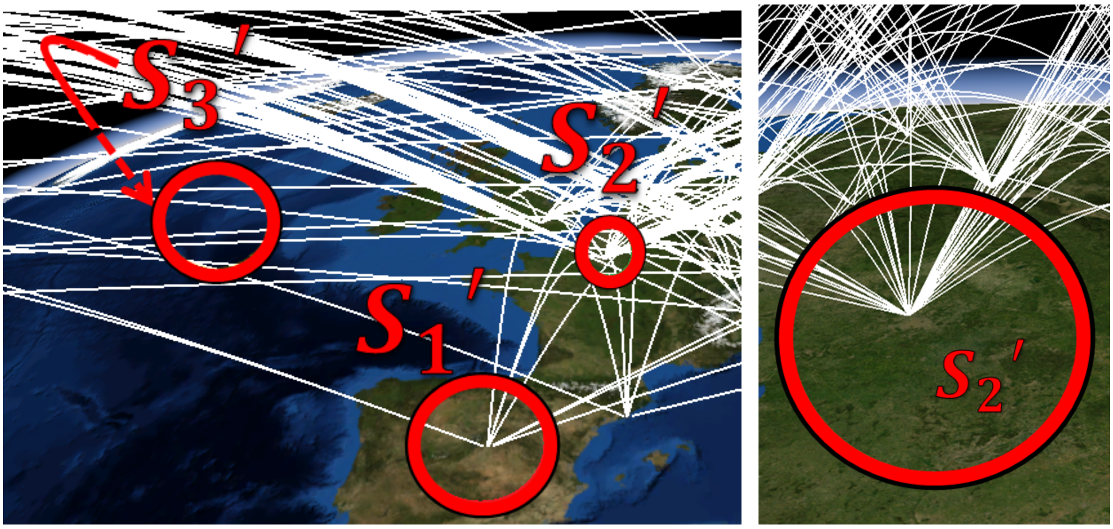


Figure 5.2: The arcs are classified according to their relations with the area of interest.

The area S' is *visible* if it is not occluded by the geographical surface S, and *fully visible* if it is *visible* and there are no *undesired arcs*.



(a)

(b)

Figure 5.3: Arc congestion over a geographic surface. (a) *fully visible* area (S_1'), *visible* area (S_2') that is partially hidden by *undesired arcs* and *not visible* area (S_3') occluded by the globe. (b) After moving the camera, the area S_2' becomes *fully visible*.

Figure 5.3 shows examples of arc congestion over a geographic surface: there are three *area of interests* S_1',S_2', and S_3' identified by a colored circle. In Figure 5.3(a), S_1' is *fully visible*, it is possible to see the nodes inside the area and their connected arcs. S_2' is *visible*, in fact some arcs occlude a portion of its area. Figure 5.3(b) shows the same area S_2' but from a different camera view v, in this case S_2' becomes *fully visible*. S_3' is placed in the hidden hemisphere so it is *not visible*.

This chapter proposes a solution to the following research question: *How can we reduce the arc congestion in a given area, revealing information about the nodes, the arcs and the geographic surface without moving the camera view and with no information loss.*

This question is formalized by the following constraints (C) and requirements (R):

C1. No arc or node is removed from the scene to preserve all the information related to the visible elements of the network.

C2. Nodes are not moved due to the importance of their geographical location.

C3. The surface S must not be deformed due to the importance of its geographical context.

R1. The hidden information on the *visible* area S' will be revealed, in particular about the following visual items, listed by importance in descending order: (a) The location

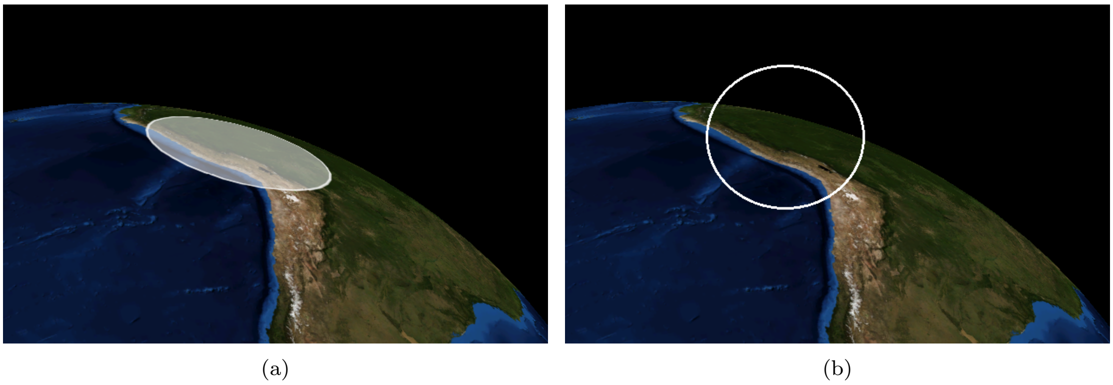


Figure 5.4: (a) The lens is represented as a circle over the globe, hence the lens' shape is not distortion-free. (b) Our adopted approach: the lens is a circle in 2D screen space.

of the nodes in N'. (b) The *arcs of high interest.* (c) The *arcs of interest.* (d) The map layered on S'. (e) The *undesired arcs*, although they cause the clutter, they may encode useful information to the viewer. (f) The *context arcs.*

R2. A *visible* selected S' will be *fully visible* without the need to change the viewpoint.

R3. After selecting an *area of interest*, the solution will work also if the camera view is changed. Formally, let S' the *visible* selected area for a set of viewpoints, S' must be always *fully visible.*

5.4 Proposed Approach

As mentioned in Section 5.2, our goal is to facilitate the visual exploration of both geographical network and layered map when arc congestion situations occur. Our proposed technique uses a lens metaphor that is defined by a given point of focus and a radius.

A possible design choice is to represent the lens as a circle that follows the terrain. Therefore, its shape is not distortion-free. As shown in Figure 5.4(a), if the camera is tilted the area of the lens becomes oval. For this reason we opted to represent the lens as a circle in screen space with the radius r in pixels. Hence, the camera movement does not affect the lens' shape and the zoom interaction does not change the lens radius, see Figure 5.4(b). Up to now, we refer to S' as the *area of interest* that is covered by the lens.

As shown in Figure 5.5, the lens combines different strategies.

Deform arcs. The *undesired arcs* are distorted in a way that they do not pass though the geographical surface covered by the lens (i.e. S'). The distortion is applied not only to the subsegments affected by the lens, but also to the entire curve making arcs easier to trace. In fact the arcs are depicted as splines composed by a sequence of control points.

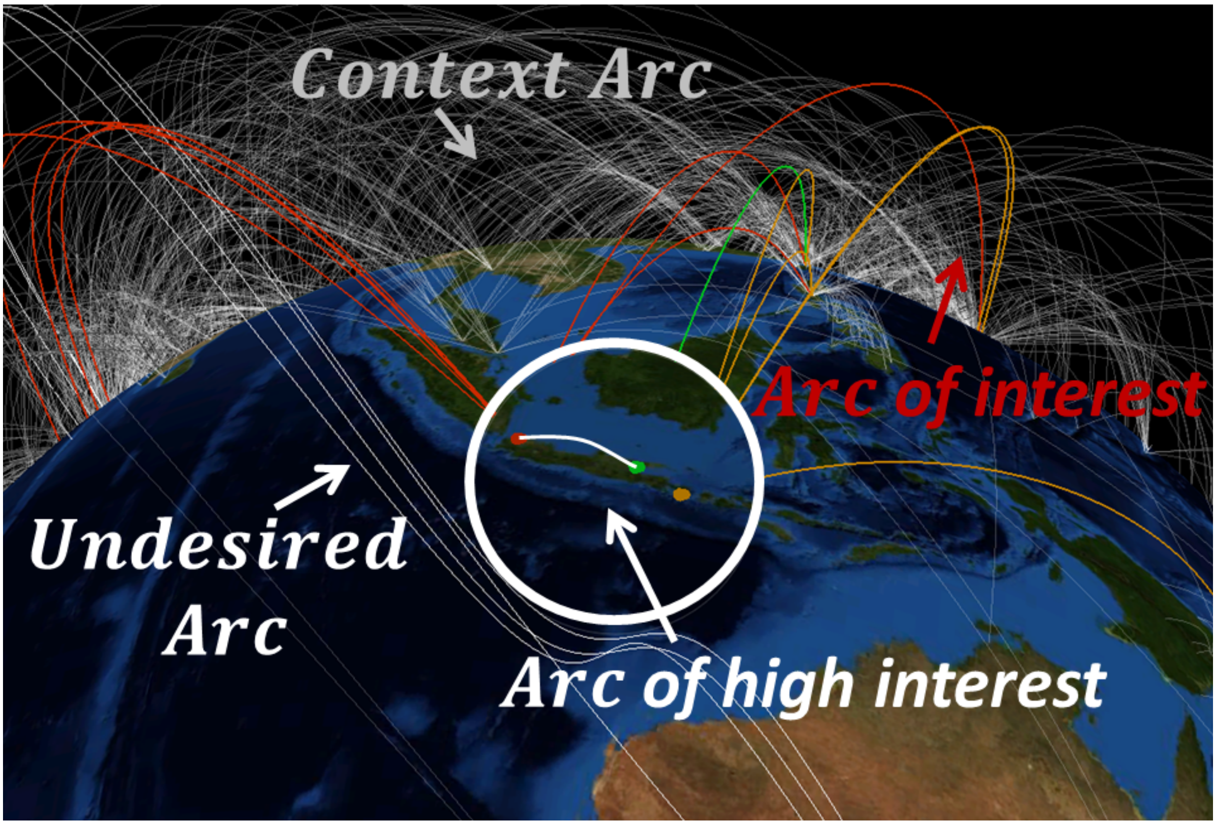
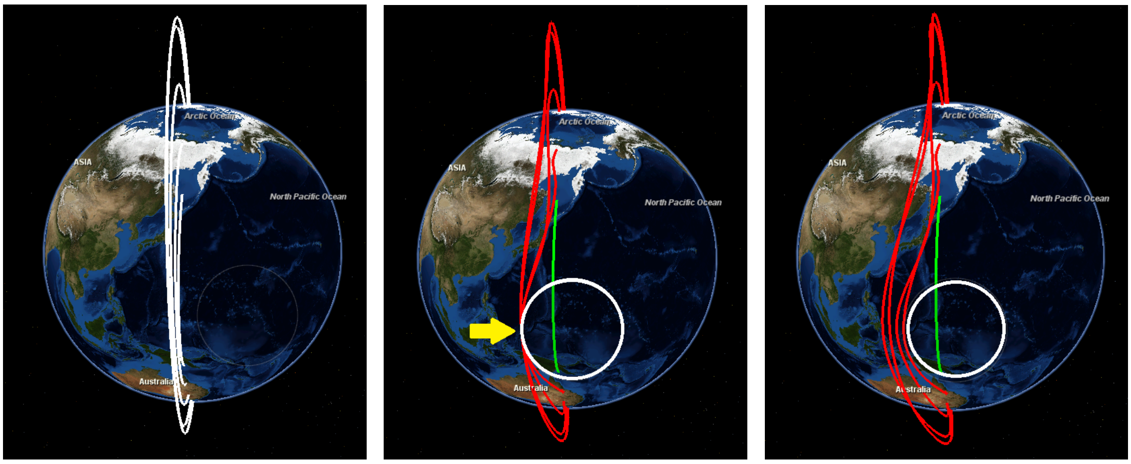


Figure 5.5: Each class of arcs is represented in a different way. *Context arcs* are drawn with transparency, *undesired arcs* are deformed, *arcs of interest* are colored differently and parts of them are filtered out, and *arcs of high interest* are white colored.

Color arcs and nodes. The arc congestion may be also caused by *arcs of interest*, thus their subsegments inside the lens are filtered out from the visualization. This strategy permits to clearly identify the nodes inside the lens and the *arcs of high interest.* What is important to know about each arc is not its shape but which are the incident nodes. Bearing this in mind, each *node of interest* is represented as a colored sphere and each *arc of interest* is colored with the same color of its incident node inside the lens. *Nodes of interest* are therefore easily identified together with their incidents arcs. Moreover, the surrounding map is also revealed (R1 is satisfied).

Disambiguate arcs. It might happen that arcs might share a common space from a given perspective causing ambiguity in the visualization, as depicted in Figure 5.6(a). Hence, these arcs are shifted to the border of the lens and because of their intersection (yellow arrow on Figure 5.6(b)) the two sides of the arcs can no longer be traced. Although this drawback can be alleviated by applying different color for each arc, we instead decided to update the arcs's distance to the lens circumference based on its distance to the point of focus (see Figure 5.6(c)). This distortion can have negative implications in: (1) the *context arcs* that are near the lens area, (2) the *arcs of interest* and consequently their incident nodes. To limit the first case, all the *context arcs* are colored with a medium level of transparency. For the analysis of those visual elements, the user can move the lens over that location or increase the size of the lens. Regarding the second case, during the rendering phase the *arcs of interest* are drawn on top of the other elements.



(a) (b) (c)

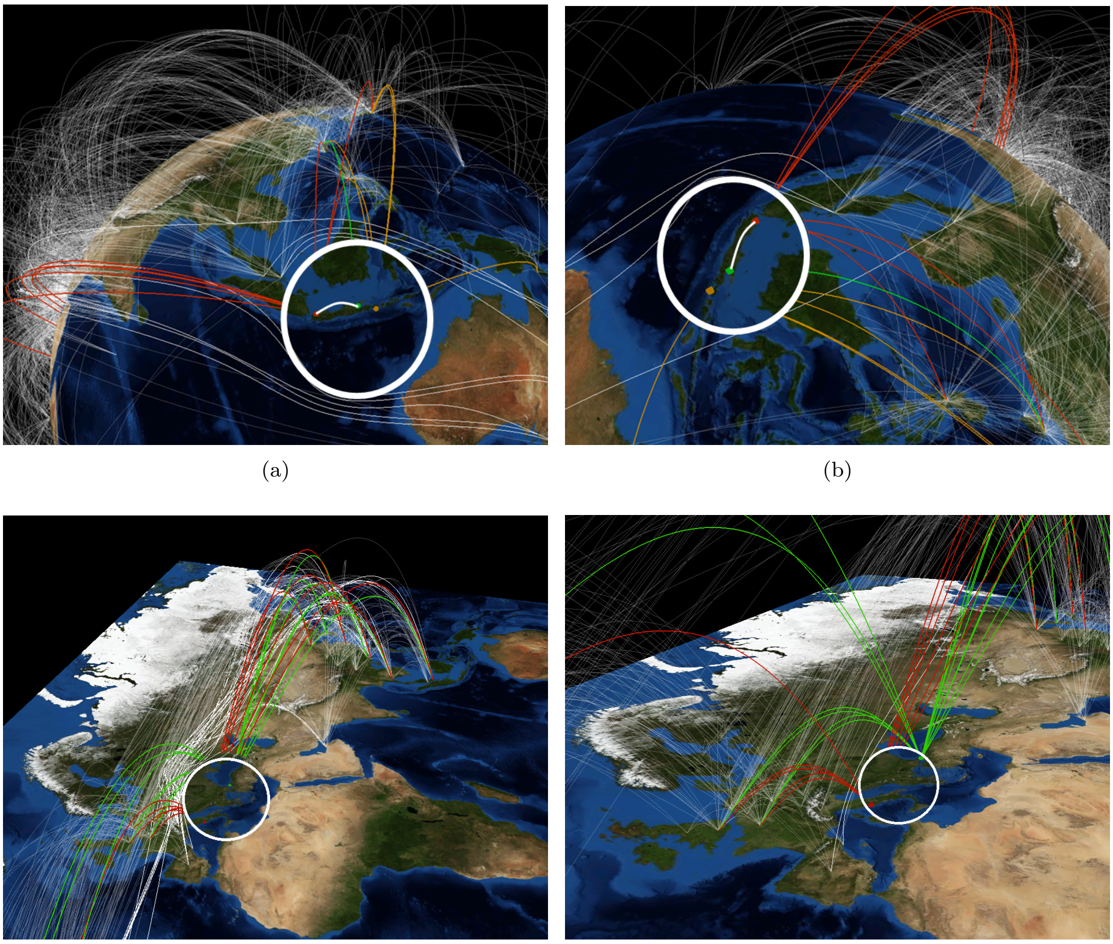
Figure 5.6: (a) Crossing arcs. (b) Crossing arcs deformed by the lens. (c) Arcs deformed by the lens with shapes easier to trace. In these images green arcs are *arcs of interests* and red arcs are *undesired arcs.*

If the user navigates, tilts or pans the camera, the lens still reveals hidden information of S' previously covered by the lens (i.e. S' remains *fully visible* thus R3 is satisfied). In fact, its movement is snapped to the terrain to improve the system usability; the camera movement affects the lens position in screen coordinates but not its geographical position, see Figure 5.7.

Moreover, the effect of the lens is transient, i.e. the deformed arcs return to their original shape when not marked as *undesired arcs.* If the user detects an area that requires further investigation and that is cluttered by arcs (i.e. S' is *visible* but not *fully visible*), then without moving the camera, he/she can drag the lens to that geo-location to reveal the hidden information (i.e. S' becomes *fully visible* thus R2 is satisfied).

5.4.1 Implementation Details

Each arc is designed as a relaxed cubic spline composed by a sequence of points $p_1, p_2, ..., p_n \in R^3$ that are defined by a sequence of equal-distance control points $cp_1, cp_2, ..., cp_m \in R^3$ where cp_1 is the source node and cp_m is the target node, see Figure 5.8(a). With cubic splines, it is possible to design polynomial curve segments controlling their shape in an



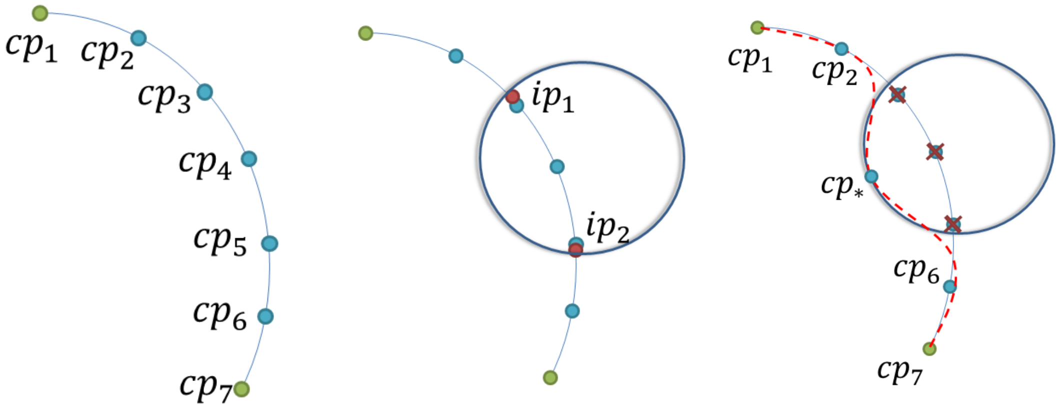
(a) (b) (c) (d)

Figure 5.7: Navigation and panning are used in combination with the lens to highlight the edges and the nodes over a given location. The camera view changes but the lens is always focused on the selected area. The lens can be applied on a globe (a)(b) or on a flat map (c)(d).

easy way and that pass through given data points.

The altitude of the control points over the geographical surface follows the formula of similar works [33, 121]: $1 + H \sin(x\pi)$ where H is the maximum height and x varies between 0 and 1 along the arcs path. The H parameter is equal to $l/4$ to take into account the path length l.

To detect whether a point that composes the arc or a node is contained inside the lens area, we use the two combined approaches: (1) in screen coordinates, we check if the distance between the element and the lens center is less than the lens radius. (2) using a ray-cast technique, we check if the line that connects the viewpoint with the element in the world space does not intersect the surface.



(a) (b) (c)

Figure 5.8: The steps required to apply the effects of the lens over the arcs are the following: (a) each arc is composed by m control points, (b) one or more couples of consecutive *intersecting points* are detected, (c) for each couple, a new control point is used to deform the arc.

Arcs deformation. The main aspect of the deformation of *undesired arcs* is that the original control points are removed and then new control points are created making curves more aesthetically pleasing and easier to trace. For this class of arcs, the algorithm is defined by two steps that are executed at every frame, before the main rendering phase:

Step 1. The *undesired arcs* are found and the *intersecting points* between these arcs and the circumference of the lens are detected. These points are defined as $ip_1, ip_2, ... \subset p_1, p_2, ..., p_n$ such that, in screen coordinates, they are the nearest to the real intersection points $\in R^2$. Then, the algorithm identifies the couples of consecutive *intersecting points* that will be used to create the new control points of the cubic curve, see Figure 5.8(b). This step is based on a GPU-based implementation to achieve an acceptable frame rate. The points that compose the lines are passed to the GPU into a single buffer. Then, a geometry shader computes the *intersecting points* to be passed to the next step using the *transform feedback buffer*.

Step 2. The sets of control points are updated and the arcs are redrawn. As shown in Figure 5.8(c), for each undesired arc, the control points inside the lens are removed and a new control point $cp_\star$ is created and placed over the lens' circumference.

The formula to calculate the screen coordinates of the new control point $cp_\star$ starting from the couple of points $\{ip_1, ip_2\}$ is the following:

$mp = (ip_1 + ip_2)/2;$

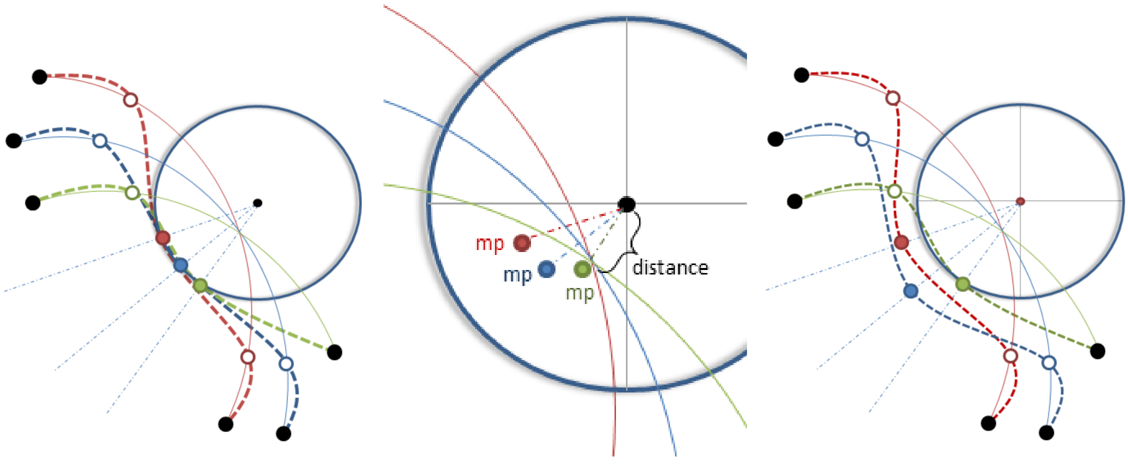
$cp_\star = c + r * (\widehat{mp} - c);$

mp is the middle point between ip_1 and ip_2, c is the center of the lens and r is the radius of the lens. The aforementioned formula is applied to the x and y attribute of $cp_\star$, meanwhile the z dimension, that is the screen depth-coordinate, is calculated taking the middle value between the z of ip_1 and ip_2. Finally, the deformed arc is drawn reconverting the $cp_\star$ in world coordinates and recomputing the points $p_1, p_2, \ldots, p_n$.

Control points relocation. The feature that reduces ambiguity when the shifted arcs can no longer be traced is shown in Figure 5.9(a) and described in the following steps:

Step 1. The lens is divided in p sections of equal area. For each new control point $cp_\star$, the distance in screen coordinates between its associated mp and the center lens is stored, see Figure 5.9(b). Therefore, for each section a list of control points ordered according to their distance is computed.

Step 2. Each control point is moved from the lens center, according to the distance as depicted in Figure 5.9(c). A maximum distance is defined to limit the effect in a closed area around the lens.



(a) (b) (c)

Figure 5.9: In the basic solution (a) new control points are placed around the lens. In the advanced solution (b) the lens is subdivided in sections. Control points are ordered according to the distance factor relative to their sector. (c) Control points are placed at different distances from the lens center around the lens.

The sector-based approach is applied to separate the control points that do not share the same location in screen space. For example, if there are two deformed arcs respectively on the opposite sides of the lens, due to the fact that are in different sectors, no control point will be relocated.

Arc segments filtering. For the creation of the *arcs of interest*, the steps are the following:

Step 1. As preprocessing phase, a different color is assigned for each node. During the rendering phase, if the node is inside the lens and it is visible, than a sphere colored with the node's color is generated.

Step 2. Every part of the arc inside the lens is drawn completely transparent using the *blending function*, and the rest of the arc is colored with the same color of its incident node inside the lens.

We added two graphical components to the GUI to facilitate the interaction with the system: a slider interface allowing the user to readjust the lens radius and a button to place the lens at the center of the screen. This last feature is handy when the user loses sight of the lens while navigating.

5.4.2 Input Parameters

The parameters required by the algorithm affect directly the quality of the result.

The first parameter is the number of control points m. If the number of control points is too high, the deformation technique affects only a small localized area. As opposite, if m is too low, the edges are deformed along their entire length. As depicted in Figure 5.10, the advantage of using low number of control points appears evident, however, if the geographical surface is over a globe and both incidents nodes are placed in different hemisphere, there are extreme cases where the curve may pass through the geographical surface. For such reason, we set m equal to 7, to avoid to overlapping with the globe.

The second parameter is the number of sectors p. If p is high, only the control points close with each other will be placed with different distances from the lens center. As opposite, if the value is too low, more layers will be used. This decision is up to the user. In our tests we used $p = 4$.

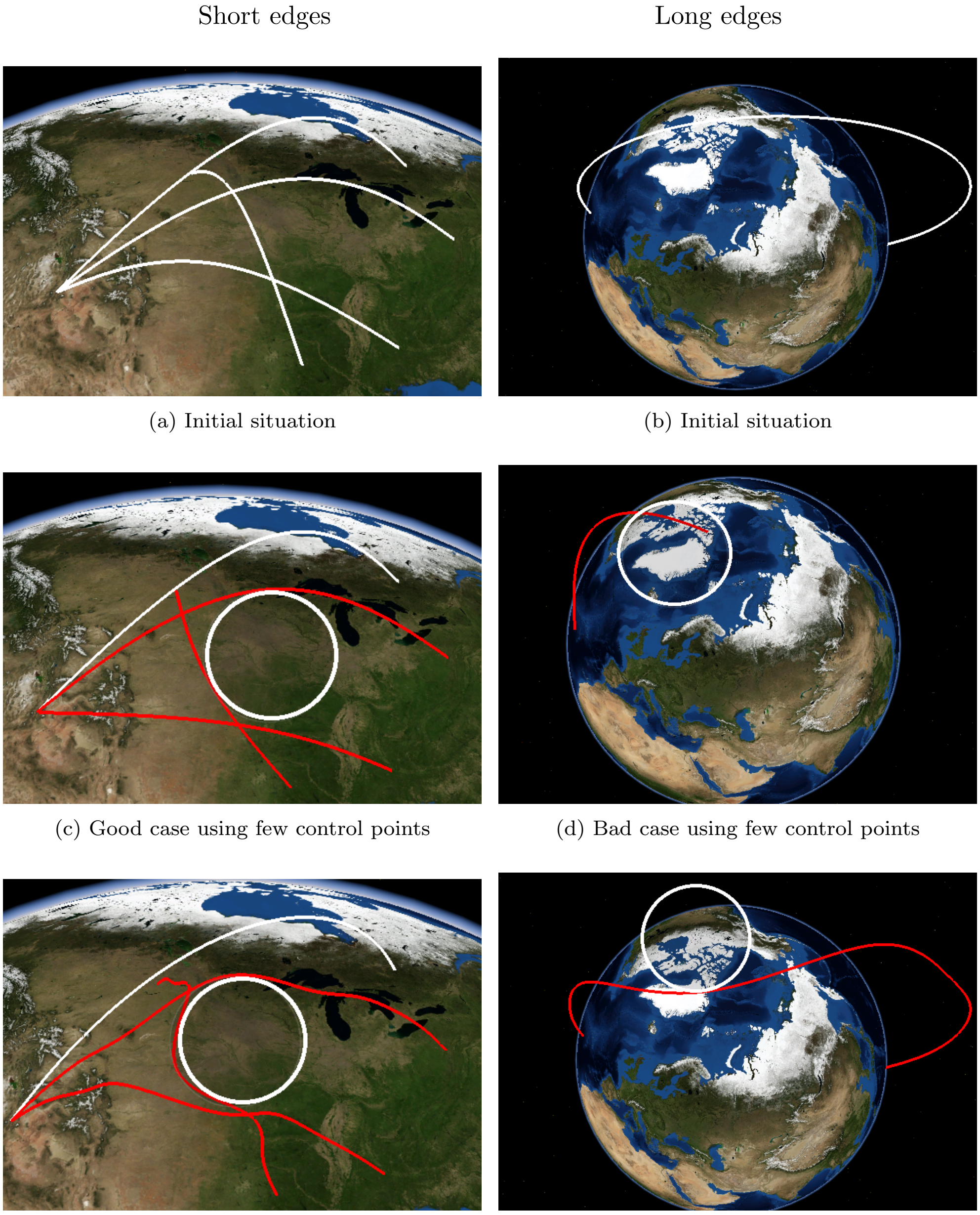
Another parameter is the number of points n that compose each arc. Higher the value is and smoother will be the curve. We set $n = 200$ to have a good balance between the aesthetic result and the performance.

The last parameter is the radius of the lens r. Since this value has a considerable impact on the effectiveness of the proposed technique, the user can modify the lens size during the exploration of the network.

5.4.3 Graphical Performance

3DArcLens was developed extending *NASA WorldWind*[2], an open source virtual globe, and was tested on a single processor Intel Xeon 2.26 GHz equipped with an NVIDIA GeForce GTX 280 with 1024 MB dedicated video memory. We take as metric the frame rate to measure the graphical performance of our technique. Table 5.1 shows the *fps*

[2] `http://worldwind.arc.nasa.gov/java/`



Short edges

Long edges

(a) Initial situation

(b) Initial situation

(c) Good case using few control points

(d) Bad case using few control points

(e) Bad case using many control points

(f) Good case using many control points

Figure 5.10: Two situations, one with short length edges (left column) and one with long edges (right column). Examples showing advantages and drawbacks of using high and low number of control points. In these images, the red color is used to depict *undesired arcs*.

on *WorldWind* with and without *3DArcLens*, taking into account datasets with different number of arcs *tot arcs*.

Table 5.1: The *fps* are shown with and without the use of *3DArcLens* on *WorldWind*.

tot arcs	fps (WW)	fps (WW + 3DArcLens)
600	55 - 56	20 - 22
3000	33 - 35	15 - 17
15000	8 - 10	6 - 7

The performance of *3DArcLens* depends mainly on the number of arcs that are distorted. The Table 5.2 shows the *fps* on *WorldWind* taking into account the deformed arcs of the dataset with 3000 arcs.

Table 5.2: The *fps* and the number of deformed arcs are shown on *WorldWind*.

def arcs	fps (WW + 3DArcLens)
0	21
150	18
300	15
400	13

5.5 Case Studies

We test our technique on two different case studies: monitor network infrastructure and analyze mobile phone records.

Monitor network infrastructure. Maintainers need to accurately understand the large-scale geographic structure of the Internet infrastructure. They want to explore interesting patterns on the global level, as well as details up to city level. Internet is composed to millions of routers. Hence, the user needs to analyze the network on a higher level of abstraction; the dataset is preprocessed such that the nodes of the graph are *Autonomous Systems* (AS), i.e. groups of routers. This network covers the entire world and presents dense areas because ASes are not distributed evenly. Their geolocation is derived from the IP addresses. Each node has a *traffic degree*, i.e. the number of unique neighbors that appear on either side of an AS in adjacent links [109]. The links are their direct connections with a traffic degree corresponding to the highest degree from their connecting nodes.

The user is supported with a node-link representation over a globe. The nodes with low transit degree were filtered out to avoid over-plotting. The graph is composed to 220

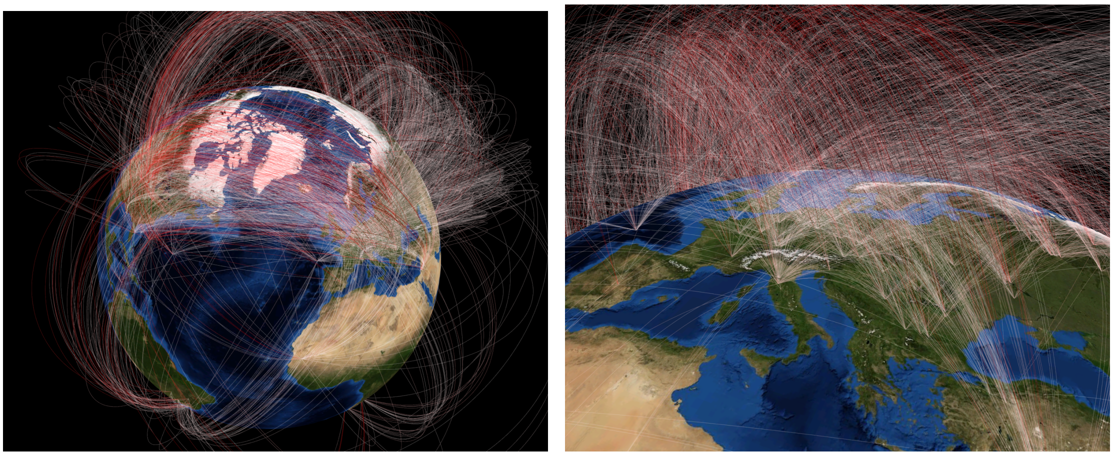


Figure 5.11: Node-link diagram over a globe. Each node is a group of routers. The color of the links expresses the data traffic. The graph is characterized by dense areas.

nodes and 4842 links[3]. The link color is based on its transit degree and may vary from a white-red gradient (white color means low degree and red color means high degree). The links are also colored with a medium level of transparency to reveal the background map, see Figure 5.11.

In this visualization, the large number of links in small areas cause occlusion of important information. There are also ambiguities on the graph representation: it is not easy to understand whether a link is connected to a node or pass through it. Moreover, two arcs may occupy the same screen space.

The user can drag 3DArcLens over the area of interest and can increase/decrease its size using the keyboard buttons, see Figure 5.12. Furthermore, the links that are completely inside the lens became visible because the segments of the other links that are inside the lens are not drawn. Here, the lens is able to reveal information about groups of ASes located in dense areas. On the other hand, if the task is the exploration of long connections from ASes inside the lens, the user has only to move the camera view without taking care of the lens object, see Figure 5.13. The links in analysis are colored without transparency to facilitate the task.

From the visual exploration of the network, it was noticed that dense groups of interconnected nodes are mainly located in the center of Europe, in Asia, and in the USA. Moreover, the ASes with highest traffic degree are located in the United States.

Analyze airport connections. Airplane trajectories recorded by the air traffic authorities enable to analyze traffic or devise new ways of managing airspace. The graph

[3] Dataset made publicly available by CAIDA Center for Applied Internet Data Analysis, `http://www.caida.org/home/`.

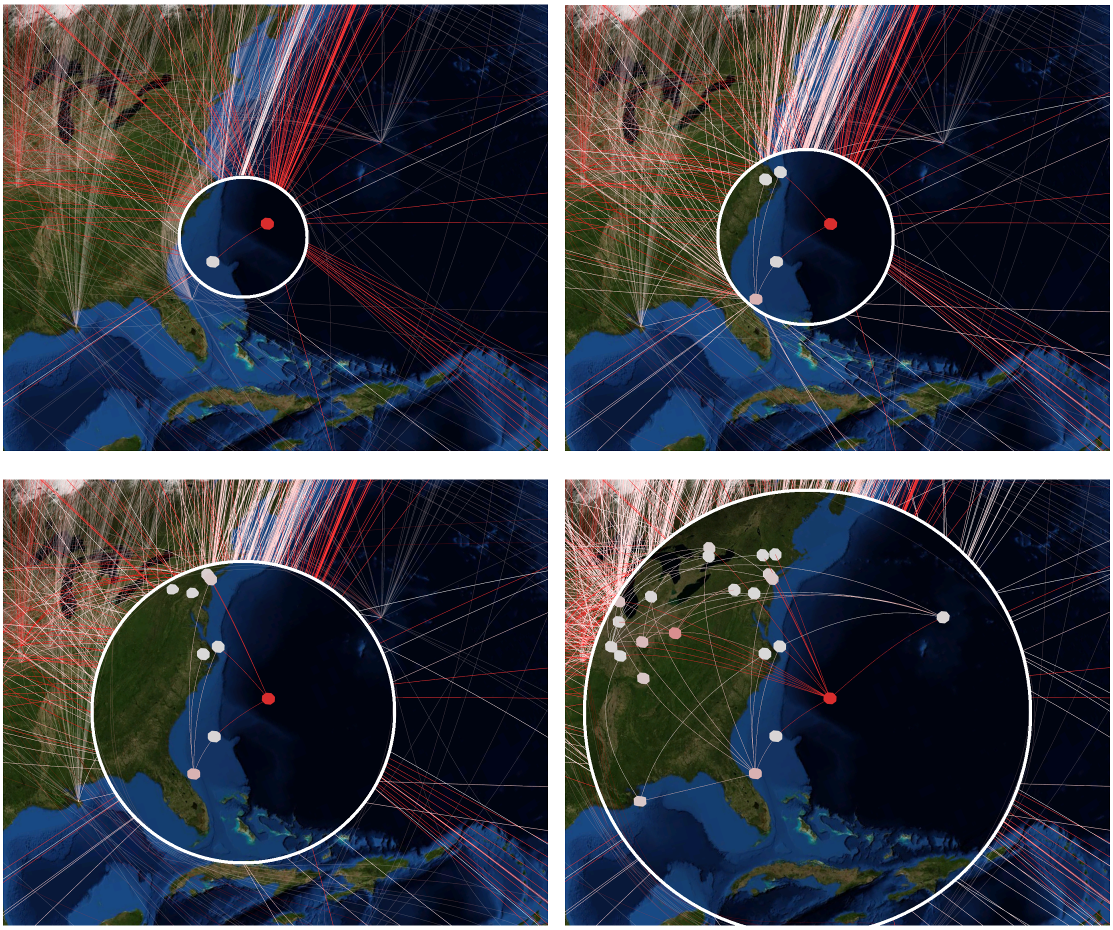


Figure 5.12: 3DArcLens reveals connections among ASes in a dense area. The user can increase/decrease the lens size.

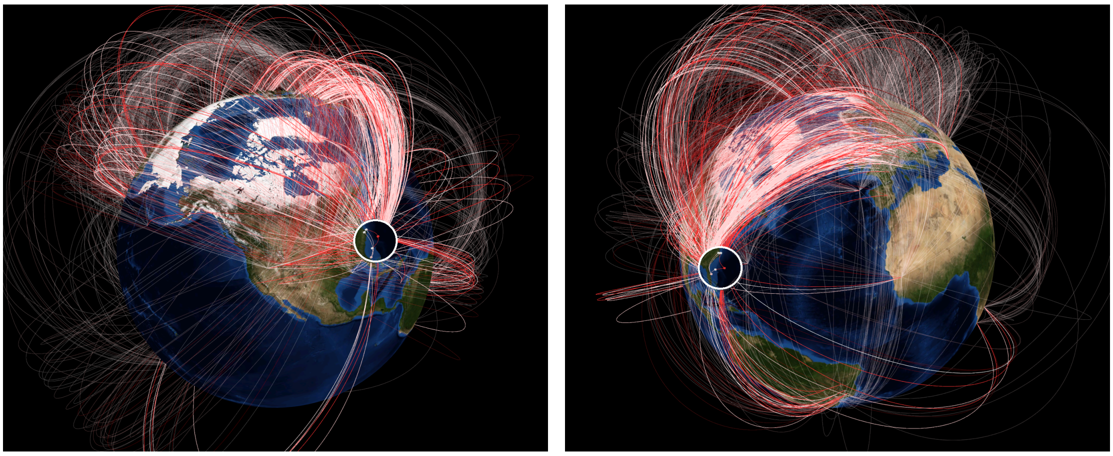


Figure 5.13: 3DArcLens highlights long connections with ASes in the lens focus. The user moves the camera view without taking care of the lens object.

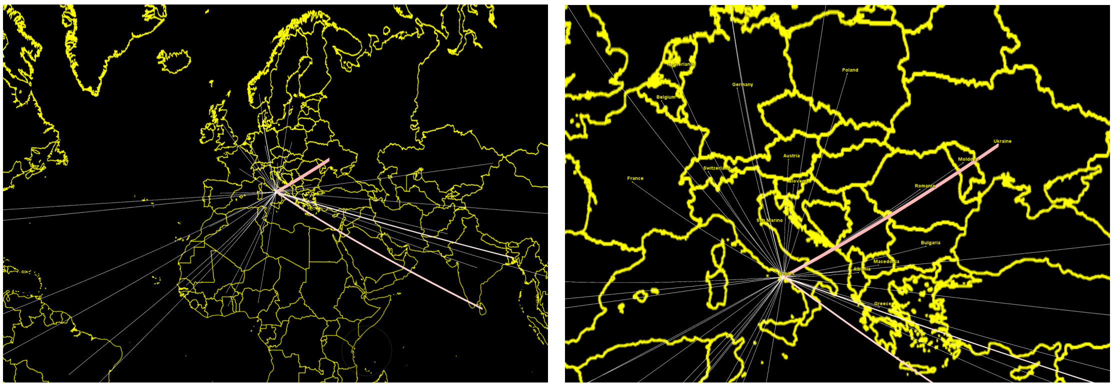


Figure 5.14: Node-link diagram over a flat map. Each node is a destination of outgoing calls from Naples in a range of three months. The color of the links expresses the number of outlier days and the thickness expresses the magnitude of calls over the entire time range. The graph is characterized by many overlapping links.

represents the air traffic[4]. Vertices are airports and edges are direct flights. The edges are derived from actual flight paths; since flights start and end in airports, edge endpoints are spatially grouped. As shown in Figure 5.7, respect to the previous case study, the color of nodes and links is used to facilitate the task “find connected airports”.

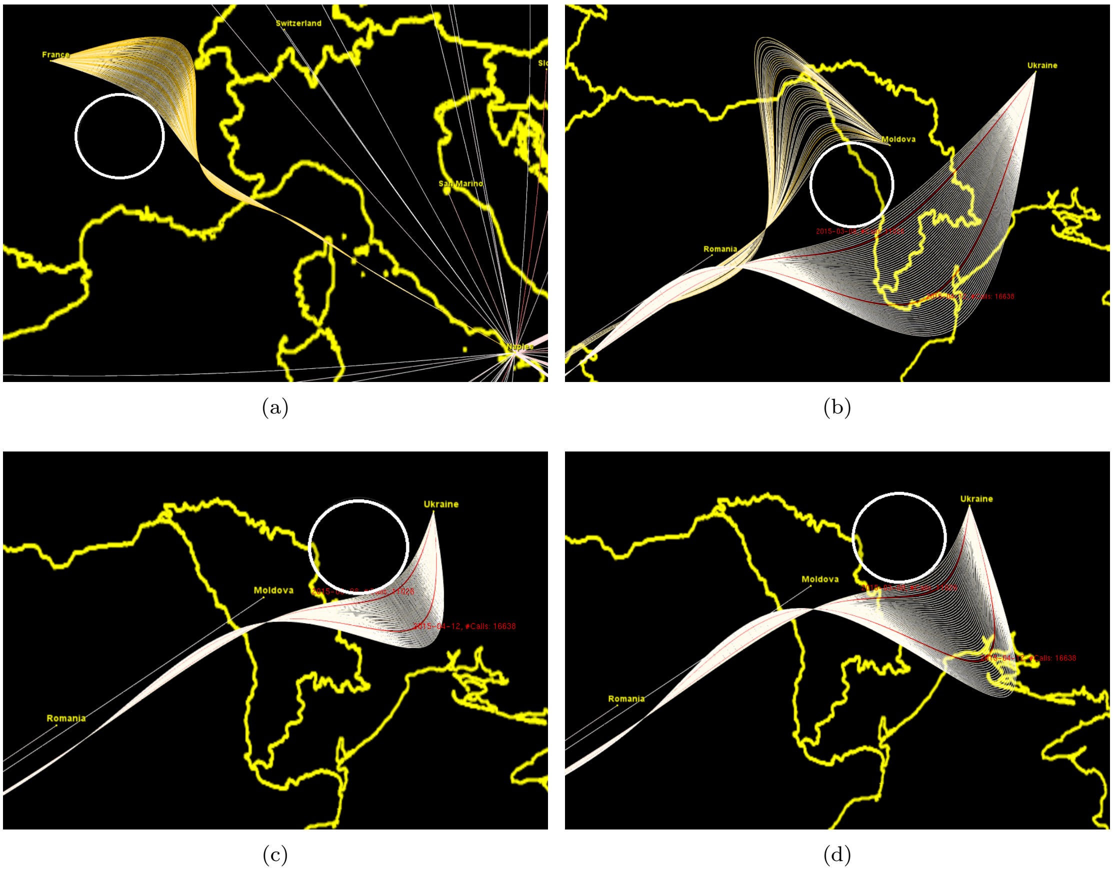
Analyze mobile phone records. In this case study, the data analyst wants to visually explore international outgoing calls from Naples registered in three months with 1-day granularity[5]. In particular, he/she wants to know: which are the main destinations of outgoing calls, which are the destinations with more *outlier days* (i.e. days with a high number of calls with respect to the other days), and which is the trend of outgoing calls during time.

The dataset is represented as a node-link diagram located on top of the world map. Each link represents the number of outgoing calls from Naples to another country in a specific day. The link color may vary from a white-red gradient (white color means few outlier days and red color means many outlier days). Their thickness is proportional to the number of calls over the entire time range on the target destination. The visualization is characterized by a huge amount of overlapping links; links with the same destination but referred to a different day will completely overlap, see Figure 5.14.

Although this is not considered the suitable visualization to answer the above questions, we use this case study to demonstrate the effectiveness of 3DArcLens in situations of overlapping links. As shown in Figure 5.15(a)(b), the lens disambiguates the overlapping links revealing temporal information for a specific country. Respect to the previous case

[4] Dataset made publicly available by OpenFlights, `http://openflights.org`.

[5] Dataset made publicly available by Telecom Italia for the *Big Data Challenge* `http://www.telecomitalia.com/tit/it/bigdatachallenge.html`.



(a) (b)

(c) (d)

Figure 5.15: 3DArcLens disambiguates the overlapping links revealing temporal information for a specific country.

study, the lens behavior is modified to reveal additional information; the color of the deformed links identifies the number of calls for a specific day. It may vary from a white-yellow gradient. White links have few calls and yellow links have many calls. Meanwhile, red links identifies outlier days. The deformed links are also ordered around the lens with respect to their day.

With the keyboard, the user can vary the screen space dedicated to the deformed links and the lens size, see Figure 5.15(c)(d).

5.6 Discussion

Chapter 3 described different interaction techniques aiming at reduce clutter and ambiguous situations. In this section, we compare our approach with three interactive lenses: *EdgeLens* [187], *MoleView* [90], and *Local Edge Lens* [164].

EdgeLens is a technique that interactively curves graph edges, that are represented by straight lines, away from the point of focus in a flat plane. Although it helps to reduce the occlusion issue and disambiguate the relationship between nodes and links without losing information, it differs from *3DArcLens* in the following aspects:

Arcs shape. EdgeLens is not designed to work with 3D arcs. In a 2D space, a straight line is deformed by taking into consideration the start and end points (together with the lens focus and radius). As opposite, with 3D arcs the distortion and selection of these objects is more complex.

The view camera information. EdgeLens is not designed to work in 3D virtual environments, i.e. with camera movement techniques such as tilting. As opposite, the behavior of our lens depends also on the position of the view camera; if the view camera changes, the arcs affected can differ in both number and shape.

The occlusion problem. In 3D virtual environments, the effect of occlusion has to be considered; arcs that intersect the lens having their extremities behind a geographical surface can generate undesirable behaviors.

The coordinate systems. In 3D virtual environments, the distortion process must take into account the screen coordinates and the world coordinates.

MoleView is another interactive technique to reveal what is hidden behind a congested area. The control points of the links are moved around the lens meanwhile the links which are selected in the lens do not move, making them easy to spot. Although this technique gives satisfactory results, it differs from *3DArcLens* in the following aspects:

Information about arcs. MoleView moves the control points inside the lens. As opposite, *3DArcLens* puts more effort on the traceability of arcs. In particular, the control points affected by the lens are replaced with others that are placed in different "orbits" around the lens.

Geographical context. Our technique allows both the arcs and the map inside the lens to be depicted without visual clutters.

Lens geo-position. In our technique the geographical position of the lens is considered. This implies that, if the lens highlights a geographical area and the camera view moves, then the lens remains still. However, the way it reveals what is hidden in that area maybe differ.

Local Edge Lens is an interactive technique that draws in a local focus region only links that connect certain nodes within the lens scope. It is designed to reveal nodes occluded by context links and to reduce ambiguity caused by overlapping links.

Ambiguity problem. 3DArcLens gives more emphasis to the context links that are deformed. As opposite, Local Edge Lens simply not draw them.

Limitations. Although choosing suitable colors is fundamental for an effective utilization of *3DArcLens*, the definition of a proper color pattern is not the aim of this work.

Moreover, the effectiveness of the colors assigned to nodes depends on the task that has to be accomplish. For example, one task might involve information about node attributes that can be encoded in a color pattern defined at priori.

A potential shortcoming may arises if the dataset presents a large number of connected nodes in a small location that is totally covered by the lens. In such case the visual clutter is caused by the *arcs of high interest*. However, if the *area of interest* is reduced changing the lens radius or zooming in, this issue can be considerably alleviated.

5.7 Conclusion

In this chapter we describe *3DArcLens*, an interactive technique for exploring both geographical networks and geographic surfaces.

Our approach can be applied to virtual environment such as virtual globes and flat maps with 3D arcs that are considered common tools for the analysis of geographic networks [33, 121, 26]. *3DArcLens* works also on simple 2D virtual environments.

In our case studies, we observed certain advantages in applying our technique to the exploration of geographical networks. Overall, our technique interactively distorts and filters links, revealing underneath information about: nodes, links and geographical surface. However, since it is locally applied to a specific area, higher-level tasks such as finding connectivity patterns remain difficult to achieve. Also, for datasets with ∼3000 edges the frame rate of our prototype was acceptable. Yet, it might drop for much larger datasets. Future improvements include shifting the generation of the splines to the GPU side and the definition of an evaluation strategy focusing on the user.

A further case study that can benefit of *3DArcLens* is the exploration of diagrams with bundled links; as mentioned in Chapter 4, bundling techniques can reduce clutter but as drawback they cause occlusion and ambiguity on the graph representation.

Chapter 6

GeoPeels: Deformation-Based Technique for Exploration of Geo-Referenced Networks

Computer science is no more about
computers than astronomy is about
telescopes.
–Edsger Dijkstra

6.1 Overview

Spatial relations between geographical entities are typically visualized as a node-link diagram that is depicted over a geographical layout. As mentioned in Chapter 2, the node-link diagram representation is well-suitable for tasks that require the analysis of the topology and properties of a spatial graph because it is more intuitive with respect to the matrix representation. However, the readability of node-link diagrams is one of the issues studied in literature. In case of global networks, the depiction of links on a flat map can be a misleading approach. This because the area of interest is the entire Earth, so one or more "cuts" are required on the entire surface to "unfold" it and project it on a plane. This aspect forces us to redesign the links that pass over a cut. Such aspect can be performed using projections that preserve the links properties or using virtual globes. The latter are considered mature visualization systems for the exploration of geo-referenced networks. However, in virtual globes, we can visualize at most one hemisphere at a time. In this chapter, we describe an interactive technique for virtual globes called *GeoPeels*, which uses spatial deformations to reveal spatial information otherwise hidden behind the geographical layout. Consequently, with respect to virtual globes, our approach reduces the overall number of camera-view movements required to visualize and understand the spatial relations, facilitating the graph-related tasks. The deformed geometries are rendered around the globe, therefore, occupying space that otherwise would be unused. The evaluation of our approach is based on the analysis of the graphical performance, on a comparison between our technique with existing approaches, and testing it within some case studies. Part of the work in this chapter was previously published in [44, 43][1].

6.2 Introduction

Map visualization systems of geographical networks retain their traditional role of providing insight into geospatial patterns and relations. On the one hand, they provide geospatial contextualization for each entity. On the other hand, they can be used to enrich entities with graphical elements like straight lines, circular arcs and curves, to give emphasis to specific spatial relations.

A representative scenario is the monitor of network infrastructures: a maintainer wants to understand the large-scale geographic structure of the network infrastructure. He/she wants to explore interesting patterns on the global level, as well as details up to city level. The navigation has to be in real-time and the visualization system has to provide a familiar context.

To navigate in presentation spaces, when the network is limited to a small region, the flat map with cylindrical projection is a widely accepted approach [116]. Moreover, different techniques were used to facilitate the exploration of networks such as zooming & panning, focus+context [130, 131, 60], overview & detail [32], and multi views [22, 18].

In case of global networks, e.g. supply chain networks [22], networks of climate stations [165], flight connections [83, 150], migration flows [161, 18, 73] and Internet traffic [118, 33], the depiction of links on a flat map can be a misleading approach. This because the area of interest is the entire Earth, so one or more “cuts” are required on the entire surface to “unfold” it and project it on a plane. This aspect forces us to redesign the links that pass over a cut. A link can be splitted in two parts or can be drawn in another direction.

In the first case, when the display space contains the entire map, the user has to mentally reconnect the link, increasing his/her mental strain. This issue is aggravated if links are high in number. Alternatively, the use of an imaginary infinite plan would cause redundant information related to nodes, arcs and map features.

In the second case, when the camera view is focused on a small region, the link would give a misleading direction of its off-screen connected node. As opposite, with the nodes both visible, the link would cross the entire map giving a misleading impression of the actual geographic distance between nodes [26].

[1] As additional material, a video showing a demo of the presented technique was included in the related published papers. It is available online at `https://www.youtube.com/watch?v=Hhg88Am-kDA`.

As described in details in Section 6.4, such aspects can be alleviated using projections that preserve the links properties or using virtual globes. The latter are considered mature visualization systems for the exploration of geo-referenced networks. In the past, they were used by Cox et al. [33] to visualize Internet traffic over the NFSNET/ANSnet backbone and by Munzner et al. [121] to map the MBone routers. In climate research domain, the analysis of climate data as complex network theory is considered a powerful approach [51]. In such context, the globe is commonly used [165, 124]. Globes are also used in frameworks for management and monitoring of large scale distributed systems [106]. With respect to general 3D visualizations, globes allow the user little chance of becoming disoriented [13, 57]. As drawback, in virtual globes, we can visualize at most one hemisphere, so nodes and links may become occluded.

In this chapter, we propose *GeoPeels*, a distortion-based technique for the visualization of spatial relations in geographical data, where the occlusion issue is solved. Our technique is optimized for graph-related tasks, such as explore node connections, for global datasets characterized of groups of nodes.

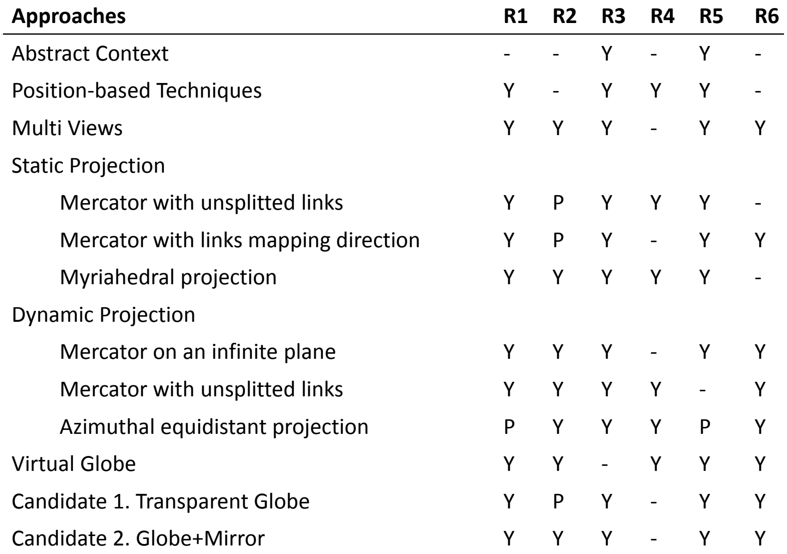
In summary, *GeoPeels* provides an answer to the following research question: *Given a geographic node-link diagram on a globe, how can we reveal nodes and links of interest that are occluded by the globe shape without moving the camera view and preserving the links and the geographical context of the nodes.*

Our approach is based on a virtual globe that automatically deforms the terrain surface to avoid the occlusion of key data nodes and links. Moreover, the terrain deformation is automatically adjusted to alleviate the issue of geometries rendered outside the screen view. Instead of reveal the entire Earth surface, only regions considered important are deformed. In this way, likewise to the interrupted projections, the distortion is reduced. *GeoPeels* requires information about the camera view, nodes and links.

In the exploration of spatial relations, our technique is used in combination to zoom and pan. Furthermore, *GeoPeels* does not require the operator to define visual node-link attributes, e.g. we do not use attributes like color and size of a node. Hence, the latter can be used to encode additional information. Alternatively, our technique can be combined with other strategies, for example, to reduce the visual clutter.

This chapter is structured as follows. Section 6.3 itemizes a list of requirements that a technique for the exploration of geo-network should take care. Section 6.4 describes previous related work and alternative novel approaches that may partially fulfill the above requirements. Section 6.5 introduces our approach. Section 6.6 describes some case studies to illustrate our technique. Section 6.7 explains the list of parameters and how to use them. Section 6.8 provides an analysis of the graphical performance. Section 6.9 describes our approach with respect to the aforementioned requirements. We conclude with some

Table 6.1: Design requirements fulfilled by existing strategies and alternatives approaches.



Approaches	R1	R2	R3	R4	R5	R6
Abstract Context	-	-	Y	-	Y	-
Position-based Techniques	Y	-	Y	Y	Y	-
Multi Views	Y	Y	Y	-	Y	Y
Static Projection						
Mercator with unsplitted links	Y	P	Y	Y	Y	-
Mercator with links mapping direction	Y	P	Y	-	Y	Y
Myriahedral projection	Y	Y	Y	Y	Y	-
Dynamic Projection						
Mercator on an infinite plane	Y	Y	Y	-	Y	Y
Mercator with unsplitted links	Y	Y	Y	Y	-	Y
Azimuthal equidistant projection	P	Y	Y	Y	P	Y
Virtual Globe	Y	Y	-	Y	Y	Y
Candidate 1. Transparent Globe	Y	P	Y	-	Y	Y
Candidate 2. Globe+Mirror	Y	Y	Y	-	Y	Y

remarks about the usability of the proposed technique and future work in Section 6.10.

6.3 Design Requirements

We define a list of requirements that a visualization should take care to facilitate the interactive exploration of geographical networks:

R1. *Geo-context awareness.* The visualization should provide a familiar geographical context, allowing the user to recognize easily the location of each visible feature (e.g. the nodes of the network) [160].

R2. *Surrounding context preserved.* In every frame, a portion of the area surrounding each visible node should be visible [60].

R3. *Topology awareness.* In every frame, the items (i.e. nodes and edges) should be visible simultaneously (or at least should be provided some visual cues to indicate their position), regardless of their location on the presentation space [116, 69, 60].

R4. *Spatial continuity.* The geographic space between visible nodes should be shown to give a frame of reference [60]. The continuity involves also the links; if a link is divided in different segments, the user has to mentally reconnect them, which increases the mental strain. Moreover, nodes and links should not appear duplicated in the same frame (i.e. with no redundancy).

R5. *Temporal continuity.* The navigation has to be a continuous process in time (with responsive frame rate), allowing the user to maintain his/her *mental map*, i.e. the cognitive model of the relationships between graph elements that he/she forms when viewing the node-link diagram [56, 114]. In the case of links, they cannot disappear on one frame and then appear in a non-coherent position.

R6. *Direction awareness.* During the navigation, some notion of the real direction of nodes of interest located off-screen should be available [60].

6.4 Candidate Techniques

Existing approaches together with novel possible alternatives that support partially the aforementioned requirements, are described in this section and summarized in Table 6.1.

Abstract context. According to req. R1, the node-link diagram should be visualized together with a background map. Hence, techniques with no spatial context, such as origin-destination matrices [188], or with a generic spatial context such as ODMaps [188] are not considered.

Multi Views. According to req. R4, nodes, links and the map have to be located in a single view in which the map does not present interruptions. Thus, strategies based on multi-views are not examined.

Position-based Techniques. The context is important, so techniques that relocate nodes without taking care to their surround areas, such as force-directed techniques [94] or interactive lenses like Bring Neighbors Lens [164] and Bring & Go [116], are not recommended (req. R2).

Static projection. Visualization systems, such as Arc Map [33], that use a static projection, i.e. with no possibility to change the way the Earth surface is cut and unfolded, are able to fulfill almost all the requirements. However, the surrounding context of nodes located on the boundaries of the map is not preserved. Moreover, some links may not express the real direction of off-screen nodes in real space. For example, in the standard projection with Europe located at the center of the map, if the user focuses on Alaska, all the visible links directed towards both Sweden and Russia will point to the same direction, see Figure 6.1(a). Van Wijk presented *Myriahedral projections* [173], that rely on methods for mapping the earth on a polyhedron with a large number of faces. The polyhedron is then cut open and unfolded. The result are maps that conserve areas and that are (almost) conformal. It uses interrupts to reduce the distortion. A Myriahedral projection could be designed to create a 2D map that selects the “cuts” on the globe, such that the spatial continuity among connected nodes and the context of regions of interest are preserved. As drawback, it should require a customized visualization for each network in input, see Figure 6.1(b).

Dynamic projection. A popular projection used to facilitate the navigation on geographical networks is a cylindrical projection, such as the Mercator, that is mapped on an infinite plan. On the one hand, if links are always depicted as unique lines, their length would not always represent the real direction of the connecting nodes, as mentioned before. Moreover, changing the camera view, the links will be redrawn with no temporal continuity, for example from Figure 6.1(e) to Figure 6.1(f). On the other hand, if links are drawn such that they always express the direction among connected nodes, it may happen that links will be composed to separated lines, for example from Figure 6.1(e) to Figure 6.1(d).

If the aim is visualize focus nodes together with their connected nodes, the use of azimuthal equidistant projection with the focus nodes in the center of the map is a recommended strategy, see Figure 6.1(c). However, such projection is not popular to explore geographical networks and the distortion of areas and shapes dramatically increases with distance from the center point. Consequently, the resulting map may not be perceived as a familiar geographical context.

Virtual globe. Compared to flat maps, globes enable continuous navigation, preserving the spatial coherence of nodes and links [53]. Moreover, with respect to general 3D visualizations, globes allow the user little chance of becoming disoriented [13, 57]. The main drawback is the occlusion of nodes and links caused by the globe shape. In fact, at most one hemisphere is visible at a time. To solve the occlusion issue of globe visualization, we studied different possible candidate solutions.

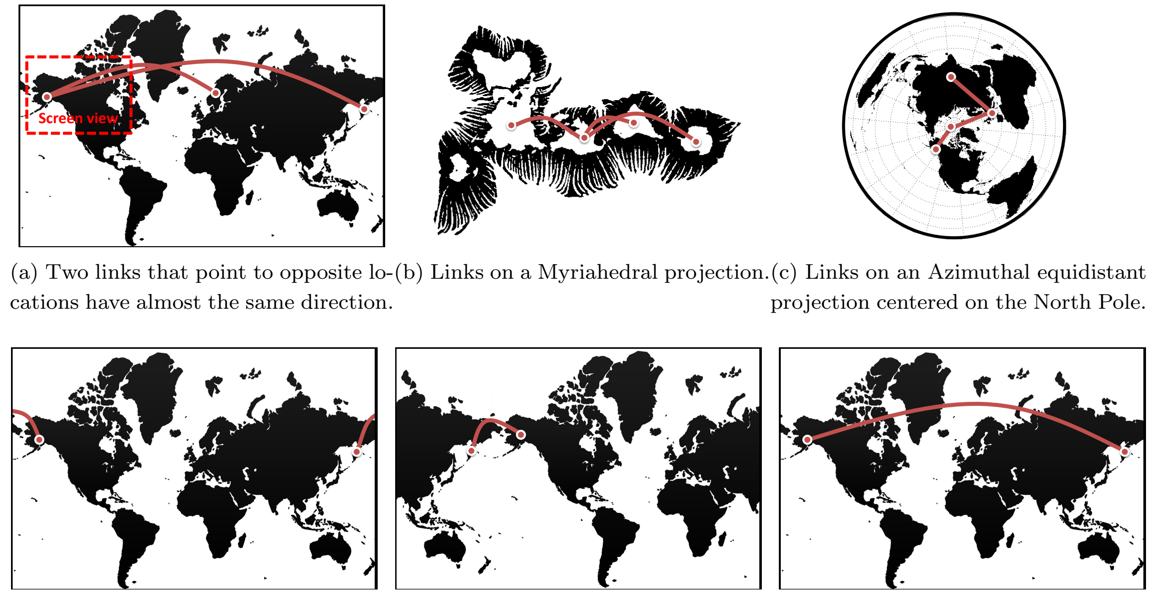
Cantidate 1. Transparent globe. Hidden nodes may be revealed applying transparency to portions of the visible surface. As drawback, the revealed areas are not geographically coherent with the surrounding areas on the visible hemisphere.

Cantidate 2. Globe+Mirror. Another approach consists in the use of two views, each showing one hemisphere, making all nodes visible. The views could be located in separated spaces or integrated in the same presentation space. The latter could be designed locating a "mirror" in the 3D space, behind the globe. In both cases, the context switching is a negative aspect to consider; the user is forced to split his/her attention between the views, increasing the cognitive load [179, 72].

6.5 Proposed Approach

In this work, we define a deformation-based technique that enables the interactive exploration of spatial relations between geographical entities; in particular to facilitate graph-related tasks, e.g. exploring the node connections.

As discussed in Section 6.4, the virtual globe has different advantages in terms of navigation, and awareness of context and topology. We propose a technique that enriches the idea of a virtual globe with the following features: a list of *important nodes* together with



(a) Two links that point to opposite locations have almost the same direction.

(b) Links on a Myriahedral projection.

(c) Links on an Azimuthal equidistant projection centered on the North Pole.

(d) Link composed to 2 separated lines.

(e) Link expressing the correct direction between nodes.

(f) Link expressing a fake direction.

Figure 6.1: Interactive Visualizations of geographic networks.

their surrounding areas, should be always occlusion-free. We formally define important nodes according to the concept of a *Degree of Interest (DOI) function*. In literature, it is used to rank the nodes located off-screen according to their interest for a particular task [66, 69]. Here, we use such function to reveal nodes occluded by the globe.

In order to relocate *important nodes* that are occluded by the globe's hemisphere, without losing their background map context, we defined the concept of *slice*, which corresponds to a portion of terrain that is occluded by the globe and that contains at least one *important node*.

In the next sections, we use the terminology *deformed slice* to describe the geometry of a slice that was "peeled" from the globe surface – which is represented as a continuous concave surface with respect to the camera view. The concept of *deformed slice* can be demonstrated with an orange: we can peel an orange to reveal its hidden parts, see Figure 6.2.

The process to create and deform a slice is transparent to the operator. It entails the creation of *slices* for *important nodes* that are not visible. The first node that is added to a *slice* is called its *primary node*. Slices are created only if an occluded *important node* can not be moved into an existent slice. This strategy ensures that nodes are never occluded by the globe and the number of *deformed slices* is minimized.

The shape of each *slice* is defined as a rectangular geometry with the *primary node*

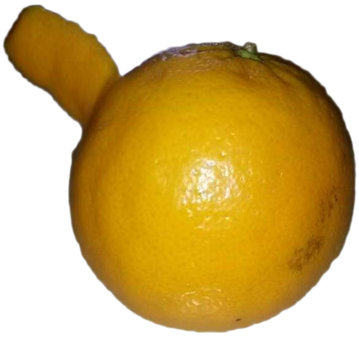


Figure 6.2: The orange peel is the metaphor that inspired this work.

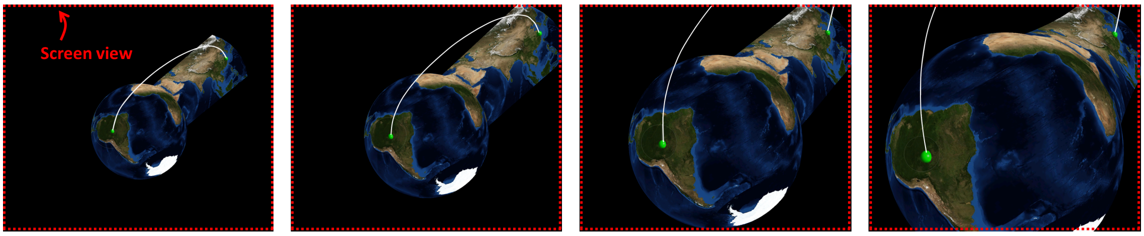


Figure 6.3: Each *deformed slice* is bended to fit inside the screen view.

located at the center with respect to the horizontal axis, and at a fixed position with respect to the vertical axis. As additional feature, during the interactive exploration, the *deformed slice* is bended to ensure the visibility of the geometry within the screen view, see Figure 6.3.

We designed also an extension that performs a spatial selection of the important nodes, that are the nodes connected to the ones *in focus*. We define *in focus* the list of nodes that are inside the user-defined area. Remaining nodes are said to be *in context*, see Figure 6.4. In this case, the DOI function is a binary function that assigns 1 to all neighbors of currently *in focus*, and 0 to all others. Moreover, the nodes in the hidden hemisphere that are not *important* are intrinsically filtered out, even if located in the deformed slice.

The focus area is represented as a circle in screen space with a radius r in pixels. The circular element remains still during the user navigation, e.g. tilt, pan and zoom. The circle is snapped to the terrain to improve the system usability; the camera movement affects the circle position in screen coordinates but not its geographical position.

GeoPeels was developed on top of *NASA WorldWind*, an open source virtual globe that approximates the earth as a sphere. The proposed technique makes use of a spherical reference system, which is described in Subsection 6.5.1. The overall procedure of

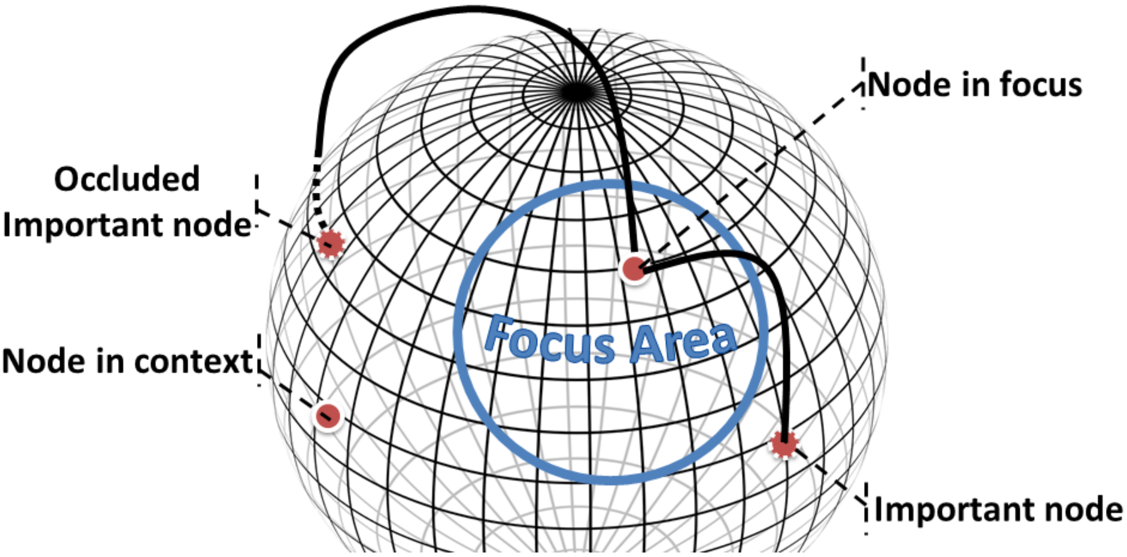


Figure 6.4: Extension of GeoPeels: nodes are classified according to their position and the position of their connected nodes.

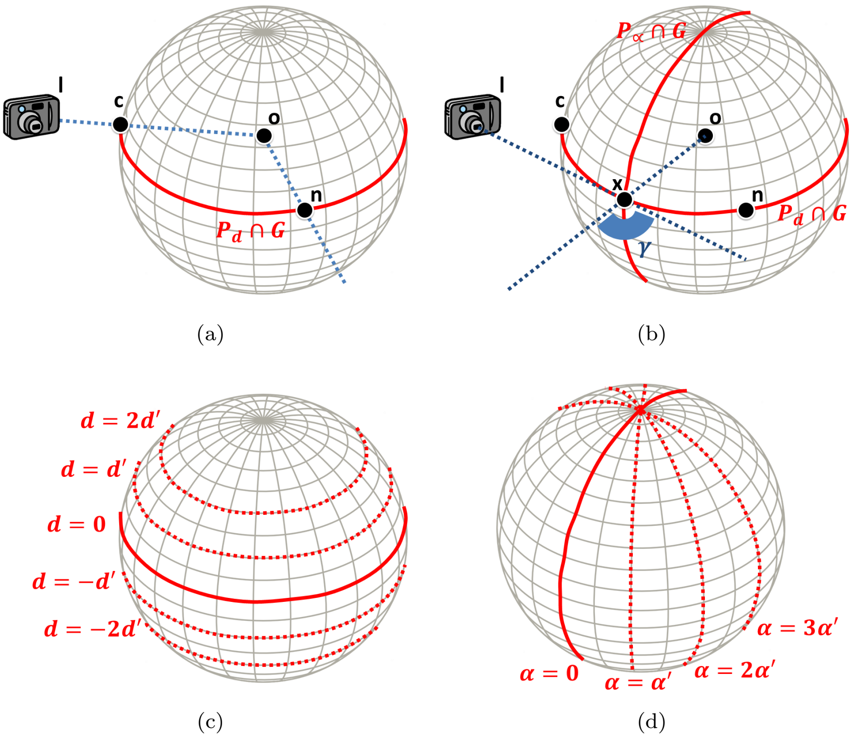
GeoPeels is summarized in Subsection 6.5.2. In Subsection 6.5.3, we describe how nodes are relocated if there are multiple *slices* as candidates. Finally, in Subsection 6.5.4, we describe the function to manage the surface deformation. The parameters mentioned in this section are described in details in Section 6.7.

6.5.1 Slice-Specific Spherical Reference System

The algorithm to check whether a node is inside a slice, to relocate a node, and to create a slice is based on a spherical reference system R_S, specific to a *slice* S. R_S is defined by the planes P_d and P_α.

The first plane, P_d, is identified by three points: the point that corresponds to the node n we want to reveal, the center of the earth o, and c that is the camera view point l projected over the globe, see Figure 6.5(a). For every point p on the globe, γ defines the angle between the vector lp and the normal of the globe surface on p. If $\gamma > \pi/2$ then the point p is visible, otherwise it is occluded by the globe. Thus, we define x as the point on the great circle, between c and n with angle $\gamma = (\pi/2) + 0.4$.

Consider a second plane, P_α, to be perpendicular to P_d and to intersect o and x, see Figure 6.5(b). Then, we have that the position of p is defined by $R_S(\alpha, d)$, where d is the perpendicular distance between p and P_d, and α is the angle between the plane P_α and the plane perpendicular to P_d, which intersects the points o and p. See, respectively, Figure 6.5(c) and Figure 6.5(d).



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

Figure 6.5: The planes P_α, P_d are used to define a distinct spherical reference system for each *slice*. α' and d' are values arbitrarily defined. Red lines are the intersections between the globe G and the planes.

6.5.2 GeoPeels

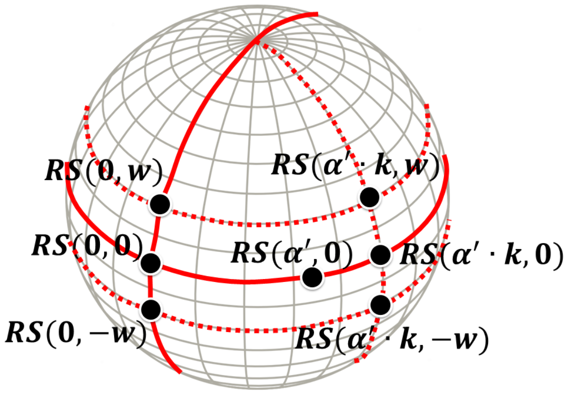
Our approach can be summarized with the following procedure, which is executed every frame before the rendering phase:

```
procedure GeoPeels
    (1) List< $Node$ > N := $getNodesToRelocate()$;
    List< $Slice$ > S := $\emptyset$;
    for each $n \in N$ do
        (3) bool hasSlice := $hasCompatibleSlice(n, S)$
        if $hasSlice = false$ then
            (2) Slice s := $createSlice(n)$
            $S.add(s)$
        end if
    end for
    for each $n \in N$ do
        (3) $Slice$ s := $getChosenCompatibleSlice(n, S)$
        (4) $relocateNode(n, s)$
    end for
    (5) $drawArcs()$
    for each $s \in S$ do
        (4) $deformSlice(s)$
    end for
end procedure
```

(1) Retrieve the list of important nodes to relocate. The list of important nodes to relocate, is computed by checking their angle γ; an important node, i.e. whose DOI function returns 1, is added to the list if $\gamma < (\pi/2) + 0.4$. The list considers the occluded nodes together with the nodes that are placed near to the occluded hemisphere. Afterwards, the list is sorted in descending order according to each node distance to the camera view projected over the globe. This operation is useful to minimize the number of *slices*.

(2) Slice creation procedure. Let $R_S(\alpha', 0)$ be the position of a primary node, and w and k the parameters to define respectively the width and the length of its *slice*. Then, the *slice* is simply the set of all the points located in $R_S(\alpha, d)$ such that $d > -w$, $d < w$, $\alpha > 0$, and $\alpha < \alpha_S$, where $\alpha_S = \alpha' \cdot k$, see Figure 6.6(a).

(3) Retrieve compatible slices for a given node. A *slice* is compatible with a node if the latter is inside the geographical area over the globe to be deformed, see Figure 6.6(b). If we have more than a compatible slice for a node, then we need to use a



(a)

(b)

Figure 6.6: (a) Points used to define the *slice*. (b) n_1 is the primary node of S. S is compatible for n_2 but not for n_3.

strategy to choose the best *slice*. Such strategy is described in details in Subsection 6.5.3.

(4) Deformation of a slice and relocation of a node. Each *deformed slice* is a grid of points where we map the texture of the globe surface that corresponds to the area of interest. Both, the nodes and the grid points are relocated over the globe and consequently in altitude using the relocation function, explained in details in Subsection 6.5.4.

(5) Arc creation procedure. Arcs are splines composed of a sequence of control points along the great circle that connects their two endpoints. Each control point has the following altitude:

$$h = 1 + h_{max} sin(\pi t) \tag{6.1}$$

The parameter t ranges from 0 to 1, along the arc path. h_{max} is equal to a quarter of the distance between the endpoints. Arcs that connect nodes that are both relocated, are composed of control points with the following altitudes:

$$h^* = h + (h_s + (h_e - h_s)t) \tag{6.2}$$

The parameters h_e and h_s are respectively the altitudes of the two endpoints e and s. The altitude is applied with respect to the normal vector of the globe surface, in the point along the great circle. With this procedure, also arcs that connect relocated nodes maintain an aesthetically pleasing shape.

6.5.3 Slice Selection Function

Deformed slices can occlude arcs, nodes and other slices. It is important to minimize the number of deformed slices as well as to ensure that each node is not occluded by its *com-*

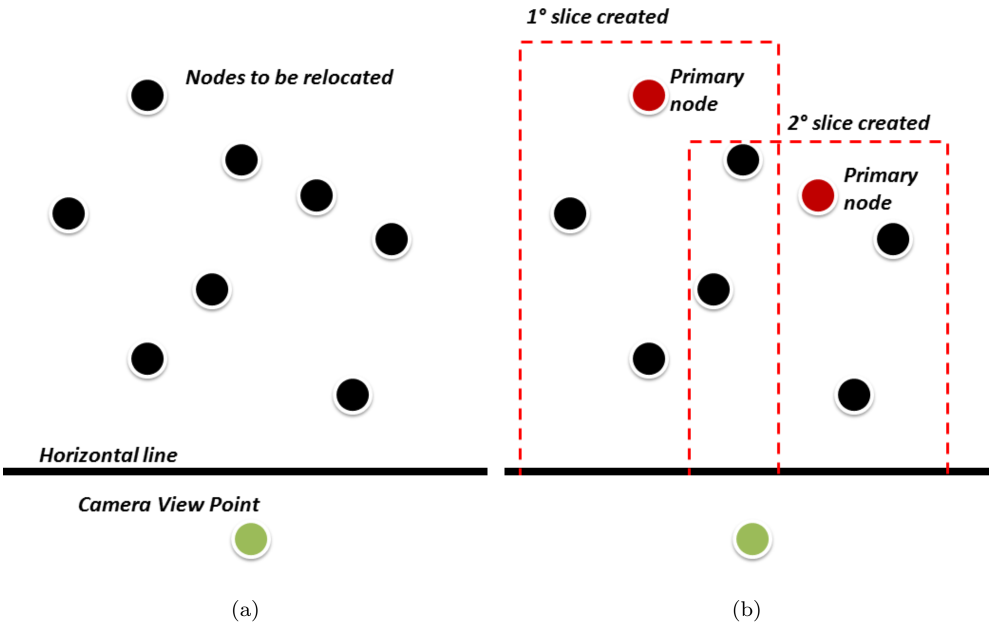


Figure 6.7: Our strategy to create slices and to choose which is the best one for each node.

patible slices. In this section, we describe the approach that satisfies these requirements.

Figure 6.7(a) shows our strategy in a projected 2D plane, to facilitate the explanation. The steps are the following:

The nodes are selected in descending order with respect to the distance with the camera view position projected over the globe. If the node has a compatible slice, then it is marked as *to relocate*. Otherwise, we create the *slice* and the node became its *primary node*, see red nodes in Figure 6.7(b). In a second round, we relocate the nodes. If a node has more than one compatible slice, then we choose the last created slice, because it is the most visible.

6.5.4 Function to Relocate Points

In this section, we describe how we relocate a point from $R_S(\alpha, d)$ to $R_S(\beta, d)$. Such point can be a node or a grid point of the deformed slice.

First, we defined the altitude of the point to relocate with the following formula:

$$h' = 1 + h'_{max} sin(\pi((\alpha - \beta)/2\alpha)) \tag{6.3}$$

where h'_{max} is equal to the great-circle distance between the original point and the point $R_S(0, d)$. Moreover, for a given point $R_S(\alpha, d)$ that is relocated to $R_S(\beta, d)$, the corresponding vector to move the point in altitude, is the normal vector of the surface globe at the point $R_S(\beta, 0)$, see Figure 6.8(a). Thus, the deformed slice maintains the same width for every curvature.

The angle β is calculated with the following formula:

$$\beta = -2\alpha f(v) + \alpha \tag{6.4}$$

where $f(v) \in [0, 1]$. The function $f(v)$ allow us to relocate points over a circular arc, see Figure 6.8(b). v is defined for each slice. If $f(v)$ is the identity function, the *deformed slices* will be created as shown in Figure 6.8(c). However, the nodes relocated on the upper area of the deformed slice may be occluded by the surface itself. For such reason, as shown in Figure 6.8(d), we update it as follows:

$$f(v) = v(\alpha/\alpha_S)^2 \tag{6.5}$$

The parameter v is initially set to 1. However, if the *primary node* is outside a defined *sub-region* of the screen view, v is decremented until the *primary node* is considered visible (i.e. inside the *sub-region*), see Figure 6.9.

6.6 Case Studies

In the following case studies the user wants to navigate through geographic node-link diagrams that cover large portions of the Earth. On the one hand, he/she wants to have an overview of the graph. On the other hand, he/she wants to focus on specific regions.

Internet Topology. We used GeoPeels to explore various Internet topology datasets[2] which are characterized of groups of nodes located in different sparse regions. Figure 6.10 shows some screenshots during an interactive exploration of different network topology datasets.

Airport connections. In this case study, we visualize a dataset describing the connections between the main airports[3], composed of 50 nodes and 677 edges. Due to the high number of nodes, during the navigation the user can select a subset of nodes to reveal. The dataset is characterized of groups of near nodes connected with long distance nodes. Hence, the circle element is appropriate to explore such dataset, see Figure 6.11(a). Due to the high arcs density, the links connected with the nodes in focus are colored in red and the remaining are colored in semi-transparent white, see Figure 6.11(b). With this combination, the *important nodes* together with their connected arcs can be easily identified.

In both use cases, the visualization enables the nodes and links to appear with limited distortion. When the user zoom in a specific area, the links express the real direction

[2] http://topology-zoo.org

[3] http://openflights.org

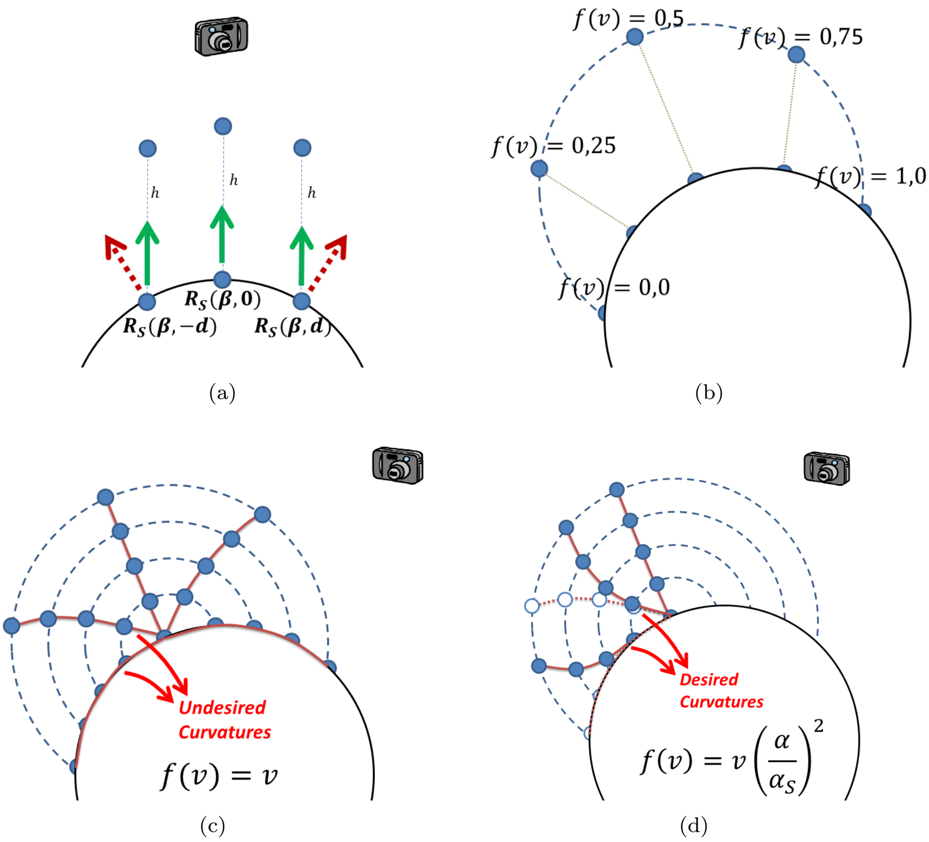


Figure 6.8: (a) Red arrows represent normal vectors over the globe surface. Green arrows represent the vectors used to move the points. (b) Each point is relocated over an imaginary arc. (c) The curvatures of a *deformed slice* modeled with the identity function, (d) and with our improved function.

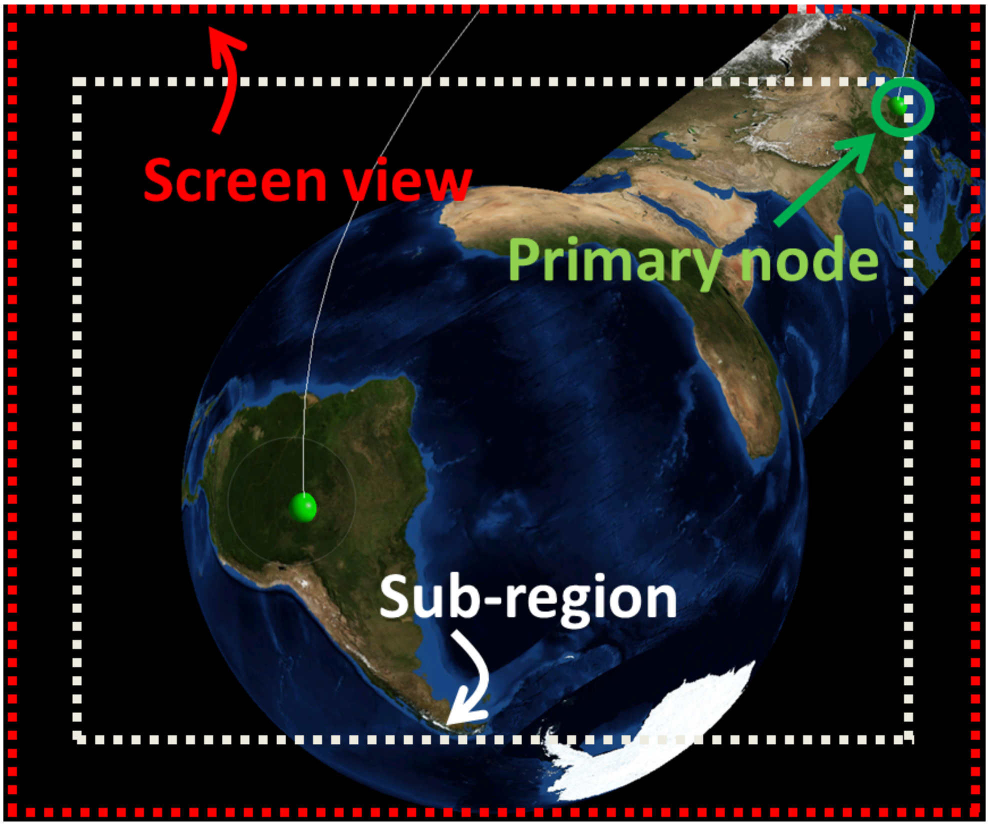
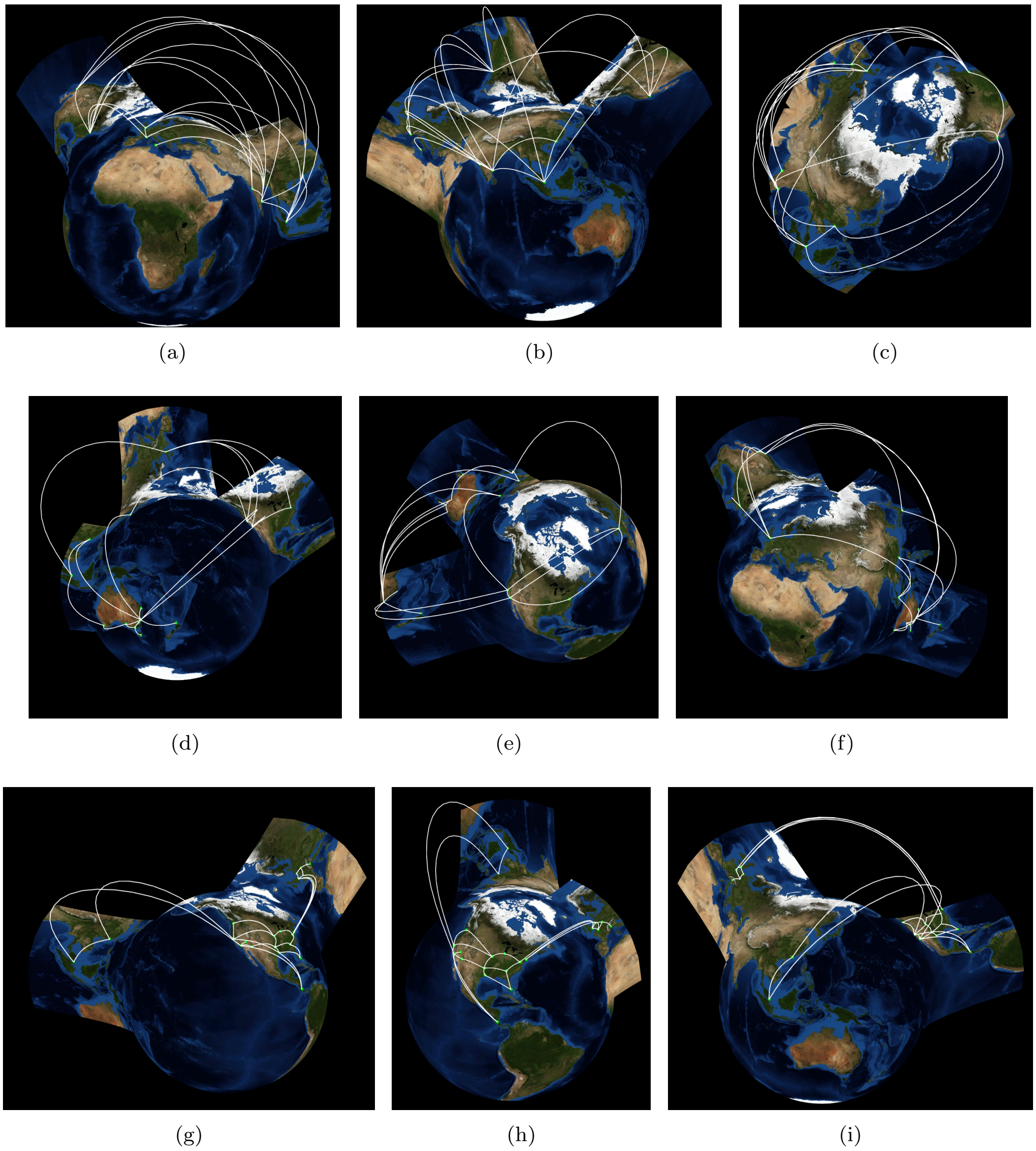


Figure 6.9: The *deformed slice* is bended until the primary node is inside the *sub-region* of the screen view.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 6.10: Geopeels used to explore various Internet topology datasets: Airtel dataset in (a)(b)(c), Internode dataset in (d)(e)(f), Packet Exchange in (g)(h)(i).

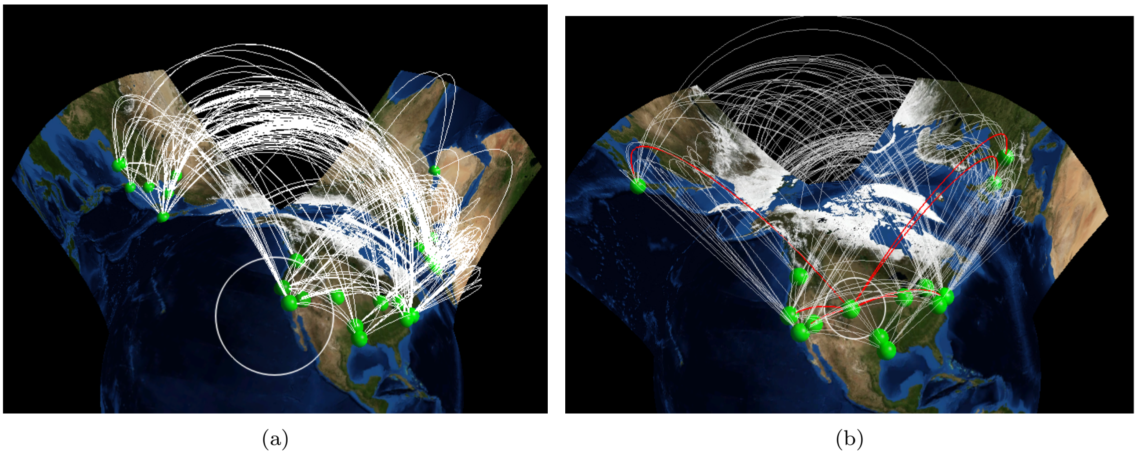


Figure 6.11: Airport connections dataset explored using *GeoPeels*. (a) Proximity based extension of GeoPeels to select a subset of nodes. (b) GeoPeels in combination with a color technique.

of the nodes located off-screen. The switch between overview and context is performed efficiently and with no discontinuities of features between frames.

6.7 Input Parameters

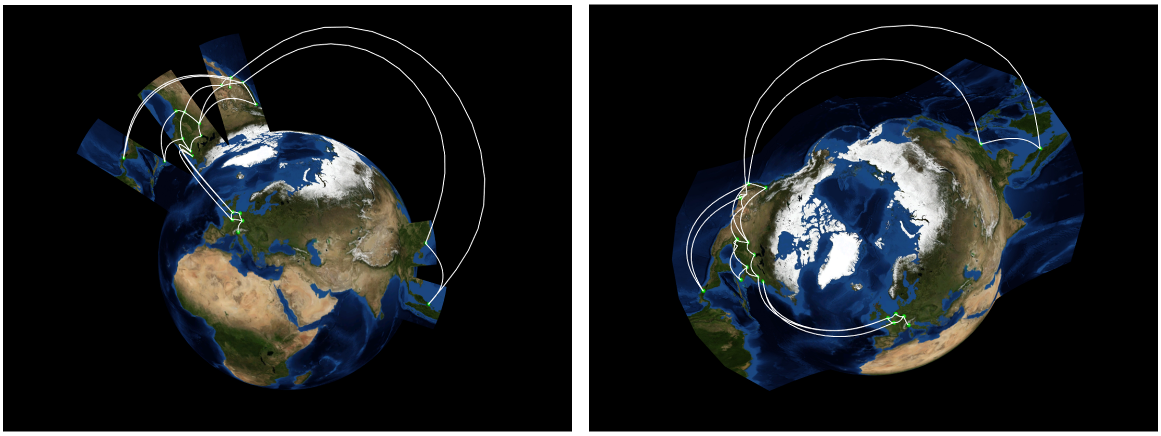
The list of user customizable parameters is the following.

1. X and Y are used to define the number of points that compose the grid of a *slice*. We do not need a dense grid to give the impression of a curved surface. Hence, heuristically we set $X, Y = 7$.

2. w defines the half width of the slice. The default value is 3.500 km that is a good balance between area to visualize and the possibility to intersect other slices. Figure 6.12 shows examples of other slices' width.

3. k defines the position of the relocated node with respect to the y-axis of the deformed slice. The default value is 1.2. It allows the near surface of the node to be visible. However, if the area is too circumscribed, the cognitive process to identify the location of the node can be problematic for the operator.

4. The dimension of the *sub-region* defines whether a node is considered visible or not. We set such parameter to be equal to 200 *pixels* less than the height and the width of the screen view. Since our strategy to decide the visibility of the nodes that are outside the screen view takes into account only the primary node of the *deformed slice*, other solutions can be adopted. For example, the visibility check can



(a) w=1.000 km

(b) w=5000 km

Figure 6.12: Examples with different slices' width.

be performed on the two upper vertices of the geometry, and not on the *primary node*.

5. For the extension that interactively selects the important nodes, r is the radius of the circle, expressed in pixels. We recommend the use of a small radius if the dataset contains many nodes in the circumscribed areas, in order to reduce the number of nodes that have to be relocated. The default value is 150 *pixels*, which can be interactively changed by the user.

6.8 Graphical Performance

To measure the performance of our technique we take as metric the frame rate. *GeoPeels* was tested on a single processor Intel Xeon 2.26 GHz equipped with an NVIDIA GeForce GTX 280 with 1024 MB dedicated video memory.

Table 6.2 shows the *fps* (frame rate per second) of *GeoPeels* taking into account two datasets respectively with 154 and 4777 nodes. *#def. nodes* indicates the number of nodes that are relocated and *#slices* indicates the number of slices that are created. In this benchmark we used the values defined in Subsection 6.7.

Our results take into account, for each frame, the time required to generate the grids and to render the geometries. In our tests, we used spheres over the terrain to represent the nodes.

After analyzing the evaluation results, we noticed that the number of created slices and the number of relocated nodes have a considerable impact on the number of *fps*. However, for datasets with hundreds of nodes our approach gives acceptable performance.

Table 6.2: Graphical performance of *GeoPeels* for different datasets.

#nodes	#def. nodes	#slices	fps
154	0	0	120
	1	1	54
	4	1	50
	11	2	34
	9	3	28
	11	3	26
	7	4	23
4777	0	0	16
	2	1	14
	1	1	13
	6	2	12
	9	3	11
	3	3	11
	15	2	10
	22	5	8

6.9 Discussion

In this section, we discuss our technique with respect to the requirements defined in Section 6.3, summarizing our findings and the overall strength and weakness of our technique.

Geo-context awareness and surrounding context preserved. GeoPeels extends the virtual globe, preserving the context of the nodes located on the visible hemisphere. As opposite, the context of the nodes relocated around the globe, is maintained using the concept of slice. However, the latter has different aspects that require a further investigation. In particular, the distortion and the orientation of the deformed regions could confuse the user. The deformed regions appear to be inverted respect to the visible hemisphere of the globe. Moreover, the distortion applied to the regions depends on the function used to bend the slices. Hence, a user study is needed to measure the effectiveness of our solution. To improve the identification of the deformed area, we can relocate together with the nodes auxiliary information, e.g. map labels.

Topology awareness. This requirement is guaranteed if the globe is inside the screen view. As opposite, if the camera view is zoomed on an specific region, i.e. with no unused space available, only the nodes and edges in focus will be visible.

Spatial continuity. This requirement is achieved because the approach is based on terrain deformations. On the other hand, there are limit cases in which the spatial continuity of links is not preserved: if portions of links leave the screen space, or if they pass behind a slice. We solved the latter case, rendering the arcs always on top of the slices. However, we did not apply such strategy in the extension that reveal only a subset of nodes, as shown in Figure 6.11.

Moreover, two slices may share a portion of the surface, causing redundant information in the visualization. This aspect is accentuated if the camera view is tilted, see Figure 6.10(b)(c)(h).

Temporal continuity. The continuity of node, links and deformed slices among frames is always preserved. The real-time navigation was proved by the performance test described in Section 6.8.

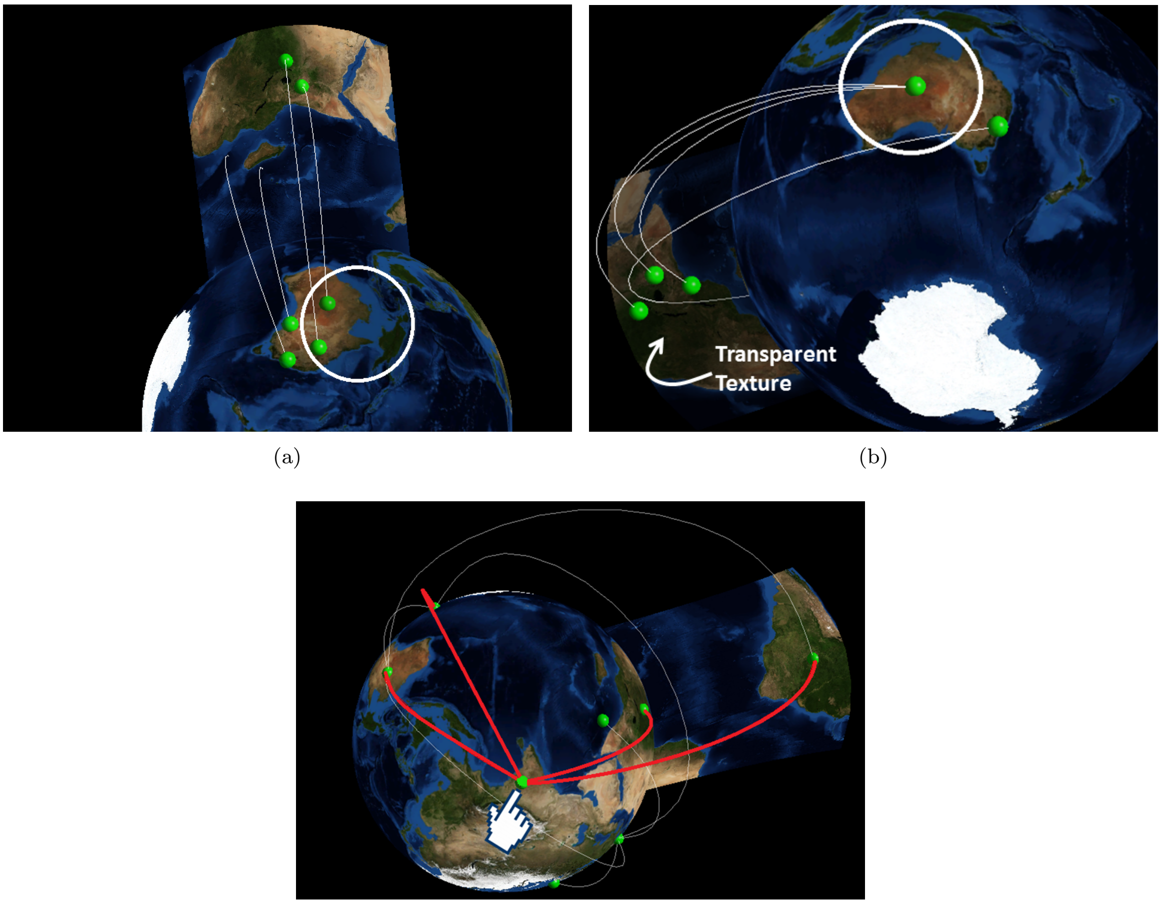
Direction awareness. Exactly as in the globe, the notion of the real direction of nodes of interest located off-screen is expressed through the links themselves. However, this requirement is not always valid for the nodes located on a slice. In that case, the user has to move the camera on that area of interest.

In a standard virtual globe, we can look at the arcs shape to gain insight on the distance between its endpoints - farther the endpoints are, higher the arc is. In our approach, this property does not hold in general.

Proximity based extension of GeoPeels. Regarding the extension used to select the nodes of interest, we identified few elements that require further investigation. First, some visual elements can be misleading for the operator because only nodes connected to nodes in the focus area are physically moved.

Second, it might happen that arcs connecting visible nodes and nodes occluded by the globe's hemisphere intersect the newly deformed surfaces, e.g. when an occluded node is located behind the surface, see Figure 6.13(a). Hence, it can mislead the operator to think that arcs end in the new deformed surface. However, this aspect can be alleviated by applying a level of transparency on the texture of each deformed geometry, see Figure 6.13(b), or applying a different color to the arcs that end over the deformed surface, see Figure 6.11(b).

Finally, this extension of *GeoPeels* requires the user to take care of the circle object. This aspect can affect negatively the performance for exploration tasks. *GeoPeels* can be redesigned in terms of interaction to improve its effectiveness. For example, a selection of a single node *in focus* can be introduced as an alternative to the proximity-based selection of groups of nodes, see Figure 6.13(d).



(a)

(b)

(c)

Figure 6.13: (a) Proximity based extension of GeoPeels to select a subset of nodes. (b) A transparent deformed slice, to keep trace of its intersecting arcs. (c) An alternative extension to select a single node.

6.10 Conclusions and Future Work

This work faces challenges that characterize only visualizations for global datasets, such as the misleading directions of links partially visible or their spatial continuity in the geographic space.

We described an interactive deformation-based technique, used in combination to zoom and pan, that capitalizes on dynamic globe deformations for an effective exploratory visual analysis of geographic networks. In particular, it solves the occlusion issue that characterizes the visual globe and alleviates the issue of geometries rendered outside the screen view.

The evaluation of our approach is based on the analysis of the graphical performance, on a comparison between our technique with existing approaches, and testing it within some case studies.

Despite the occlusion issue, in literature, many approaches have been proposed to improve the globe visualization during network analysis. Brandes et al. [19] minimized clutter at important nodes by adjusting the angle of their connecting links. A few techniques [102, 41] were provided to bundle edges around the globe. Alper et al. [6] presented a visual technique to better understand the relations of the network; the globe is deformed to place the nodes connected with each other to appear closer. In Chapter 5, we presented *3DArcLens*, a focus+context technique that can be applied on globes to reveal areas occluded by undesired edges.

GeoPeels does not require the operator to define visual node-link attributes, e.g. we do not use attributes like color and size of a node, hence, they can be used to encode additional information. Consequently, our technique can be combined with other works that make use of such visual attributes, for example, to reduce the visual clutter or combined with focus+context techniques applied on virtual globes.

As future work, we plan to empirically measure the performance of the presented technique on a more comprehensive set of tasks. We also want to extend the orange peeling metaphor to other contexts such as the visualization of points of interest represented as icons and 3D models, or even to more complex scenarios such as the visualization of moving objects in real-time.

Chapter 7

Supervised Force-Directed Algorithm for the Generation of Flow Maps

Much change in the world is due to geographical movement.
–Waldo R. Tobler

7.1 Overview

Cartographic flow maps are graphical representations for portraying the movement of objects, such as people, goods or traffic network, from one location to another. As mentioned in Chapter 2, on the one hand, flow maps can reduce visual clutter by merging single representations of movement. On the other hand, flow maps are also fast to produce and simple to understand.

In this chapter, we present a novel method for the automatic generation of flow maps. Our method is based on a theoretically grounded physical system to describe the motion and forces of attraction and repulsion between the data points that compose the flow lines. It allows the user to supervise the graph during the creation process of the flow map. We evaluated our algorithm discussing its computational cost and providing a comparison with existing implementations. Results have shown a good compromise between the computational complexity of the algorithm and the visual quality of the generated maps. Part of the work in this chapter was previously published in [39, 41][1].

7.2 Introduction

Flow maps depict the movement of phenomena between geographic locations [151]. Phenomena can represent the movement in geographical space of both tangible (e.g. people,

[1] As additional material, a video showing a demo of the presented technique was included in the related published papers. It is available online at `https://www.youtube.com/watch?v=ajsMVqS7-yA&list=PLlB_rfTJJP4S6ErhbjToH0JHe89wc5WZg&index=2`.

bank notes, and goods) and intangible objects (e.g. energy, ideas, and reputation). The links in a flow map, called flow lines, describe the movement of objects from one location to another. The way flow lines are aggregated and depicted is what makes a given flow map algorithm unique.

The Sankey flow drawing technique [146] is an example of method to aggregate flow lines. It describes the thickness of the aggregated flow lines to be proportional to the sum of the flows' magnitude they represent. Although the idea of defining the thickness to be proportional to the magnitude was intensively used by Matthew Sankey in 1898, it is known that he was not first to use it.

In this chapter, we describe an automatic technique for the generation of natural and high visual quality flow maps using a force-directed algorithm. On the one hand, it provides an alternative to already available tools for the generation of flow maps that shift the modelling effort to the user side [1]. On the other hand, our algorithm gives also the possibility of supervising the layout of the flow map during the creation process. Our algorithm not only shows a good computational complexity, but it also satisfies the following aesthetic criteria characterizing well drawn flow maps [129, 174]:

C1. The use of smooth curves for aesthetic purposes;

C2. The possibility of aggregate flows, reducing the visual clutter;

C3. Put emphasis on the main branches of the flow tree; straight lines are correlated to target destinations with high magnitude;

C4. The target destinations represented by geometrical features such as circles are not overlapped with flow;

C5. Each flow line is crossing-free.

This chapter is structured as follows. Section 7.3 explains the proposed algorithm and covers all the implementation details. Section 7.4 presents the benchmarks based on a test case and explores the meaning of this study in terms of visual appearance and performance. Section 7.5 wraps up possible extensibility.

7.3 System Design

In this section, we describe the four components that characterise our application (see Figure 7.1). The first layer, entitled *Basic Structure Generation*, is responsible for the generation of the flow tree. Once the tree structure is generated, a second layer - *Flow Graph Layout* - is responsible for applying interaction forces to nodes and then for updating the tree structure accordingly. As described later, this phase is executed for a number of iterations. Then, the third layer - *Rendering* - depicts the flow map. Finally, the layer

Interaction is designed to give users the possibility to modify the output by means of simple user's actions.

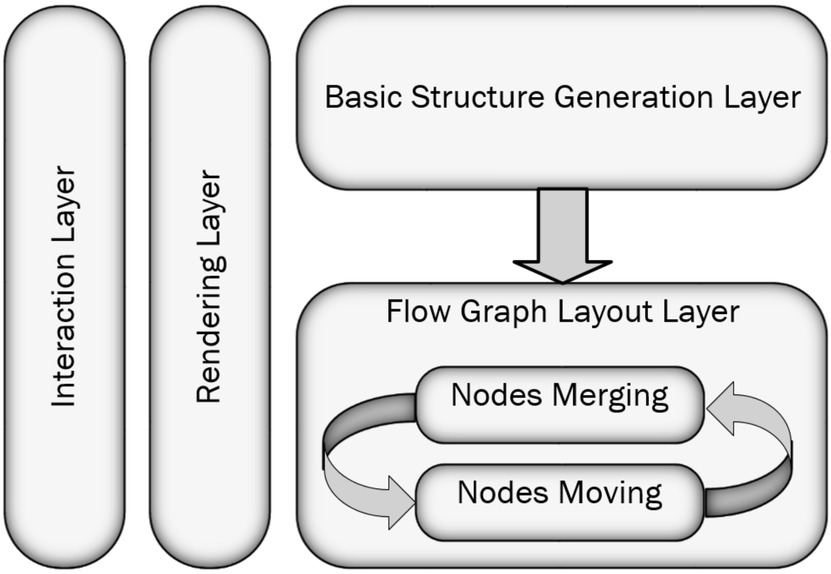


Figure 7.1: Overall system diagram.

The algorithm receives as an input the geographic position of the origin, that is, the root node r of the tree t and n geographical destinations corresponding to the leaves $l_1, l_2, \ldots, l_n$, respectively with flow magnitudes $mgn(l_1)$, $mgn(l_2)$, $\ldots$, $mgn(l_n)$.

7.3.1 Basic Structure Generation Layer

Initially each flow line, i.e. the line that connects the root r with a leaf node l_i, is a straight line. The first step is to divide each flow line in sub-segments of length d composed by the intermediate nodes $m_{i,j}$. The index i identifies the leaf node l_i of the original segment. This index takes into account the position of the leaves in clockwise order. The sorting operation required to obtain such indexes has a computational cost of $O(n \cdot \log(n))$. The index j identifies the order of the intermediate nodes associated to l_i. For example, if the distance from the root r to the leaf l_1 is equal to d, there will be the intermediate nodes $m_{1,1}, m_{2,1}, ..., m_{n,1}$. If the distance to the leaf l_2 is $2d$, the set of nodes is $m_{1,2}, m_{2,2}, ..., m_{2n,2}$, and so on (see Figure 7.2). Therefore, the number of intermediate nodes $m_{i,j}$ is proportional to the distance between the leaf l_i and its root r. Each intermediate node $m_{i,j}$ is tagged with the following information:

- the parent node; initialised as $pred(m_{i,j}) = m_{i,j-1}$.
- the child nodes; where $next(m_{i,j}) = \{m_{i,j+1}\}$ is the initial set.

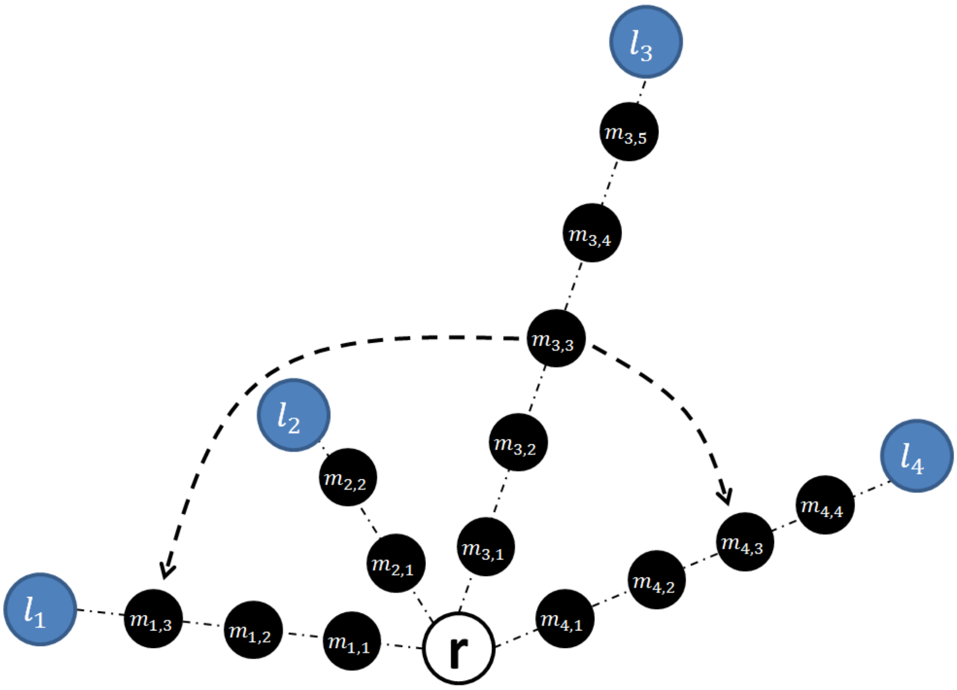


Figure 7.2: Node $m_{3,3}$ is tagged with the following information: $mgn(m_{3,3}) = mgn(l_3)$, $pred(m_{3,3}) = m_{3,2}$, $next(m_{3,3}) = \{m_{3,4}\}$ and $int(m_{3,3}) = \{m_{1,3}, m_{4,3}\}$.

- the interacting nodes; a set containing the surrounding nodes. $int(m_{i,j}) = \{m_{i+1,j}, m_{i-1,j}\}$ is the initial set. Leaves cannot be interacting nodes.

- the magnitude: $mgn(m_{i,j}) = mgn(l_i)$.

When a node $m_{i,j}$ does not have $m_{i-1,j}$ as interacting node (due to the fact that the distance from $m_{i,j}$ to r is greater than the distance from l_{i-1} to r), it will be substituted with the first of the previous ones, as shown in Figure 7.2. The same applies to the absence of the node $m_{i+1,j}$. Consequences are explained in Section 7.4.1.

7.3.2 Flow Graph Layout Layer

This layer aims at computing the flow graph layout. Figure 7.3 illustrates the evolution of a tree structure during the execution of the algorithm. The two steps that define this layer are executed in sequence, and for a number of iterations.

Nodes merging phase. This phase was designed to update the structure of the tree t, which becomes obsolete after the merging procedure. The following pseudo-code describes the merging steps:

```
for each intermediate node $m_{i,j}$ do
    for each $s \in int(m_{i,j})$ do
```

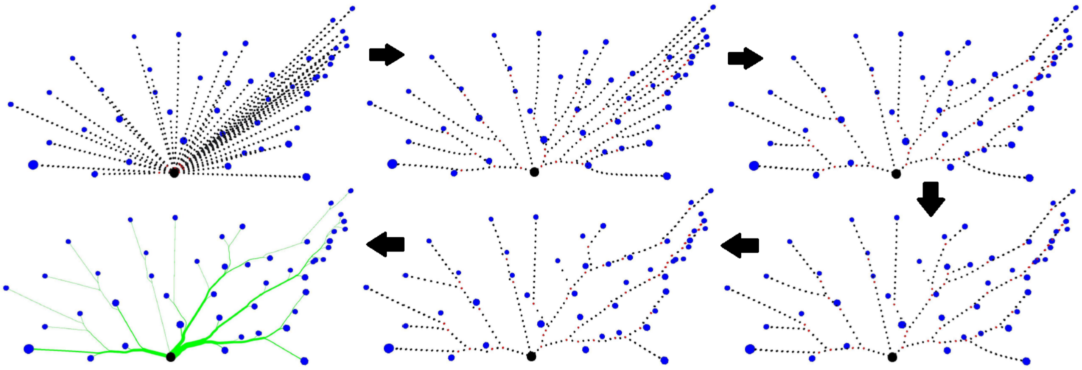


Figure 7.3: Tree structure during the generation of the flow map.

```
        if $samePosition(m_{i,j}, s)$ & $Pred(m_{i,j}) \equiv Pred(s)$ then
            $createNode(s, m_{i,j})$
            $removeNode(s)$
            $removeNode(m_{i,j})$
        end if
    end for
end for
```

The function *samePosition* requires a threshold d_s to define whenever two positions are equivalent. The existence of a common parent for s and $m_{i,j}$ is a necessary condition to maintain the structure as a tree and not only as a directed acyclic graph. The function *createNode* inserts a new node. The new node shares the parent node of s and $m_{i,j}$; the child nodes of s and $m_{i,j}$; has as interaction nodes the interaction nodes of s and $m_{i,j}$; has as magnitude the sum of the magnitudes of s and $m_{i,j}$; and has as position the middle position between s and $m_{i,j}$. The function $removeNode(k)$ deletes the intermediate node k from the system. The computational complexity of this phase is $O(m)$ in each iteration.

Nodes moving phase. This phase aims at readjusting the position of intermediate nodes. The behavior of each intermediate node depends on two forces: the attractive and the stress force. The attractive force is calculated only between the node and its interactive nodes, within a certain distance d_a. The attractive force F_a, is given by the equation:

$$F_a(m_{i,j}) = \sum_{s \in Int_{m_{i,j}}} \frac{mgn(s)}{mgn(s) + mgn(m_{i,j})} \frac{1}{\|m_{i,j} - s\|} \widehat{(m_{i,j} - s)} \tag{7.1}$$

The notation $\hat{A}$ and $\|A\|$ gives respectively the unit vector and the norm of A. The factor $\widehat{(m_{i,j} - s)}$ gives the direction of the force vector and $\frac{1}{\|m_{i,j}-s\|}$ the force length; the closer the two nodes are, the higher is the force between them. $mgn(s)$ and $mgn(m_{i,j})$

are respectively the magnitude of s and $m_{i,j}$. Without the factor $\frac{mgn(s)}{mgn(s)+mgn(m_{i,j})}$ the formula takes into account the distance of the interacting nodes but not their magnitude. According to the criterion C3, we want nodes of smaller magnitudes to be attracted by nodes of higher magnitude. Moreover, we want flows with higher magnitude to have straighter shapes than the ones with lower magnitude.

In addition to the attractive force, a stress force F_s is applied to each intermediate node, to keep a 'middle-aligned' position from the parent node, as well as from the child nodes.

$$F_s(m_{i,j}) = (prev(m_{i,j}) - m_{i,j}) + \sum_{s \in next_{m_{i,j}}} \frac{mgn(s)}{mgn(m_{i,j})}(m_{i,j} - s) \tag{7.2}$$

The condition defined in Equation 7.2 ensures that nodes move always towards children of higher magnitude. $mgn(m_{i,j})$ is the magnitude of the current node, that is, the sum of the magnitude of all children. $mgn(s)$ is the magnitude of a child node. In order to avoid oscillations, the stress force is applied only if the force is greater than a threshold t. The final formula to compute the force corresponds to the sum of the stress and the attractive forces:

$$F_{final}(m_{i,j}) = F_a(m_{i,j}) + k_s \cdot F_s(m_{i,j}) \tag{7.3}$$

where the constant k_s is the oscillation that defines the stress force. In this context, the concept of force is the one of displacement applied to a node.

The total force of the system is equal to the scalar sum of all forces applied to the intermediate nodes. After each iteration the total force decreases, firstly because the number of nodes is reduced, and then because the interacting nodes become out of range (i.e. when the distance from their associated node is greater than d_a).

The tree structure is marked as ready once a stable total force is reached. Unfortunately, the tree structure does not take into consideration the overlap of nodes, one of the criteria that this algorithm aims to achieve. Hence, the objective of the next step is to apply a repulsive force F_r to intermediate nodes within a certain distance d_r from the leaf nodes. The displacement is applied only at the end because otherwise it would affect the evolution of the tree t.

$$F_r(m_{i,j}) = \sum_{l \in leaves} \frac{-1}{\|m_{i,j} - l\|}(m_{i,j}\hat{} - l) \tag{7.4}$$

The repulsive force is equal to the opposite of the attractive force without the factor that takes into account the magnitudes. Additionally, we add the stress force to guarantee that the position of children nodes is updated properly once the node location changes (see Figure 7.4).

In pseudo-code the operations of this phase are described as follows:

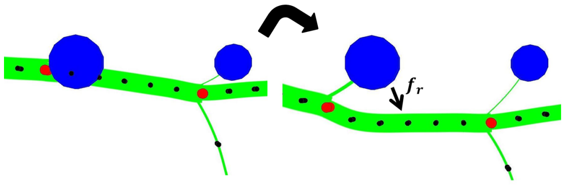


Figure 7.4: (left) Overlapping between a flow line and a leaf node. (right) Repulsive and stress forces are applied to avoid overlapping.

```
for each intermediate node $m_{i,j}$ do
    if $SystemIsStable == FALSE$ then
        $displ(m_{i,j}) = F_{final}(m_{i,j})$
    else
        $displ(m_{i,j}) = F_r(m_{i,j}) + k_s \cdot F_s(m_{i,j})$
    end if
end for
for each intermediate node $m_{i,j}$ do
    $position(m_{i,j}) += displ(m_{i,j})$
end for
$SystemIsStable = checkStability()$
```

In summary, the execution of the algorithm runs in two steps. First, Equation 7.3 is applied until the total force of the system is stable. Then, the attractive force is substituted with the repulsive force to avoid overlapping with destination nodes. When the system becomes stable for the second time, the algorithm stops. At the end, the intermediate nodes' position is updated.

7.3.3 Rendering Layer

The classes responsible for rendering the maps on the screen are defined in the Rendering layer. The 3D rendering library used is JOGL, a binding of OpenGL for Java, which has been released by Oracle for Windows, Solaris, Linux and Mac OS platforms. Maps can be visualised both in 3D and 2D.

For aesthetic purposes, curves were preferred over straight lines during the rendering phase [9] (Criterion C1). Xu et al. published a study to empirically evaluate their effectiveness on common graph-related tasks [189]. In our implementation, we used natural Cubic Splines for the depiction of the flow lines. We decided for this kind of curve representation because it interpolates the intermediate nodes and its design dynamics are intuitive. The line width is proportional to the magnitude of the intermediate nodes that compose the flow. Moreover, for each split the starting point of the child's spline is shifted by a distance proportional to their width, in a direction perpendicular to the edge' vector (see Figure 7.5).

Additionally, leaf nodes are represented as circles whose size is proportional to their magnitude. It is possible to visualise the tree structure as well; intermediate nodes are depicted in red if they have more than one child node and black otherwise.

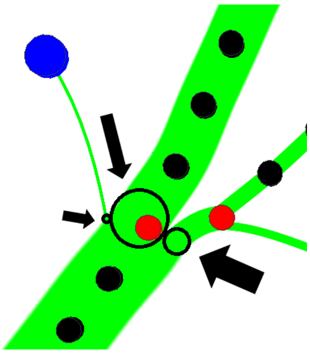


Figure 7.5: Splits of flow lines. Unfilled circles shows the starting point of the child's spline.

7.3.4 Interaction Layer

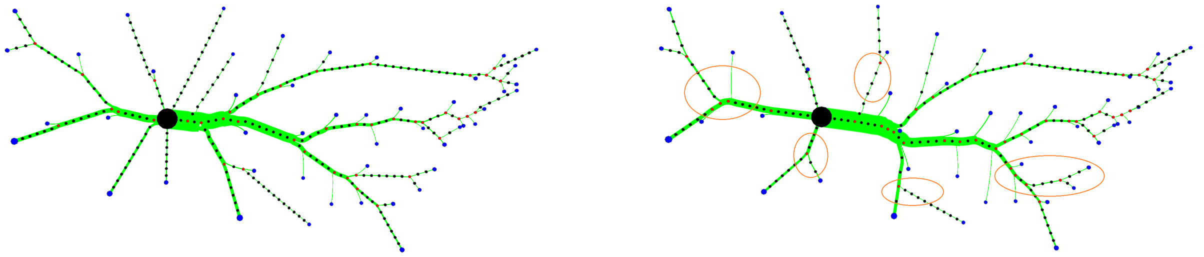
The interaction layer enables the interaction with the structure of the flow graph tree to improve its visual quality. At the current state of the research, automatic methods for the generation of flow maps do not support supervised layouts before concluding the generation process. Manual updates are possible once the layout is generated.

The interaction layer supports two scenarios: the automatic generation of a flow map and the supervised step-by-step layout generation. The latter option gives users additional control over the final layout. Hence, a given area can be progressively refined, for example, by manually moving certain intermediate nodes to a different location. After applied the translation force, the system takes it into account and recomputes the new flow graph tree.

The user interface has three input components to control the generation process. A button *next* to compute the *node merging phase* and the *node moving phase* for the next step. Since the number of iterations can be in the order of ∼ 1000, the user has also the possibility to run it in batches. At any moment, the user has the possibility to manually modify the attractive, stress and repulsive forces and decide when its time to stop the algorithm execution.

As described in Section 7.3.2, the algorithm is responsible for the management of all the forces involved in the system.

The use of the supervised mode is useful when a flow map is composed by a large number of branches. In such scenario, the user can 'drag & drop' intermediate nodes with the goal of activating new attractive forces to reduce the number of branches of the tree structure. The Figure 7.6 shows a geographical dataset depicted with and without the user supervision.



(a) Automatically generated flow map.

(b) Flow map generated with the supervision of the user.

Figure 7.6: Flow maps illustrating the migration from California between 1995-2000. Intermediate nodes are visible only during the supervised mode. The ovals identify the areas where the user requested the aggregation of flows.

7.4 Evaluation of the Algorithm

In this section, we compare our algorithm to existing implementations. We also provide an evaluation of its computational cost. Results have shown a good balance between computational complexity and the visual quality of the generated maps.

7.4.1 Comparison with existing algorithms

Table 7.1 provides a description of the most relevant methods for the automatic generation of flow maps according to the aesthetic criteria defined in Section 7.2. The Y/N letters stand for Yes/No and the properties marked with a star have limitations that are also explained.

Methods	C1/C2	C3	C4	C5	Time Complexity
Flow Map Layout	Y	Y*	N	N	$O(n^2)$
Flow Map Layout via Spiral Tree	Y	Y*	Y	Y	$O(n \cdot log(n)) + O(i \cdot m \cdot n)$
Confluent Spiral Drawing	Y	N*	Y	Y	$O(n \cdot log(n)) + O(k \cdot n \cdot \triangle) + O(s \cdot log(n + s\triangle))$
Force Directed Flow Map Layout	Y	Y	Y*	Y*	$O(n \cdot log(n)) + O(i \cdot m \cdot n)$

Table 7.1: Table for the comparison of the algorithms for automatically generated flow maps.

C1. All methods evaluated in this section use smooth lines. The flow map layout algorithm [129] uses Catmul-Rom splines, Flow Map Layout via Spiral Tree [174] uses cubic Hermite splines and Confluent Spiral Drawing [123] uses logarithmic spirals. Our algorithm uses a natural Cubic Spline to represent flow lines.

C2. The first two methods compute the aggregation of flow lines through the creation of binary splits. Unlike the first, the second method can recursively perform binary splits that are then merged if the distance meets a certain threshold. Confluent Spiral Drawing and our algorithm do not impose such limitation.

C3. The criterion states that the flow magnitude should affect the layout of the generated map. However, such condition is not completely satisfied by the first three algorithms. In the first two algorithms only the position of the intermediate nodes is affected meanwhile the structure of the tree remain unmodified. The authors justify such restriction with the claim that it helps the comparison across different time periods. In the third algorithm the magnitude is not taken into consideration. Although the authors claimed the possibility to use other attributes instead of only the distance from the root node, their paper not goes into details describing this aspect. Nevertheless, it can be useful to have a dynamic tree typology to give emphasis to flow lines with greater flow magnitude.

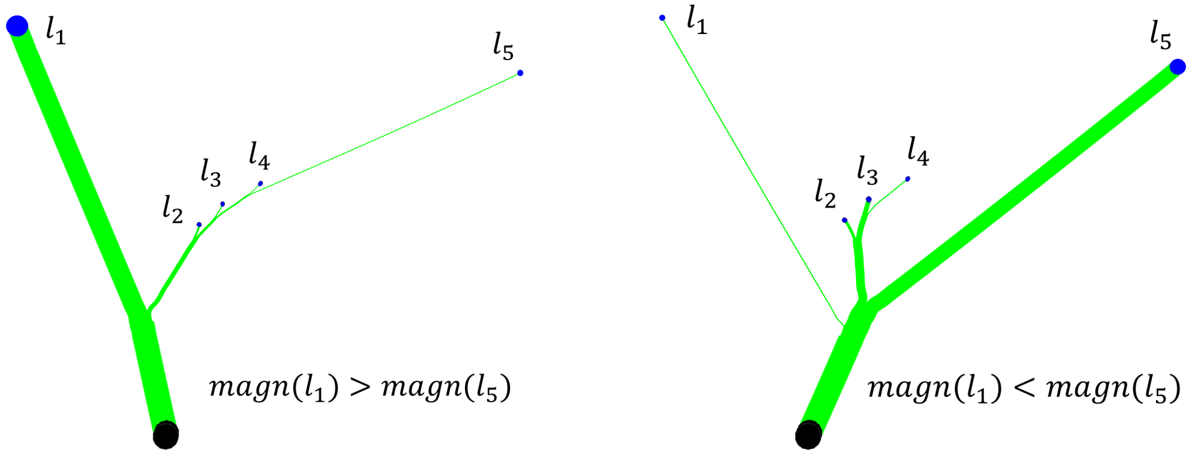


Figure 7.7: Representation of a flow maps with different magnitudes

As showed in Figure 7.7, our method can depict a flow tree with a structure specifically designed to account for the magnitude of each destination node, that is, it takes into consideration the magnitude factor in the force behaviours.

C4. The leaf nodes are not overlapped for all the methods except the first; a buffer is used to distance the flow lines from the nodes. In our algorithm, if the number of intermediate nodes is not sufficient some overlapping can occur - i.e if the d_r is less than

$d/2$.

C5. In our algorithm the overlap of flow lines depends on the criteria used to define the interacting nodes. If the rule is to assign for each $m_{i,j}$ the nodes $m_{i-1,j}$ and $m_{i+1,j}$ (if they exist) as interacting nodes, the tree is crossing-free. On the other hand, if we extend the rule as described in Section 7.3.1 (for example the node $m_{1,3} \in int(m_{3,3})$ in Figure 7.2), the overlapping is permitted but the quality of the generated output is improved. In Figure 7.8 is possible to assess the benefits of this decision. However, by increasing the number of intermediate nodes we can reduce considerably the possibility of such scenario.

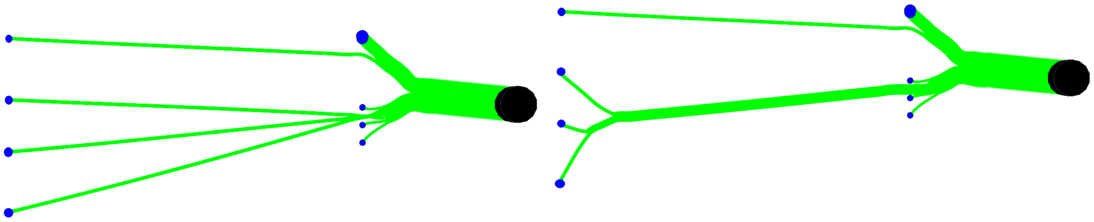
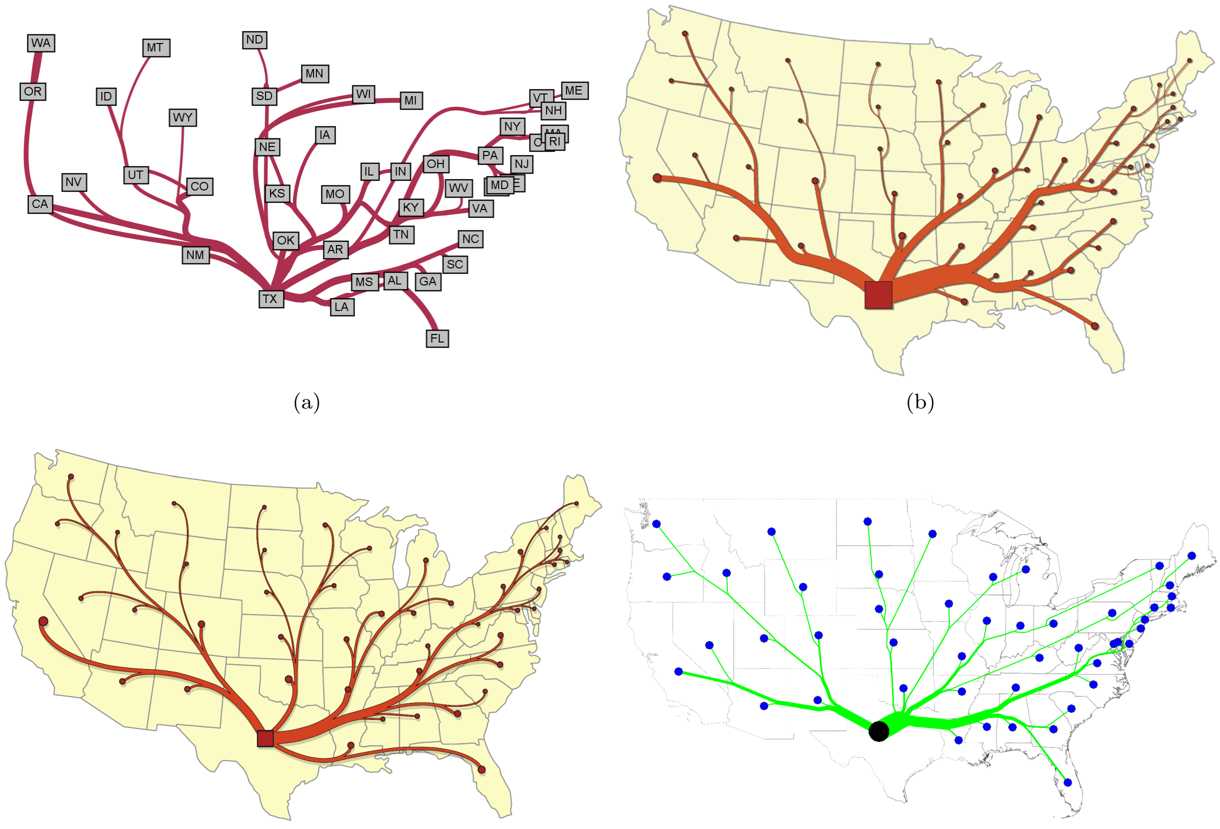


Figure 7.8: This example highlights a weak case in terms of visual quality and how it is solved. A flow map where the absence of overlapping is guaranteed (left), flow map with the possibility of overlapping but with better visual quality (right).

Figure 7.9 shows the comparison of flow maps depicted with Flow Map Layout, Flow Map Layout via Spiral Tree, Confluent Spiral Drawing, as well as with our algorithm. The output of the first algorithm contains many crossings, the grouping of nodes is somewhat unnatural, and the edges are often difficult to follow. The other outputs can be considered aesthetically pleasing, although no methodology exists to evaluate flow maps generated automatically.

7.4.2 Evaluation of the computational complexity and visual quality

To decide the number of intermediate nodes to generate, we need to set the distance d between the intermediate nodes. We define the parameter f_n to be the maximum number of intermediate nodes in a flow line. Then we assign $d = d_{max}/(f_n + 1)$, where d_{max} is the length of the longest flow line. This parameter consequently affects the performance and visual quality of the flow map. Figure 7.10 compares the number of nodes, time and iterations for a flow map that depicts the top 30 exports of Whisky from the UK. Let $f_n = 30$ and $m = 338$ be the initial number of intermediate nodes, then the algorithm stops after 2,51 seconds and 1597 iterations. With $f_n = 60$ and $m = 695$, the algorithm halts after 9,46 seconds and 2487 iterations. For $f_n = 100$ and $m = 1177$, the algorithm stops after 23,1 seconds with a total of 4240 iterations. Note that a small number of



(a)

(b)

(c)

(d)

Figure 7.9: Maps illustrating migration from Texas 1995-2000. The output of [129] (a), the output of [174] (b), the output of [123] (c) and the output of our algorithm (d).

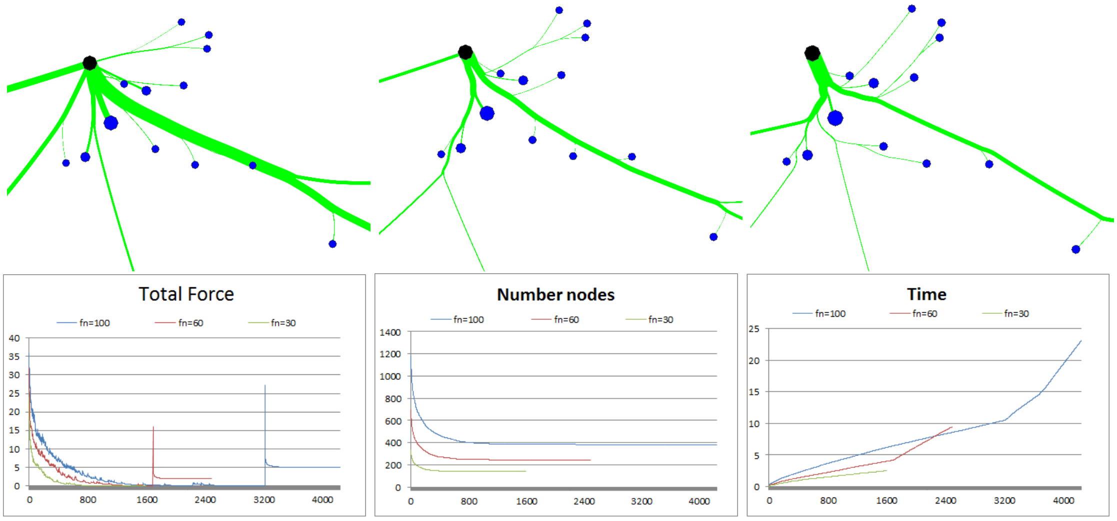


Figure 7.10: On top: flow map with $f_n = 30$ (left), flow map with $f_n = 60$ (centre), flow map with $f_n = 100$ (right). On bottom: the x axis represents the number of iterations and the y axis the total force of the system (left plot), the number of nodes (central plot), and the execution time (right plot).

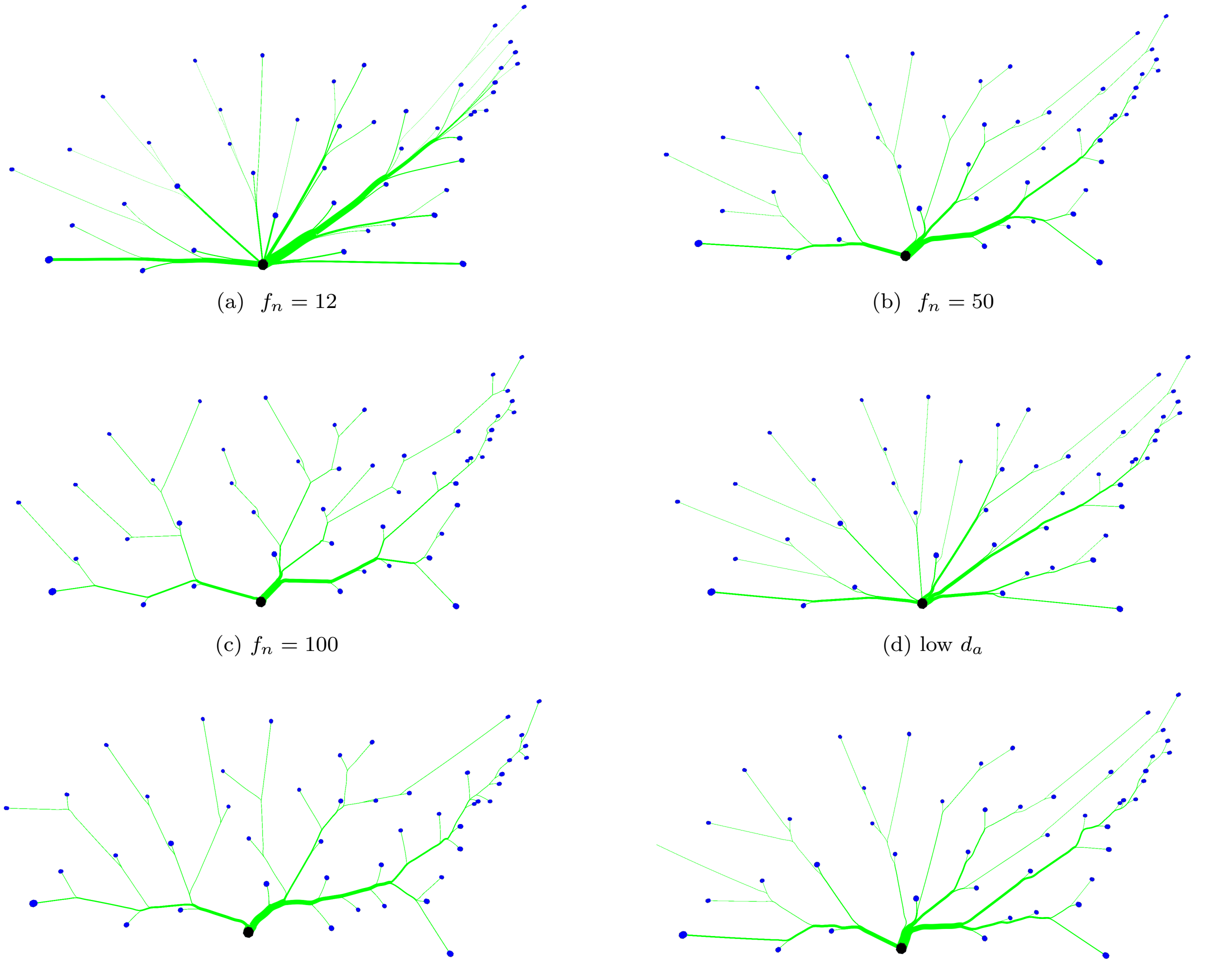
intermediate nodes does not guarantee that all flows are cross-free or the absence of crossing with the leaf nodes, see respectively the second and first map in Figure 7.10.

The plot depicting the total force illustrates when the algorithm finds a local minimum of the total force and when it applies the repulsive force instead of the attractive force to its nodes; for each of 3 cases the force has a pick and then it decreases until a minimum is reached. It is possible to notice that the convergence is fast using the total force of the system as cost value.

In the plot depicting time, when the stress repulsive force is introduced to overcome the overlapping of destination nodes, the function increases the time; in the first step the time complexity is in the order of $O(i \cdot n)$, meanwhile in the second step the destination nodes must be considered, making the time complexity in the order of $O(i \cdot m \cdot n)$. Our algorithm performs a flow map with intermediate nodes in the order of one thousand in less than a minute.

In addition to f_n and distance d between intermediate nodes, we have the following parameters: the threshold d_s that defines when two positions are equivalent; the distance d_a for the attractive force; the distance d_r for the repulsive force; and the constant oscillation k_s that varies between 0 to 1. After performing several tests we concluded empirically that 0.1 is an optimal value.

Figure 7.11 depicts the effects of varying the main parameters of the algorithm. The first three images represent the same dataset as flow map but varying the number of intermediate nodes. In Figure 7.11(a) the factor f_n is 12, in Figure 7.11(b) f_n is 50, and



(a) $f_n = 12$

(b) $f_n = 50$

(c) $f_n = 100$

(d) low d_a

(e) the magnitude factor is removed from equation 7.1

(f) the magnitude factor is removed from equation 7.2

Figure 7.11: Various results of our algorithm with different parameters.

in Figure 7.11(c) the value is 100.

We observed that in situations where the number of nodes is low, the merging of flow lines is reduced and the result is often not satisfactory. We have also noticed that after a certain threshold the algorithm produces no significant improvements.

We also noticed what happens if the factor that takes into account the magnitude is removed from Equation 7.1 (Figure 7.11(e)) or from Equation 7.2 (Figure 7.11(f)).

The first case generates sparser results. In the former case, the position of each split does not follow the direction of the lines (for example, the main branch on the right of the origin). In Figure 7.11(d), the range for the attractive force is reduced (low d_a). It cause an increment of the number of branches mainly near the origin.

The datasets used in this work can be retrieved online from: the Scotch Whisky Association in Edinburgh [7], Statistics Norway [3] and U.S. Census Bureau, County-to-County

Migration Flows [25].

All flow maps depicted in this chapter were computed and visualized on a Intel laptop with 1.64 GHz and 2 GB of RAM. The algorithm was initially developed using a 2D Java canvas, but was later ported to 3D and integrated in NASA WorldWind [122]. Hence, both 2D and 3D views are supported.

7.5 Conclusion and Future Work

This chapter describes a force-directed algorithm for the automatic generation of flow maps. The algorithm for computing the flow map iteratively minimizes the energy of a system composed by a set of forces that characterise a well-drawn flow map.

The main aspect is that the flow tree, as well as the flow lines, are mainly based on the magnitude of the destinations. The second aspect is the possibility to supervise the output during each iteration. Even if the visual quality of our algorithm was not proved to be the best compared with the previous work, it depicts good quality flow maps. Moreover, the method is of easier implementation, and the time requested for the execution is in the order of seconds for flow maps of thousand of intermediate nodes.

As a further improvement, we could use a quadtree to store the position of each node and to reduce the time needed to avoid crossings between leaf nodes and flow lines. The algorithm could also be extended to support multi-origin representations. The use of a 3rd dimension can be another challenging direction for future work.

Chapter 8

Depiction of Multivariate Data through Flow Maps

> *The greatest value of a picture*
> *is when it forces us to notice*
> *what we never expected to see.*
> –John W. Tukey.

8.1 Overview

Flow maps are graphical representations that depict the movement of a geospatial phenomenon, e.g., migration and trade flows, from one location to another. These maps depict univariate spatial origin-destination datasets, with flows represented as lines and quantified by their width. As mentioned in Chapter 2, a main feature of these maps is the aggregation of flows that share the same origin. Thus, visual clutter is reduced and the readability improved.

This chapter describes a novel technique that extends flow maps to visualize multivariate geographical data. Instead of a univariate color scheme, we interpolate strategic color schemes to communicate multivariate quantitative information. Additionally, our approach crystallizes on augmenting flows with pie charts. To evaluate the relevance and impact of our approach, a case study is presented. Part of the work in this chapter was previously published in [38, 40].

8.2 Introduction

An increasing amount of spatial datasets are now freely available on the Internet, and this number is expected to increase as sensor networks are becoming a ubiquitous part of our environment. A substantial subset includes origin-destination data describing the movement of phenomena between locations. Phenomena can involve both tangible (e.g. bank notes, people, and goods) and intangible objects (e.g. reputation, ideas, and energy).

Static media, such as paper maps, and dynamic media, such as electronic devices can be used to transport geoinformation. Compared to the latter, static media impose certain requirements on the map production, especially when multivariate data has to be communicated. Since the user cannot manipulate the mapping information in a static context, it must be encoded in such a way that is easily understandable for the viewer.

This chapter is focused on the development of new static representations to depict spatial origin-destination data with multiple attributes. The requirements fulfilled by our approach can be clustered in constraint and analytical requirements.

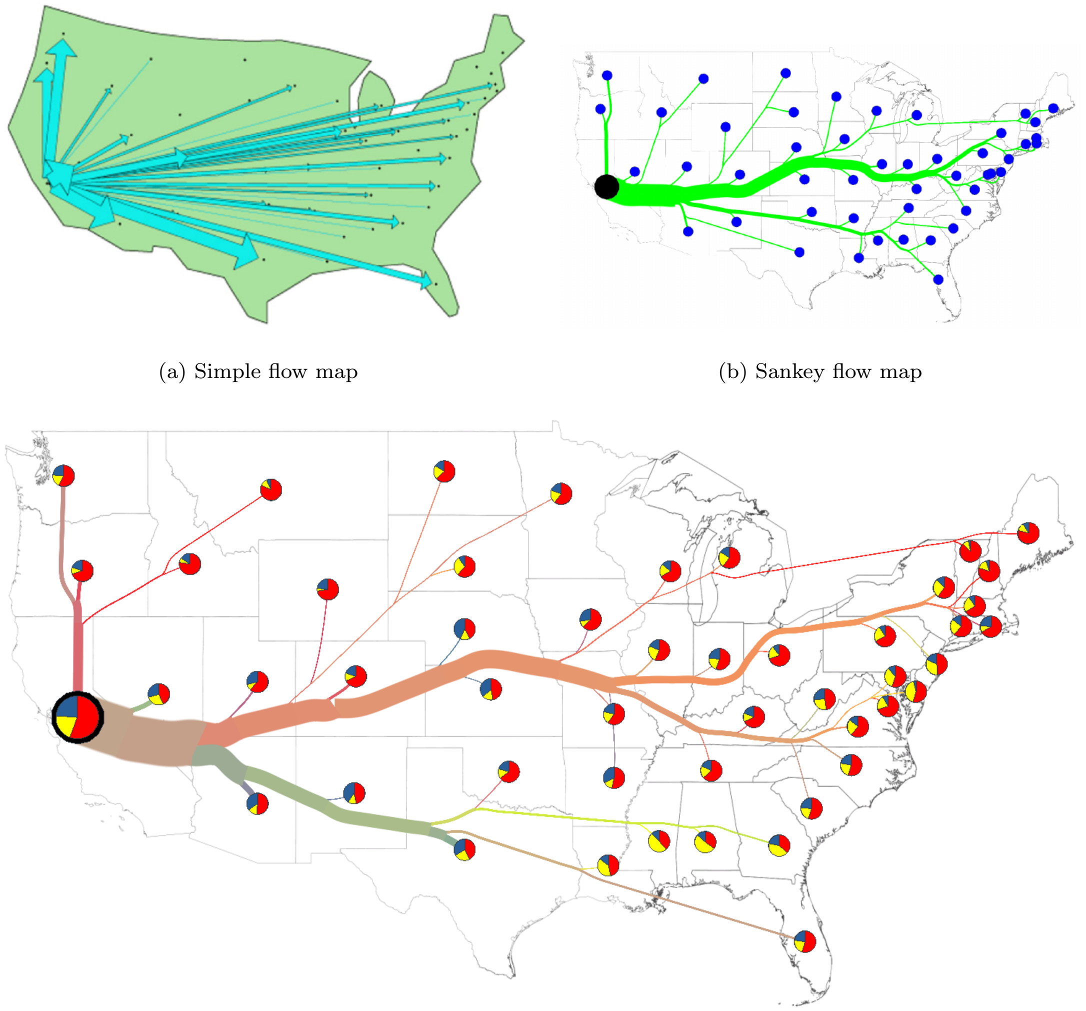
Constraint requirements:

- CR1. The mapping must be static.
- CR2. The information must be depicted on a limited space (e.g. the computer screen or a page on a newspaper) in such a way that it is clearly understandable by the viewer.
- CR3. The number of visual primitives must be minimized to reduce the cognitive overload.

Analytical requirements:

- AR1. Possibility to reason about the geographic patterns.
- AR2. Must enable, simultaneously, the visualization of the flow structure and its multivariate patterns.
- AR3. Possibility to analyze the outliers and commonality between each destination.
- AR4. Possibility to analyze the outliers and commonality for groups of moving entities.

Parallel coordinates are popular visualization methods for displaying multidimensional data. They consist of several vertical axes, each representing one data dimension. A data entity consists of several data points, each located on one of the axes. *Star plot*, also called radar chart, is a similar method. The axes are equally spaced around a circle, originating from the center. The values of neighboring axes are connected by a line, resulting in a star-like shape. Although parallel coordinates and star plots are powerful methods, which do not require any interactive media, they do not preserve the geographic information, i.e. the position of each location (the requirement AR1 is not satisfied). Flow map representations are more suitable to communicate univariate spatial origin-destination data. They depict the movement of phenomena between geographic locations [151]. The curves in a flow map, called flow lines, describe the movement of a phenomenon from one location to another. Flow maps offer a simple and effective instrument to depict



(a) Simple flow map

(b) Sankey flow map

(c) Sankey flow map enriched with information about the ethnicity of migrations: red are Caucasian people, yellow are African Americans people and blue are Asian people. Compared to (a) and (b), (c) communicates more information without heavily increasing the visual complexity.

Figure 8.1: Single-origin flow maps depicting immigration from California

geographical movement. However, its visualization without visual clutter and cognitive overload, can be challenging (Figure 8.1(a)). An extension to tackle those problems is the Sankey flow drawing technique [146]. Flow lines sharing the same origin are aggregated together and their width is defined proportional to the sum of the flows' magnitude that they represent (Figure 8.1(b)).

There are three approaches that extend Sankey flow maps so they can be used to depict multivariate data:

- (1) Subdivision of flow lines: In the first approach each line is subdivided in "parallel channels", one for each attribute, with a specific color and width proportional to its value. Since the number of flow lines grows proportional to the number of attributes and each flow line has a specific width and margin, then this approach also requires more space (which conflicts with CR2). Because space is limited, this re-organization of the map is a challenging task. Further, the increased number of visual primitives may increase the cognitive load of the user, thus, decreasing its performance (the requirement CR3 is not satisfied).

- (2) Animation: The second approach uses a dynamic view and represents each attribute in separated maps. The latter can be shown in sequence in predefined time intervals, or interactively using triggers to switch between the separated maps (the requirement CR1 is not satisfied).
- (3) Small Multiples: The last approach uses small multiples ([171]) defined as a grid of images. Depicting a limited number of attributes they can provide a very good support for the analysis of the changes of the overall distribution of the flows for different attributes.

We propose a new approach that capitalizes on the value of color techniques, as already done in *choropleth maps*, and on the aggregation feature of Sankey flow maps (Figure 8.1(c)). First, a primary color is associated with each numerical attribute of the multivariate data set. Then, these colors are combined proportionally to the value of their attribute to colorize the flow lines (the requirement CR3 satisfied). Moreover, each leaf node is depicted as a *pie chart* where the slices identify the amount of entities moving to that destination. This method does not require an interactive platform, thus it can depict a map as a static image (the requirement CR1 is satisfied). Compared to small multiples, this method requires less screen space because one map is used (the requirement CR2 is satisfied). The main limitation of our method is that it manages only up to 3 attributes simultaneously. Yet, this limitation can be tackled by combining colors with shape patterns.

8.3 Proposed Approach

We extend the algorithm for the automatic generation of Sankey Flow map (described in Chapter 7), to depict more than one attribute. Compared to simple Sankey Flow maps, the expressiveness of nodes and lines are enriched by a continuous color scheme (see Section 8.3.1) applied to the lines (see Section 8.3.2) and to the nodes represented by circular diagrams (see Section 8.3.3).

The geographic dataset is represented as a tree t where the origin is the root node r and the n geographical destinations correspond to the leaves $l_1, l_2, ..., l_n$, respectively with flow magnitudes $magn(l_1)$, $magn(l_2)$,..., $magn(l_n)$. Each destination (represented by a leaf) can have associated complementary attributes, i.e., their union is equal to the whole magnitude. An example of supported attributes is the number of males and the number of females of migrating people. Another example is the motivation that led them to travel: family, job and study. Let $attr\_x_i(l) \in [0, 1]$ the value of the characteristic x (e.g. x can identify the motivations) of the attribute with index i (e.g. i can identify the family, job or study) assigned for each destination l, such that:

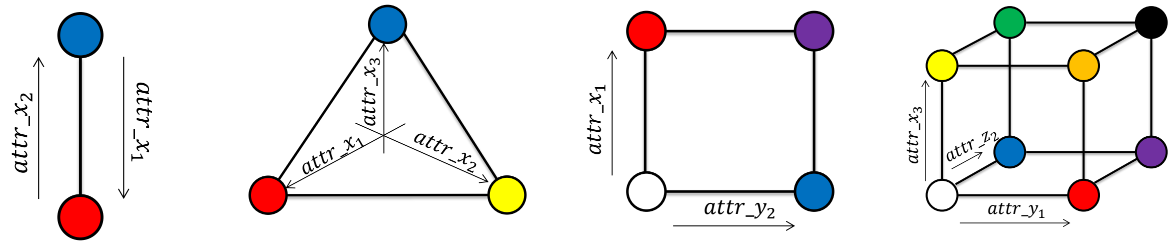
$\sum\limits_{i=1}^{h} attr\_x_i(l) = 1$

where h can be at most 3. Our approach can be applied to datasets with the following data characteristics:

- Case 1: Two attributes $attr\_x_1$ and $attr\_x_2$, which compose the whole amount, i.e., for each leaf l, $attr\_x_1(l)$ and $attr\_x_2(l)$ sum up to 1.
- Case 2: Three attributes $attr\_x_1$, $attr\_x_2$, and $attr\_x_3$, which compose the whole amount, i.e., for each leaf l, $attr\_x_1(l)$, $attr\_x_2(l)$ and $attr\_x_3(l)$ sum up to 1.
- Case 3: Two complementary attributes from different characteristics $attr\_x_1$ and $attr\_y_2$, i.e., for each leaf l, $attr\_x_1(l)$ and $attr\_y_2(l)$ not necessarily sum up to 1.
- Case 4: Three complementary attributes from different characteristics $attr\_x_1$, $attr\_y_2$ and $attr\_z_3$, i.e., for each leaf l, $attr\_x_1(l)$, $attr\_y_2(l)$ and $attr\_z_3(l)$ not necessarily sum up to 1.

8.3.1 Color models

The color is used to convey information about the attributes of the geographical data depicted in a flow map. There are many color models to specify colors in maps. Models that are based on hardware specifications, such as RGB, are frequently used because of their long history and their consequent familiarity. Although RGB color spaces reflect color mixing operations in color printers and computer monitors, they are not recommended as reference model of color for human beings. In fact, pigment mixing in early childhood art training makes use of subtractive color models that differ from RGB the model. For such reason, our work employs a technique designed by Gossett et al. [71]; inspired by paint mixing using a subtractive color space with Red, Yellow and Blue as primary colors. To obtain an effective color coding it is fundamental to have color mapping function that is invertible, i.e., every attribute value (or every group of data values) is associated with exactly one color, and viceversa [166]. Moreover, a color compositing technique is only useful if it allows the viewer to discern the individual components that form the visualization[71].



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

Figure 8.2: RYB interpolation patterns used for each case.

For each aforementioned cases, different attributes are associated to the color mapping function, thus different color spaces are used.

In Case 1, we use a pattern composed by red and blue. $attr\_x_1$ is associated to the red channel and $attr\_x_2$ is associated to the blue channel, see Figure 8.2(a).

In Case 2, we use a triaxial graph composed by red, yellow and blue. In this case $attr\_x_1$ is associated to the red channel, $attr\_x_2$ is associated to the yellow channel and $attr\_x_3$ is associated to the blue channel, see Figure 8.2(b). Only a small portion of the color space is used because the sum of the primary colors has to be 255.

In Case 3, we use the square pattern composed by red and blue colors. The association between attributes is the same as Case 1, see Figure 8.2(c).

Finally, in Case 4, we use the entire colors space N^3, and the association between attributes is the same as Case 2, see Figure 8.2(d).

Generating a large number of colors from a limited range of values can be challenging. If users cannot distinguish between colors then they cannot perceive the magnitude of the value it represents. Hence, we normalize each primary color interval using their minimum and maximum value, see Section 8.4.

As shown in Figure 8.2(a), for Case 1 the two attributes share the same range, therefore the formula to normalize their value is:

$$norm\_attr\_x_{var}(l) = \frac{attr\_x_{var}(l) - min(attr\_x_{var})}{max(attr\_x_{var}) - min(attr\_x_{var})}, var = 1, 2. \tag{8.1}$$

Figure 8.3 shows an example of Case 1 where the values of the attributes are normalized. For Case 2 the formula to normalize the value of each attribute is defined as follows:

$$norm\_attr\_x_{var}(l) = \frac{attr\_x_{var}(l) - min(attr\_x_{var})}{1 - min(attr\_x_1) - min(attr\_x_2) - min(attr\_x_3)}, var = 1, 2, 3. \tag{8.2}$$

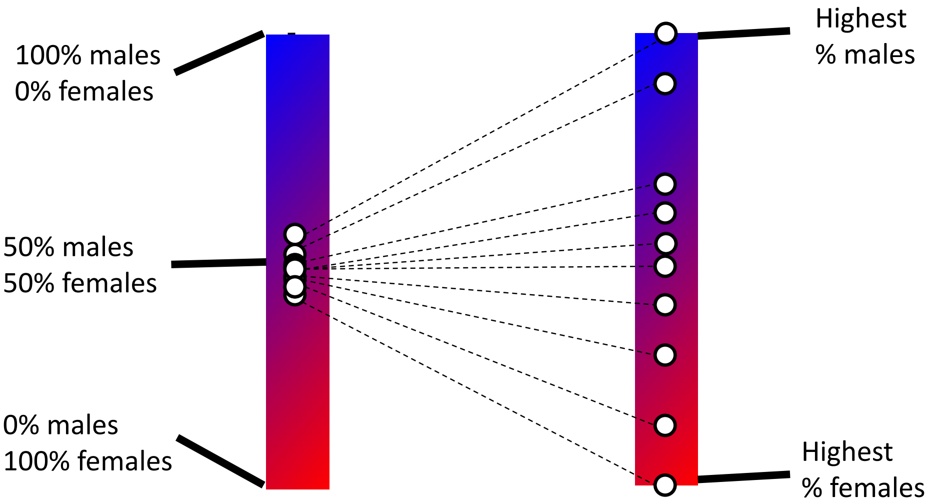


Figure 8.3: Example of procedures to normalize the attributes in Case 1.

For Case 3 and 4, where attributes describe different characteristics, we normalize the attributes according to its minimum and maximum value (see Equation 8.1).

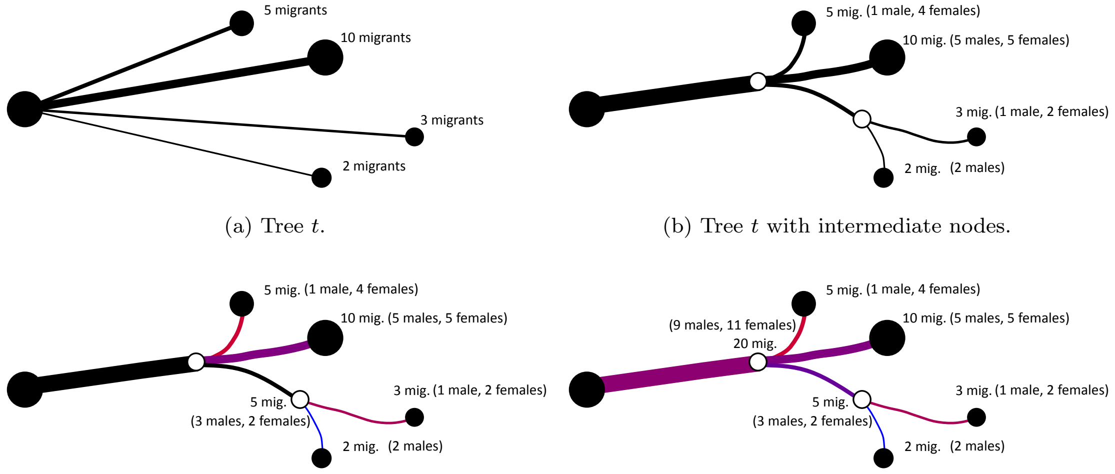
8.3.2 Flow Representation

Instead of splitting the generalized flow in child flows with distinct colors, we decided to keep the original flow representation and work on its color. Hence, our approach does not increase the complexity of the representation in terms of number of depicted features. In particular, the color of a line is the mix of the colors of the aggregated lines.

The extended algorithm aims to create, aggregate, and color the lines that connect the root with the leaves. The flow lines are defined as cubic splines that connect the root with every leaves. Each flow line is composed by polynomial curve segments that pass through a set of intermediate nodes. The width and the color of each curve segment depends on the attribute $attr\_x_i$ of the node corresponding to its end point. The value of an attribute $attr\_x_i$ for a given intermediate node n is calculated as follows:

$$attr\_x_i(n) = \sum_{m \in child} (magn(m) * attr\_x_i(m))/magn(n)$$

where the set *child* contains all the child nodes of the node n. The color of each curve segment is calculated using the attributes of its end point and the appropriate mapping function described in Section 8.3.1. Figure 8.4 shows an example of generation of flow lines.



(a) Tree *t*.

(b) Tree *t* with intermediate nodes.

(c) Tree *t* with some colored flow lines.

(d) Final flow lines representation.

Figure 8.4: Example of generation of flow lines. Each destination has associated the number of migrants and two complementary attributes: number of males and number of females.

8.3.3 Node Representation

Flow maps usually require only a base map and straight lines or curves as graphical features. An optional feature is the circle used to represent the destinations, which can be a valuable asset during the visualization process. It can help to better identify the target of the flows, and its size can be used to communicate the magnitude of an attribute. In our work, we want to increase its value even further, as pie charts, to maximize the information that can be communicated about the attributes. A pie chart is a circular chart divided into sectors, illustrating numerical proportion. Although this representation is not always suitable to depict data, as claimed by Few [65, 64], we decided to adopt it due to the following reasons:

- Nodes are usually depicted as circles. Therefore, pie charts can be easily integrated without affecting the layout of the flow map.
- The attributes are at most three, thus this representation is not misleading.
- The empty space inside the node representation is converted into visual information.

The primary colors that compose the flow line are used to create the pie chart. This helps the reader to estimate the percentage of each variable, simplifying his/her cognitive process. The root node is represented as a pie chart depicting the percentage of each attribute from the whole amount of moving entities. Depending on the number of attributes to depict, different variants of pie chart are used, as shown in Figure 8.5.

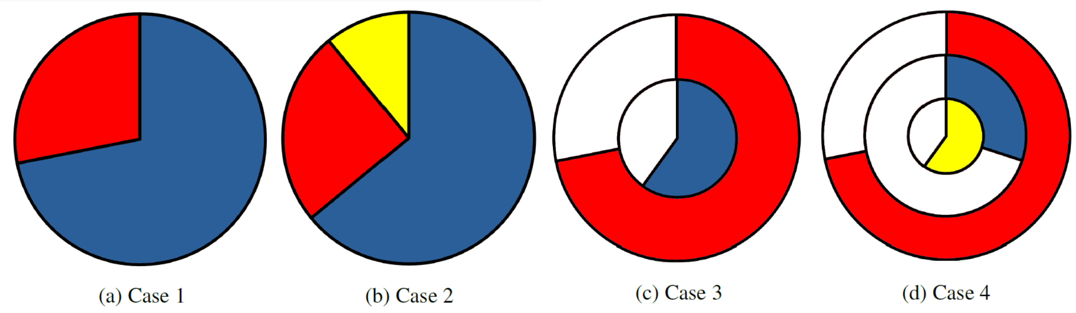

(a) Case 1
(b) Case 2
(c) Case 3
(d) Case 4

Figure 8.5: Different node representations using pie charts. The cases are related to the RYB interpolation patterns of Figure 8.2.

The representation of two (Case 1) or three (Case 2) complementary attributes is depicted with a pie chart with respectively two or three slices. Multi-Series pie charts are used to depict unrelated attributes, with each ring representing an attribute (Case 3 and 4).

8.4 Case Studies

In this section, we provided an evaluation of our approach on the basis of the County-to-County Migration Flows dataset [24]. It describes migration phenomena from multiple states of the USA. Although this dataset has been extensible used to generate flow maps [62], [129], [133], [174], existent algorithms depict only one attribute. As opposite, with our approach, the color of the flow lines and the pie charts enable the viewer to gain information not only about the number of people moving but also about their age group and ethnicity.

In order to emphasize differences between the attributes, all values are normalized following the procedure described in Section 8.3.1. Thus, the color of the flow lines does not identify exactly the percentages of each attribute, but communicates whether the attribute is greater, lesser or equal to the same attribute in other flow lines. For example if a flow line is 100% red, it does not mean that the percentage of the attribute associated to red is 100%. It means that the flow line has the highest percentage of the red attribute compared to other flow lines and the lowest percentage of other attributes respectively.

In Figure 8.6 it is possible to extract information about age groups in the migratory flows from Colorado. This attribute is subdivided into three groups: younger than 30 (red), ranging from 30 to 54 (yellow) and over 54 (blue). Pie charts depict the percentage of people for each group. Moreover, flows color encodes such information also for aggregated destinations. For example, by looking at the main split in the west of Colorado, we can conclude that the flow directed to north (area A) has more migrants with age between 30 and 54 than those directed to California and Nevada (area B) that have more young people. The flows directed to the north east (area C) contain a high percentage of young people.

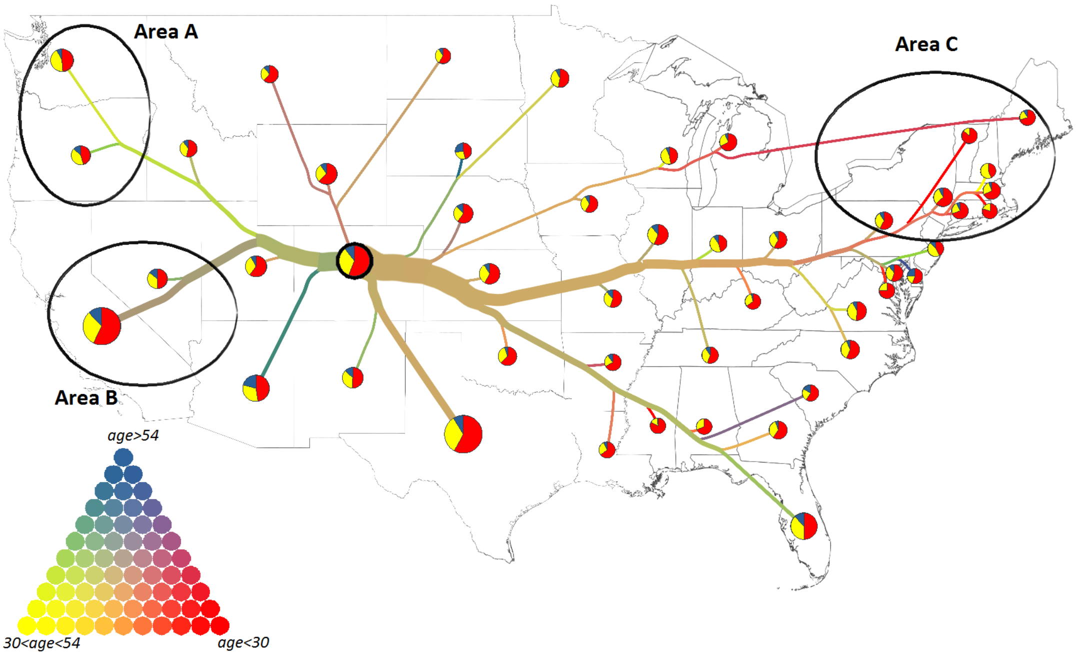


Figure 8.6: Migratory flows from Colorado. Color identifies age: red are people younger than 30, yellow from 30 to 54 and blue the over 54.

In Figure 8.7 the flow of migrants is shown together with the information about their ethnicity. The colors red, yellow and blue represent the Caucasian, African Americans and Asian ethnicity, respectively. The thick flow directed to Arizona, New Mexico and Texas (Area A) is colored with a more vivid blue, which indicates a higher percentage of Asians migrants. As opposite, in the south east (area B) African Americans people are the majority of the migrants.

The flow map depicted in Figure 8.8 shows the migration from Texas. It is enriched with information describing different characteristics: the Asian people that are moving (red) with age from 30 to 54 (blue). The map can help analysis to find correlations between these two attributes. If the color tends to be more red, a high percentage of migrants are Asian people and if the color tends to be more blue, a high percentage of migrants have an age between 30 and 54. Moreover, the lightness/darkness of the color defines respectively if both attributes have low/high percentage with respect to the whole amount of people that are moving. The thick flow directed to west (Area A) is colored with a strong red, indicating that the presence of Asian is higher over the migrants directed in that zone. In Area B the lines are colored primarily with blue, which implies a high percentage of

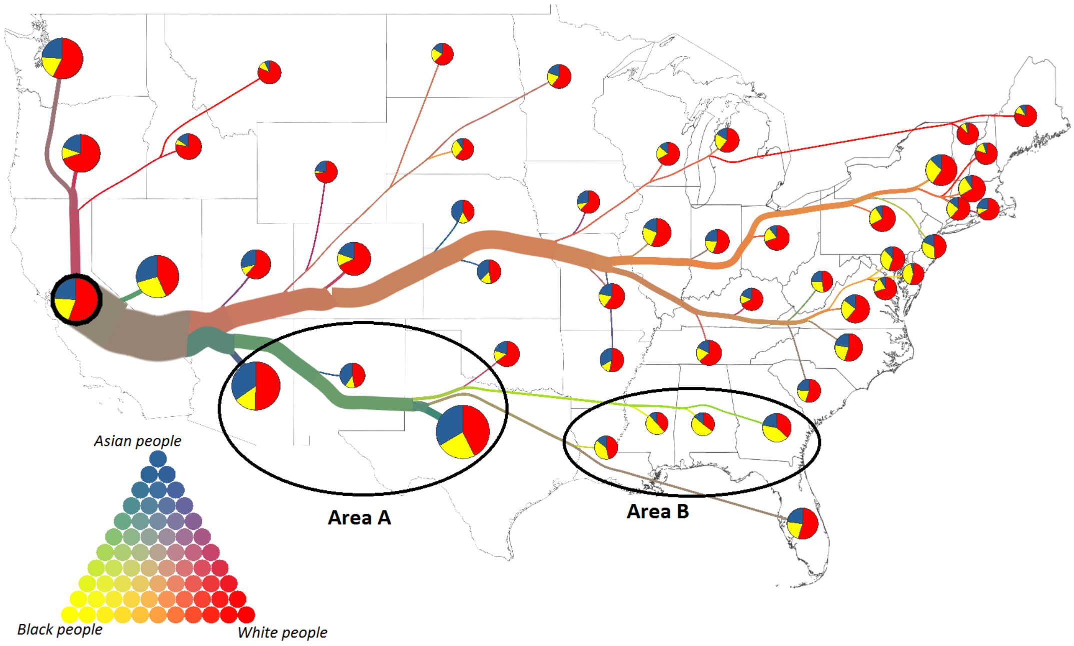


Figure 8.7: Migratory flows from California. Color identifies the ethnicity of migrants: red are Caucasian people, yellow are African Americans people and blue are Asian people.

people with ages between 30 and 54 and a low percentage of Asian, if compared with other flows of migrants.

8.5 Conclusion and Future Work

A color scheme blending is used in conjunction with the aggregation aspects of flow maps and pie charts, to visualize in a static representation spatial origin-destination data with multiple attributes. Compared with parallel coordinates and star plots, our approach analyses the outliers and commonality between each destination and for groups of moving entities. Moreover, it overcomes the limitations of the techniques that extend flow maps with animation, small multiples and subdivision of flow lines.

Case studies are performed through three automatically generated flow maps to communicate information not only about the number of migratory flows but also about their age group and ethnicity. However, to further validate our work a user study must be performed.

The analysis of the geographic patterns depends on the criteria used to aggregate the flow lines, i.e., different flow maps that depict the same dataset can give a different interpretation. However, the algorithm used for the automatic generation of the flow map gives the cartographer the possibility to modify the direction and the aggregation of the flow lines.

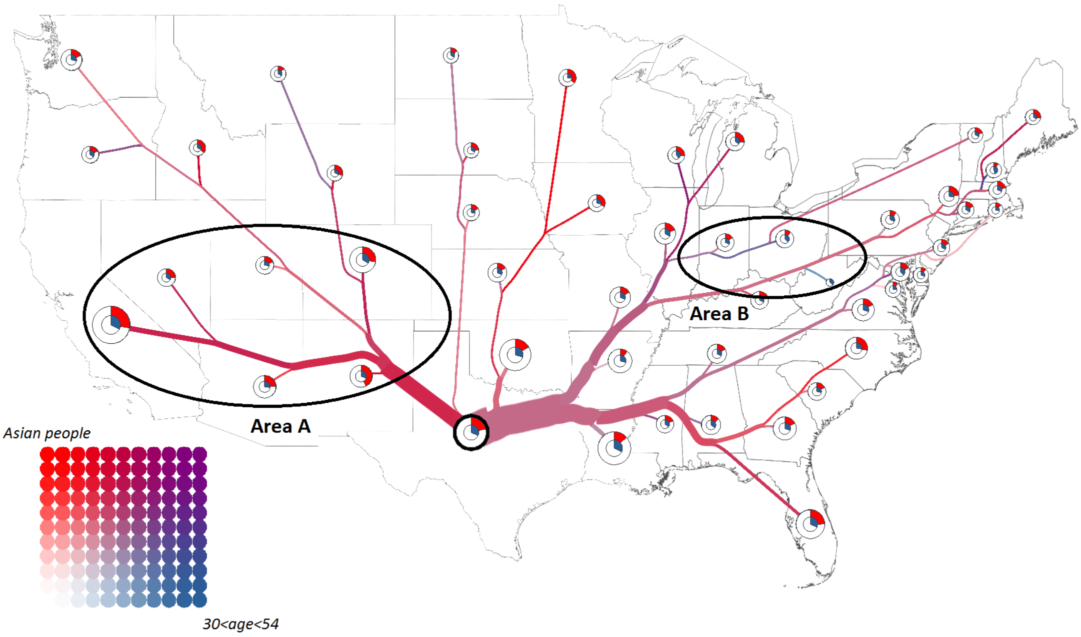


Figure 8.8: Migratory flows from Texas. Color identifies two attributes from different characteristics (ethnicity and age) of people that are moving: (Red) Asian people, (Blue) people with age between 30 and 54.

As future work, we plan to reduce significantly the used colors discretizing the color scale. In fact, cartographers recommend to use five to seven classes of color for a map [47].

Chapter 9

Conclusion

The only source of knowledge is experience.
–Albert Einstein

This thesis focused on the class of geographical visualization where nodes are added to a background map. They are successfully used in different real case scenarios: the tourist consulting metro maps is facilitated in the use of subways; the citizen observing flow maps on newspaper become aware of the main migration flows that are occurring today. Moreover, geographic node-link diagrams are typically embedded in interactive systems to monitor communication networks or to investigate the spatial correlation of climate data.

This thesis has mainly investigated the problems related to visual clutter in geographic node-link diagrams. It is motivated by the fact that, although such visualizations are very popular to perform geographical analysis and convey information, they often suffer of readability issues.

9.1 Contributions

In Chapter 2, we started classifying the geographic node-link diagrams according to the way the nodes are spatially organized in the presentation space. This solution facilitates the choice of an appropriate visualization, for a given geographic graph, according to the geographic attributes the designer wants to visually represent.

For that class of diagrams, in Chapter 3, we surveyed from literature, the main approaches designed to solve a cluttered state. The solutions are divided in two groups: graph visualizations and interactive techniques.

In Chapter 4, the problems related to visual clutter are tackled proposing a schematization of techniques that helps the reader to decide, given a task and a geographical node-link diagram, which are the candidate solutions that facilitate the reduction of the visual clutter. This work provides a list of criteria to classify the effects of the visual clutter on different scenarios. Moreover, it offers guidelines for the design of novel techniques, helping the researchers to focus on a well-defined list of criteria to fulfill.

Furthermore, we propose solutions aimed to face the visual clutter through the design and implementation of interactive techniques. *3DArcLens*, described in Chapter 5, is an interactive lens that deforms the links shape to reveal underneath information and to reduce ambiguities related to graph topology. It is designed to work on both 2D and 3D visualizations.

GeoPeels, described in Chapter 6, is an interactive deformation-based technique, used in combination to zoom and pan, that capitalizes on dynamic globe deformations for an effective exploratory visual analysis of global networks. The graphical performance evaluation of both 3DArcLens and GeoPeels is based on the measurement of frame rate, expressed in frames per second. Meanwhile, the qualitative evaluation of the techniques relied on a comparison between them and existing approaches respect to criteria they are able to fulfill. Finally, the evaluation of usefulness is performed presenting some case studies on real datasets.

Chapter 7 is devoted to present an algorithm to automatically create flow map layouts through attractive forces among flow lines. It allows the user to support for an optional supervision of the graph layout. We evaluated its performance comparing our approach and the existing techniques respect to: the computational cost, the list of aesthetic criteria we defined, and the visual quality of the generated maps. Chapter 8 presents an extension to visualize multivariate geographical networks. Three case studies are performed to evaluate the usefulness of the presented technique.

9.2 Limitations and Future Work

Additionally to limitations and future work described at the end of each chapter, this section focuses on the general aspects of this Thesis.

Survey of alternative visualizations. This Thesis surveys existing techniques and proposes different approaches to improve the readability of a geographic node-link diagram acting of its visual variables and attributes. However, in many cases is preferred to change the visualization rather than to improve it. The consideration of other design alternatives such as *Adjacency Matrix* [80] and *OD Map* [188] could be marked as future work. In particular, the classification of visualizations techniques described in Chapter 3 and 4 should be updated to describe also alternative representations.

Task-based solutions. The word *task* has different meaning in the information visualization literature and it is used at multiple levels of abstraction and granularity [119]. For example the word *problem* can denote a task described in domain terms, meanwhile the word *operation* can denote a task in an abstract level.

This thesis only partially mention the tasks the user (or viewer) wants to achieve starting from a geographic network. Hence, this aspect could be investigated in future work. In particular, the classification of geographic visualizations, proposed in Chapter 2, did not include the recommended tasks as criteria for each visualization group. As opposite, the schematization of clutter reduction techniques, described in Chapter 4, it is based on operations derived from a taxonomy of general graph tasks. Moreover, *3DArcLens* and *GeoPeels* were designed to facilitate generic operations such as "following a path through a graph". However, it was not performed an investigation based on the domain problems such techniques support.

Although there are task taxonomies for general graphs [105], dynamic graphs [5, 99], partitioned graphs [144] and multivariate graphs [132], no taxonomy is specifically designed to group tasks for geographic graphs. Hence, the definition of a task taxonomy for such class of networks can be definitively an interesting future work.

Integration in complex systems. If the operations involve the analysis and exploration of geographic networks, a node-link diagram is not able to work as a stand alone tool. As opposite, in most of cases, it becomes a feature of complex systems [157]. Thus, one future challenge is to find applications where network visualization creates enough value and then embed our interactive techniques into them as part of the system. However, since both users and tasks vary considerably, systems need to be flexible and easily configurable by users [125].

Notable examples of existing systems that support visual network analysis and exploration are the following:

- *CGV* [163] focuses on dense graphs. It allows the user to interact with lens techniques over multiple interlinked views.

- *Gephi* [10] is a framework that supports different graph types: clustered, multivariate and dynamic graphs.

- *Tulip* [8] is a library to develop multiple interlinked views. It supports the visual exploration of general graphs.

- *GeoVISTA Studio* [156] is an open source software focusing on geographic visualizations. A visual programming user interface allows the designer to interconnect different graphical components. This aspect enables scientists with little programming knowledge to develop useful tools in short time.

The final goal is to customize our solutions for specific applications with real users. This step is mandatory to move the research to innovation.

Bibliography

[1] Adobe illustrator. `http://www.adobe.com/it/products/illustrator.html`. Accessed: 2014-04-24.

[2] Caidas ipv4 and ipv6 as core as-level internet graph. `https://www.caida.org`. Accessed: 2016-02-09.

[3] Statistics norway. `http://www.ssb.no`. Accessed: 2014-04-24.

[4] James Abello, Frank Van Ham, and Neeraj Krishnan. Ask-graphview: A large scale graph visualization system. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):669–676, 2006.

[5] Jae-wook Ahn, Catherine Plaisant, and Ben Shneiderman. A task taxonomy for network evolution analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 20(3):365–376, 2014.

[6] Başak Alper. *Dynamic visualization of geographic networks using surface deformations*. PhD thesis, Citeseer, 2006.

[7] Scotch Whisky Association. statistical report 2012, 2012.

[8] David Auber. Tulipa huge graph visualization framework. In *Graph Drawing Software*, pages 105–126. Springer, 2004.

[9] Moshe Bar and Maital Neta. Humans prefer curved visual objects. *Psychological science*, 17(8):645–648, 2006.

[10] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.

[11] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G Tollis. *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998.

[12] Patrick Baudisch and Ruth Rosenholtz. Halo: a technique for visualizing off-screen objects. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 481–488. ACM, 2003.

[13] Richard A. Becker, Stephen G. Eick, and Allan R. Wilks. Visualizing network data. *Visualization and Computer Graphics, IEEE Transactions on*, 1(1):16–28, 1995.

[14] Jacques Bertin. Semiology of graphics: diagrams, networks, maps. 1983.

[15] Joachim Bottger, Ulrik Brandes, Oliver Deussen, and Hendrik Ziezold. Map warping for the annotation of metro maps. *Computer Graphics and Applications, IEEE*, 28(5):56–65, 2008.

[16] Joachim Böttger, Martin Preiser, Michael Balzer, and Oliver Deussen. Detail-in-context visualization for satellite imagery. In *Computer Graphics Forum*, volume 27, pages 587–596. Wiley Online Library, 2008.

[17] Quirijn W Bouts and Bettina Speckmann. Clustered edge routing. In *Visualization Symposium (PacificVis), 2015 IEEE Pacific*, pages 55–62. IEEE, 2015.

[18] Ilya Boyandin, Enrico Bertini, Peter Bak, and Denis Lalanne. Flowstrates: An approach for visual exploration of temporal origin-destination data. In *Computer Graphics Forum*, volume 30, pages 971–980. Wiley Online Library, 2011.

[19] Ulrik Brandes, Galina Shubina, and Roberto Tamassia. *Improving angular resolution in visualizations of geographic networks*. Springer, 2000.

[20] Ulrik Brandes, Galina Shubina, Roberto Tamassia, and Dorothea Wagner. Fast layout methods for timetable graphs. In *Graph Drawing*, pages 127–138. Springer, 2001.

[21] Ulrik Brandes and Dorothea Wagner. Using graph layout to visualize train interconnection data. In *Graph Drawing*, pages 44–56. Springer, 1998.

[22] Felix Brodkorb, Arjan Kuijper, Gennady Andrienko, Natalia Andrienko, and Tatiana von Landesberger. Overview with details for exploring geo-located graphs on maps. *Information Visualization*, page 1473871615597077, 2015.

[23] John Brosz, Sheelagh Carpendale, and Miguel A Nacenta. The undistort lens. In *Computer Graphics Forum*, volume 30, pages 881–890. Wiley Online Library, 2011.

[24] Census Bureau. Migration/geographic mobility, June 2014.

[25] U.S. Census Bureau. County-to-county migration flows, 2012.

[26] Stefan Buschmann, Thomas Nocke, Christian Tominski, and Jrgen Dllner. Towards visualizing geo-referenced climate networks. In *Proceedings of Workshop GeoViz Hamburg 2013*, 0 2013.

[27] David M Butler, James C Almond, R Daniel Bergeron, Ken W Brodlie, and Robert B Haber. Visualization reference models. In *Proceedings of the 4th conference on Visualization'93*, pages 337–342. IEEE Computer Society, 1993.

[28] Stuart K Card and Jock Mackinlay. The structure of the information visualization design space. In *Information Visualization, 1997. Proceedings., IEEE Symposium on*, pages 92–99. IEEE, 1997.

[29] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[30] Chaomei Chen. Top 10 unsolved information visualization problems. *Computer Graphics and Applications, IEEE*, 25(4):12–16, 2005.

[31] Bill Cheswick, Hal Burch, and Steve Branigan. Mapping and visualizing the internet. In *USENIX Annual Technical Conference, General Track*, pages 1–12. Citeseer, 2000.

[32] Andy Cockburn, Amy Karlson, and Benjamin B Bederson. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):2, 2009.

[33] Kenneth C Cox, Stephen G Eick, and Taosong He. 3d geographic network displays. *ACM Sigmod Record*, 25(4):50–54, 1996.

[34] Weiwei Cui and Huamin Qu. A survey on graph visualization. *PhD Qualifying Exam (PQE) Report, Computer Science Department, Hong Kong University of Science and Technology, Kowloon, Hong Kong*, 2007.

[35] Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li. Geometry-based edge clustering for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1277–1284, 2008.

[36] Giordano Da Lozzo, Marco Di Bartolomeo, Maurizio Patrignani, Giuseppe Di Battista, Davide Cannone, and Sergio Tortora. Drawing georeferenced graphs - combining graph drawing and geographic data. In Lars Linsen, Andreas Kerren, José Braz, Lars Linsen, Andreas Kerren, and José Braz, editors, *Proceedings of the 6th International Conference on Information Visualization Theory and Applications, IVAPP 2015, Berlin, Germany, 11-14 March, 2015.*, pages 109–116, 2015.

[37] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. 3darclens: A technique for the exploration of geographical networks. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pages 245–246, Oct 2014.

[38] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. Depiction of multivariate data through flow maps. In *Proceedings of the SOMAP - Symposium on Service-Oriented Mapping*, 2014.

[39] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. Force directed flow map layout. In *Proceedings of the 5th International Conference on Information Visualization Theory and Applications (VISIGRAPP 2014)*, pages 170–177, 2014.

[40] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. A novel method for the depiction of multivariate data through flow maps. In *Poster in the IEEE Visualization conference, SciVis*, 2014.

[41] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. Supervised force directed algorithm for the generation of flow maps. In *Proceedings of the WSCG - 22nd International Conference on Computer Graphics*, pages 193–202, 2014.

[42] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. 3darclens: Interactive network analysis on geographic surfaces. In *Proceedings of the 6th International Conference on Information Visualization Theory and Applications*, pages 291–299, 2015.

[43] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. Deformation-based lens for the exploration of geo-networks. In *Poster in the EuroVis - Eurographics/IEEE Conference on Visualization*, 2015.

[44] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. Geopeels: Deformation-based technique for exploration of geo-referenced networks. In *Proceedings of the WSCG - 23rd International Conference on Computer Graphics*, pages 53–62, 2015.

[45] Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. Schematization of clutter reduction techniques in geographic node-link diagrams using task-based criteria. In *Proceedings of the 7th International Conference on Information Visualization Theory and Applications*, 2016.

[46] Alberto Debiasi, Bruno Simoes, and Raffaele De Amicis. Schematization of node-link diagrams and drawing techniques for geo-referenced networks. In *Cyberworlds (CW), 2015 International Conference on*, pages 34–41, Oct 2015.

[47] Borden D Dent. *Cartography: Thematic map design.* WC Brown Dubuque, IA, 1990.

[48] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G Tollis. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry*, 4(5):235–282, 1994.

[49] R Diestel. Graph theory, ser. *Graduate Texts in Mathematics. Springer-Verlag, Heidelberg*, 173, 2005.

[50] Martin Dodge and Rob Kitchin. *Mapping cyberspace*. Routledge London, 2001.

[51] Jonathan F Donges, Yong Zou, Norbert Marwan, and Jürgen Kurths. The backbone of the climate network. *EPL (Europhysics Letters)*, 87(4):48007, 2009.

[52] Selan Dos Santos and Ken Brodlie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28(3):311–325, 2004.

[53] Steve Dubel, Martin Rohlig, Heidrun Schumann, and Matthias Trapp. 2d and 3d presentation of spatial data: A systematic review. In *3DVis (3DVis), 2014 IEEE VIS International Workshop on*, pages 11–18, Nov 2014.

[54] Cody Dunne and Ben Shneiderman. Improving graph drawing readability by incorporating readability metrics: A software tool for network analysts. *University of Maryland, HCIL Tech Report HCIL-2009-13*, 2009.

[55] Peter Eades. A heuristics for graph drawing. *Congressus numerantium*, 42:146–160, 1984.

[56] Peter Eades, Wei Lai, Kazuo Misue, and Kozo Sugiyama. *Preserving the mental map of a diagram*. International Institute for Advanced Study of Social Information Science, Fujitsu Limited, 1991.

[57] Stephen G Eick. Aspects of network visualization. *Computer Graphics and Applications, IEEE*, 16(2):69–72, 1996.

[58] Geoffrey Ellis and Alan Dix. The plot, the clutter, the sampling and its lens: occlusion measures for automatic clutter reduction. In *Proceedings of the working conference on Advanced visual interfaces*, pages 266–269. ACM, 2006.

[59] Geoffrey Ellis and Alan Dix. A taxonomy of clutter reduction for information visualisation. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1216–1223, 2007.

[60] Niklas Elmqvist, Yann Riche, Nathalie Henry-Riche, and Jean-Daniel Fekete. Mélange: Space folding for visual exploration. *Visualization and Computer Graphics, IEEE Transactions on*, 16(3):468–483, 2010.

[61] Niklas Elmqvist and Philippas Tsigas. A taxonomy of 3d occlusion management for visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(5):1095–1109, 2008.

[62] Ozan Ersoy, Christophe Hurter, Fernando Vieira Paulovich, Gabriel Cantareiro, and Alexandru Telea. Skeleton-based edge bundling for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2364–2373, 2011.

[63] Jean-Daniel Fekete, Jarke J Van Wijk, John T Stasko, and Chris North. The value of information visualization. In *Information visualization*, pages 1–18. Springer, 2008.

[64] Stephen Few. Show me the numbers. *Analytics Pres*, 2004.

[65] Stephen Few. Save the pies for dessert. *Perceptual Edge Visual Business Intelligence Newsletter*, pages 1–14, 2007.

[66] George W Furnas. *Generalized fisheye views*, volume 17. ACM, 1986.

[67] Emden R Gansner, Yifan Hu, Stephen North, and Carlos Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 187–194. IEEE, 2011.

[68] John Gantz and David Reinsel. Extracting value from chaos. *IDC iview*, (1142):9–10, 2011.

[69] Sohaib Ghani, N Henry Riche, and Niklas Elmqvist. Dynamic insets for context-aware graph navigation. In *Computer Graphics Forum*, volume 30, pages 861–870. Wiley Online Library, 2011.

[70] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 17–24. Ieee, 2004.

[71] Nathan Gossett and Baoquan Chen. Paint inspired color mixing and compositing for visualization. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 113–118. IEEE, 2004.

[72] Jonathan Grudin. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 458–465. ACM, 2001.

[73] Diansheng Guo. Flow mapping and multivariate visualization of large spatial interaction data. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1041–1048, 2009.

[74] Diansheng Guo, Shufan Liu, and Hai Jin. A graph-based approach to vehicle trajectory analysis. *Journal of Location Based Services*, 4(3-4):183–199, 2010.

[75] Sean G Gustafson and Pourang P Irani. Comparing visualizations for tracking off-screen moving targets. In *CHI'07 Extended Abstracts on Human Factors in Computing Systems*, pages 2399–2404. ACM, 2007.

[76] Steffen Hadlak, Heidrun Schumann, and Hans-Jrg Schulz. A survey of multi-faceted graph visualization. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015.

[77] Stefan Hahmann and Dirk Burghardt. How much information is geospatially referenced? networks and cognition. *International Journal of Geographical Information Science*, 27(6):1171–1189, 2013.

[78] Stefan Hahmann, Dirk Burghardt, and Beatrix Weber. 80% of all information is geospatially referenced??? towards a research framework: Using the semantic web for (in) validating this famous geo assertion. In *Proceedings of the 14th AGILE Conference on Geographic Information Science*, 2011.

[79] Stefan Hennemann. Information-rich visualisation of dense geographical networks. *Journal of Maps*, 9(1):68–75, 2013.

[80] Nathalie Henry and Jean-Daniel Fekete. Matrixexplorer: a dual-representation system to explore social networks. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):677–684, 2006.

[81] Ivan Herman, Guy Melançon, and M Scott Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.

[82] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006.

[83] Danny Holten and Jarke J Van Wijk. Force-directed edge bundling for graph visualization. In *Computer Graphics Forum*, volume 28, pages 983–990. Wiley Online Library, 2009.

[84] Seok-Hee Hong, Damian Merrick, and Hugo AD Do Nascimento. The metro map layout problem. In *Graph Drawing*, pages 482–491. Springer, 2005.

[85] Kasper Hornbæk and Erik Frøkjær. Reading of electronic documents: the usability of linear, fisheye, and overview+ detail interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 293–300. ACM, 2001.

[86] Florian Hruby, Andreas Riedl, and Harald Tomberger. Virtual representations of antique globes–new ways of touching the untouchable. *International Journal of Digital Earth*, 1(1):107–118, 2008.

[87] Yifan Hu and Lei Shi. A coloring algorithm for disambiguating graph and map drawings. In *Graph Drawing*, pages 89–100. Springer, 2014.

[88] Weidong Huang. An eye tracking study into the effects of graph layout. *arXiv preprint arXiv:0810.4431*, 2008.

[89] Christophe Hurter, Ozan Ersoy, and Alexandru Telea. Graph bundling by kernel density estimation. In *Computer Graphics Forum*, volume 31, pages 865–874. Wiley Online Library, 2012.

[90] Christophe Hurter, Alexandru Telea, and Ozan Ersoy. Moleview: An attribute and structure-based semantic lens for large element-based plots. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2600–2609, 2011.

[91] Waqas Javed, Sohaib Ghani, and Niklas Elmqvist. Polyzoom: multiscale and multifocus exploration in 2d visual spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 287–296. ACM, 2012.

[92] Radu Jianu, Adrian Rusu, Andrew J Fabian, and David H Laidlaw. A coloring solution to the edge crossing problem. In *Information Visualisation, 2009 13th International Conference*, pages 691–696. IEEE, 2009.

[93] Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15, 1989.

[94] Michael Kaufmann and Dorothea Wagner. *Drawing graphs: methods and models*, volume 2025. Springer Science & Business Media, 2001.

[95] K Kaur, Alistair G Sutcliffe, and Neil AM Maiden. Towards a better understanding of usability problems with virtual environments. In *Proceedings of INTERACT*, volume 99, pages 527–535, 1999.

[96] Daniel A Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering the information age-solving problems with visual analytics.* Florian Mansmann, 2010.

[97] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. *Visual analytics: Scope and challenges*. Springer, 2008.

[98] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. Challenges in visual data analysis. In *Information Visualization, 2006. IV 2006. Tenth International Conference on*, pages 9–16. IEEE, 2006.

[99] Natalie Kerracher, Jessie Kennedy, and Kevin Chalmers. Tasks for temporal graph visualisation. *arXiv preprint arXiv:1402.2867*, 2014.

[100] Kurt Koffka. *Principles of Gestalt psychology*. Routledge, 2013.

[101] Robert Kosara. Presentation-oriented visualization techniques. *Computer Graphics and Applications, IEEE*, 36(1):80–85, 2016.

[102] Antoine Lambert, Romain Bourqui, and David Auber. 3d edge bundling for geographical data visualization. In *Information Visualisation (IV), 2010 14th International Conference*, pages 329–335. IEEE, 2010.

[103] Antoine Lambert, Romain Bourqui, and David Auber. Winding roads: Routing edges into bundles. In *Computer Graphics Forum*, volume 29, pages 853–862. Wiley Online Library, 2010.

[104] Ove Daae Lampe and Helwig Hauser. Interactive visualization of streaming data with kernel density estimation. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 171–178. IEEE, 2011.

[105] Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 1–5. ACM, 2006.

[106] Iosif Legrand, Harvey Newman, Ramiro Voicu, Catalin Cirstoiu, Costin Grigoras, Ciprian Dobre, Adrian Muraru, Alexandru Costan, Mihaela Dediu, and Corina Stratan. Monalisa: An agent based, dynamic service system to monitor, control and optimize distributed systems. *Computer Physics Communications*, 180(12):2472–2498, 2009.

[107] Ying K Leung and Mark D Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(2):126–160, 1994.

[108] Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, 2014.

[109] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, et al. As relationships, customer cones, and validation. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 243–256. ACM, 2013.

[110] Sheng-Jie Luo, Chun-Liang Liu, Bing-Yu Chen, and Kwan-Liu Ma. Ambiguity-free edge-bundling for interactive graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 18(5):810–821, 2012.

[111] Derek Hylton Maling. *Coordinate systems and map projections.* Elsevier, 2013.

[112] Charles Joseph Minard. *Carte figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.* Graphics Press., 1869.

[113] CJ Minard. Carte figurative et approximative des quantités de vin français exportés par mer en 1864. *Lithograph (835× 547)*, 1864.

[114] Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout adjustment and the mental map. *Journal of visual languages and computing*, 6(2):183–210, 1995.

[115] Mark Monrnonier. On the design and application of biplots in geographic visualization. *Journal of the Pennsylvania Academy of Science*, 65(1):40–47, 1991.

[116] Tomer Moscovich, Fanny Chevalier, Nathalie Henry, Emmanuel Pietriga, and Jean-Daniel Fekete. Topology-aware navigation in large networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2319–2328. ACM, 2009.

[117] Tamara Munzner. H3: Laying out large directed graphs in 3d hyperbolic space. In *Information Visualization, 1997. Proceedings., IEEE Symposium on*, pages 2–10. IEEE, 1997.

[118] Tamara Munzner. *Interactive visualization of large graphs and networks.* PhD thesis, Citeseer, 2000.

[119] Tamara Munzner. A nested model for visualization design and validation. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):921–928, 2009.

[120] Tamara Munzner. *Visualization Analysis and Design.* CRC Press, 2014.

[121] Tamara Munzner, Eric Hoffman, K Claffy, and Bill3D Geographic Network Displays Fenner. Visualizing the global topology of the mbone. In *Information Visualization'96, Proceedings IEEE Symposium on*, pages 85–92. IEEE, 1996.

[122] NASA. Nasa worldwind, 2013.

[123] Arlind Nocaj and Ulrik Brandes. Stub bundling and confluent spirals for geographic networks. In *Graph Drawing*, pages 388–399. Springer, 2013.

[124] Thomas Nocke. Basics and visual analytics of climate networks. In *Proceedings of the 2011 workshop on Climate knowledge discovery*, pages 2–2. ACM, 2011.

[125] Martin Nöllenburg. Geographic visualization. In *Human-Centered Visualization Environments*, pages 257–294. Springer, 2007.

[126] Alexandros Panagiotidis, Harald Bosch, Steffen Koch, and Thomas Ertl. Edgeanalyzer: Exploratory analysis through advanced edge interaction. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–10. IEEE, 2011.

[127] Dichao Peng, Neng Lu, Wei Chen, and Qunsheng Peng. Sideknot: Revealing relation patterns for graph visualization. In *Pacific Visualization Symposium (PacificVis), 2012 IEEE*, pages 65–72. IEEE, 2012.

[128] Vsevolod Peysakhovich, Christophe Hurter, and Alexandru Telea. Attribute-driven edge bundling for general graphs with applications in trail analysis. In *2015 IEEE Pacific Visualization Symposium, PacificVis 2015, Hangzhou, China, April 14-17, 2015*, pages 39–46, 2015.

[129] Doantam Phan, Ling Xiao, Ron Yeh, and Pat Hanrahan. Flow map layout. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 219–224. IEEE, 2005.

[130] Emmanuel Pietriga and Caroline Appert. Sigma lenses: focus-context transitions combining space, time and translucence. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1343–1352. ACM, 2008.

[131] Cyprien Pindat, Emmanuel Pietriga, Olivier Chapuis, and Claude Puech. Jellylens: Content-aware adaptive lenses. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 261–270. ACM, 2012.

[132] A Johannes Pretorius, Helen C Purchase, and John T Stasko. Tasks for multivariate network analysis. In *Multivariate Network Visualization*, pages 77–95. Springer, 2014.

[133] Sergey Pupyrev, Lev Nachmanson, Sergey Bereg, and Alexander E Holroyd. Edge routing with ordered bundles. In *Graph Drawing*, pages 136–147. Springer, 2012.

[134] Helen C Purchase, D Carrington, and J Allder. Evaluating graph drawing aesthetics: defining and exploring a new empirical research area. *Computer Graphics and Multimedia*, pages 145–178, 2004.

[135] Huamin Qu, Hong Zhou, and Yingcai Wu. Controllable and progressive edge clustering for large networks. In *Graph Drawing*, pages 399–404. Springer, 2007.

[136] Nathalie Henry Riche, Tim Dwyer, Bongshin Lee, and Sheelagh Carpendale. Exploring the design space of interactive link curvature in network diagrams. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 506–513. ACM, 2012.

[137] Jonathan C Roberts. Multiple view and multiform visualization. In *Electronic Imaging*, pages 176–185. International Society for Optics and Photonics, 2000.

[138] George Robertson, David Ebert, Stephen Eick, Daniel Keim, and Ken Joy. Scale and complexity in visual analytics. *Information Visualization*, 8(4):247–253, 2009.

[139] Arthur H. Robinson. The 1837 maps of henry drury harness. *The Geographical Journal*, 121(4):pp. 440–450, 1955.

[140] Arthur H Robinson. *Early thematic mapping in the history of cartography.* University of Chicago Press Chicago, 1982.

[141] Peter Rodgers. Graph drawing techniques for geographic visualization. *Exploring geovisualization*, pages 143–158, 2005.

[142] Ruth Rosenholtz, Yuanzhen Li, Jonathan Mansfield, and Zhenlan Jin. Feature congestion: a measure of display clutter. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 761–770. ACM, 2005.

[143] Amalia Rusu, Andrew J Fabian, Radu Jianu, and Adrian Rusu. Using the gestalt principle of closure to alleviate the edge crossing problem in graph drawings. In *Information Visualisation (IV), 2011 15th International Conference on*, pages 488–493. IEEE, 2011.

[144] Bahador Saket, Paolo Simonetto, and Stephen Kobourov. Group-level graph visualization taxonomy. *arXiv preprint arXiv:1403.7421*, 2014.

[145] Manojit Sarkar and Marc H Brown. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 83–91. ACM, 1992.

[146] Mario Schmidt. The sankey diagram in energy and material flow management. *Journal of Industrial Ecology*, 12(1):82–94, 2008.

[147] Sebastian Schmidt, Miguel A Nacenta, Raimund Dachselt, and Sheelagh Carpendale. A set of multi-touch graph interaction techniques. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 113–116. ACM, 2010.

[148] Hans-Jorg Schulz, Thomas Nocke, Magnus Heitzler, and Heidrun Schumann. A design space of visualization tasks. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2366–2375, 2013.

[149] Hans-Jorg Schulz and Heidrun Schumann. Visualizing graphs-a generalized view. In *Information Visualization, 2006. IV 2006. Tenth International Conference on*, pages 166–173. IEEE, 2006.

[150] David Selassie, Brandon Heller, and Jeffrey Heer. Divided edge bundling for directional network data. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2354–2363, 2011.

[151] Terry A Slocum, Robert Bach McMaster, Fritz C Kessler, and Hugh H Howard. *Thematic cartography and geovisualization*. Pearson Prentice Hall Upper Saddle River, NJ, 2009.

[152] John P Snyder and Philip M Voxland. An album of map projections. Technical report, 1989.

[153] Bettina Speckmann and Kevin Verbeek. Necklace maps. *IEEE Trans. Vis. Comput. Graph.*, 16(6):881–889, 2010.

[154] Stanley Smith Stevens. On the theory of scales of measurement, 1946.

[155] Guo-Dao Sun, Ying-Cai Wu, Rong-Hua Liang, and Shi-Xia Liu. A survey of visual analytics techniques and applications: State-of-the-art research and future challenges. *Journal of Computer Science and Technology*, 28(5):852–867, 2013.

[156] Masahiro Takatsuka and Mark Gahegan. Geovista studio: A codeless visual programming environment for geoscientific data analysis and visualization. *Computers & Geosciences*, 28(10):1131–1144, 2002.

[157] Roberto Tamassia. *Handbook of graph drawing and visualization*. CRC press, 2013.

[158] Alexandru Telea and Ozan Ersoy. Image-based edge bundles: Simplified visualization of large graphs. In *Computer Graphics Forum*, volume 29, pages 843–852. Wiley Online Library, 2010.

[159] James J Thomas, Kristin Cook, et al. A visual analytics agenda. *Computer Graphics and Applications, IEEE*, 26(1):10–13, 2006.

[160] Dirk Tiede and Stefan Lang. Analytical 3d views and virtual globesscientific results in a familiar spatial context. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(3):300–307, 2010.

[161] Waldo R Tobler. Experiments in migration mapping by computer. *The American Cartographer*, 14(2):155–163, 1987.

[162] Ioannis Tollis, Peter Eades, Giuseppe Di Battista, and Loannis Tollis. *Graph drawing: algorithms for the visualization of graphs*, volume 1. Prentice Hall New York, 1998.

[163] Christian Tominski, James Abello, and Heidrun Schumann. Cgvan interactive graph visualization system. *Computers & Graphics*, 33(6):660–678, 2009.

[164] Christian Tominski, James Abello, Frank Van Ham, and Heidrun Schumann. Fisheye tree views and lenses for graph visualization. In *Information Visualization, 2006. IV 2006. Tenth International Conference on*, pages 17–24. IEEE, 2006.

[165] Christian Tominski, Jonathan F Donges, and Thomas Nocke. Information visualization in climate research. In *Information Visualisation (IV), 2011 15th International Conference on*, pages 298–305. IEEE, 2011.

[166] Christian Tominski, Georg Fuchs, and Heidrun Schumann. Task-driven color coding. In *Information Visualisation, 2008. IV'08. 12th International Conference*, pages 373–380. IEEE, 2008.

[167] Christian Tominski, Stefan Gladisch, Ulrike Kister, Raimund Dachselt, and Heidrun Schumann. A survey on interactive lenses in visualization. *EuroVis State-of-the-Art Reports*, pages 43–62, 2014.

[168] Melanie Tory, Arthur E Kirkpatrick, M Stella Atkins, and Torsten Möller. Visualization task performance with 2d, 3d, and combination displays. *Visualization and Computer Graphics, IEEE Transactions on*, 12(1):2–13, 2006.

[169] Anne Treisman. Preattentive processing in vision. *Computer vision, graphics, and image processing*, 31(2):156–177, 1985.

[170] Edward R Tufte. Envisioning information. *Optometry & Vision Science*, 68(4):322–324, 1991.

[171] Edward R Tufte and PR Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.

[172] Robert Van Liere and Wim De Leeuw. Graphsplatting: Visualizing graphs as continuous fields. *Visualization and Computer Graphics, IEEE Transactions on*, 9(2):206–212, 2003.

[173] Jarke J Van Wijk. Unfolding the earth: Myriahedral projections. *The Cartographic Journal*, 45(1):32–42, 2008.

[174] Kevin Verbeek, Kevin Buchin, and Bettina Speckmann. Flow map layout via spiral trees. *IEEE transactions on visualization and computer graphics*, 17(12):2536–2544, 2011.

[175] Tatiana von Landesberger and Sebastian Bremm. A taxonomy for uncertainty in static geo-located networks. *Visually-Supported Reasoning with Uncertainty Workshop, GiScience*, 2014.

[176] Tatiana von Landesberger, Felix Brodkorb, Philipp Roskosch, Natalia Andrienko, Gennady Andrienko, and Andreas Kerren. Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *Visualization and Computer Graphics, IEEE Transactions on*, 22(1):11–20, 2016.

[177] Tatiana Von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J van Wijk, J-D Fekete, and Dieter W Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. In *Computer graphics forum*, volume 30, pages 1719–1749. Wiley Online Library, 2011.

[178] Yong Wang, Qiaomu Shen, Daniel Archambault, Zhiguang Zhou, Min Zhu, Sixiao Yang, and Huamin Qu. Ambiguityvis: Visualization of ambiguity in graph layouts. *Visualization and Computer Graphics, IEEE Transactions on*, 22(1):359–368, 2016.

[179] Michelle Q Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM, 2000.

[180] Matthew O Ward, Georges Grinstein, and Daniel Keim. *Interactive data visualization: foundations, techniques, and applications.* CRC Press, 2010.

[181] Colin Ware. *Information visualization: perception for design.* Elsevier, 2012.

[182] Martin Wattenberg. Arc diagrams: Visualizing structure in strings. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 110–116. IEEE, 2002.

[183] M Withall, Iain Phillips, and D Parish. Network visualisation: a review. *Communications, IET*, 1(3):365–372, 2007.

[184] Alexander Wolff. Drawing subway maps: A survey. *Informatik-Forschung und Entwicklung*, 22(1):23–44, 2007.

[185] Alexander Wolff. Graph drawing and cartography. In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 23, pages 697–736. CRC Press, Boca Raton, FL, 2013.

[186] Nelson Wong and Sheelagh Carpendale. Supporting interactive graph exploration with edge plucking. *Proc. VDA'07*, 2007.

[187] Nelson Wong, Sheelagh Carpendale, and Saul Greenberg. Edgelens: An interactive method for managing edge congestion in graphs. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 51–58. IEEE, 2003.

[188] Jo Wood, Jason Dykes, and Aidan Slingsby. Visualisation of origins, destinations and flows with od maps. *The Cartographic Journal*, 47(2):117–129, 2010.

[189] Kai Xu, Chris Rooney, Peter Passmore, Dong-Han Ham, and Phong H Nguyen. A user study on curved edges in graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2449–2456, 2012.

[190] Ji Soo Yi, Youn ah Kang, John T Stasko, and Julie A Jacko. Toward a deeper understanding of the role of interaction in information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1224–1231, 2007.

[191] Polle T Zellweger, Jock D Mackinlay, Lance Good, Mark Stefik, and Patrick Baudisch. City lights: contextual views in minimal space. In *CHI'03 extended abstracts on Human factors in computing systems*, pages 838–839. ACM, 2003.

[192] Hong Zhou, Panpan Xu, Xiaoru Yuan, and Huamin Qu. Edge bundling in information visualization. *Tsinghua Science and Technology*, 18(2):145–156, 2013.

[193] Hong Zhou, Xiaoru Yuan, Weiwei Cui, Huamin Qu, and Baoquan Chen. Energy-based hierarchical edge clustering of graphs. In *Visualization Symposium, 2008. PacificVIS'08. IEEE Pacific*, pages 55–61. IEEE, 2008.

[194] Marianna Zichar. Geovisualization-related issues with cognitive aspects. In *Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on*, pages 503–508. IEEE, 2013.

[195] Michael Zinsmaier, Ulrik Brandes, Oliver Deussen, and Hendrik Strobelt. Interactive level-of-detail rendering of large graphs. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2486–2495, 2012.

List of Tables

List of Figures

List of Definitions

Aesthetic criteria. “Criteria formulated as optimization goals for the drawing algorithms” [48], see Section 1.2.2.

Geographic network. “Network with associated geographical information” [33], see Section 1.1.

Geographic network visualization. “Visual representation of networks where data are laid out with respect to the physical location of the entities being presented” [183, 6], see Section 2.

Geographic node-link diagram. “Node-link diagrams that encode both relational information and geographic information”, see Section 1.3.

Geographic space. “The space of the visualization used to give a geographical reference to the visual features, mapping their position expressed in latitude and longitude to the coordinate axes of the drawing area”, see Section 2.

Global network. “Network with geographical attributes not restricted in a region of the Earth. In other words, the network spans the entire Earth”, see Section 2.

Graph Drawing (discipline). “Discipline that study the readability of graphs”, see Section 1.2.2.

Graph drawing (technique). “The process and methods for converting a graph to its geometric form” [11], see Section 1.2.2.

Graph Visualization. “It is a branch of Information Visualization that study the problem of scalability in visualizations of graphs” [120], see Section 1.2.2.

Information Visualization. “The use of computer-supported, interactive, visual representations of abstract data to amplify cognition” [29], see Section 1.2.

Node-Link diagram. “Visual representation where nodes are drawn as points and the links connecting them are drawn as lines” [49], see Section 1.2.2.

Visual Clutter. “The state in which excess items, or their representation or organization, lead to a degradation of performance at some task” [142], see Section 1.3.

Visualization. “The graphical representation of data or concepts” [97], see Section 1.2.

Visualization Pipeline. “Chain of steps and operations that describes the process of creating visual representations of data” [180], see Section 1.2.1.