Informatics

Exploring Networks Over Time and Space Utilizing Visual Analytics

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering/Internet Computing

eingereicht von

Andreas Scheidl, BSc
Matrikelnummer 00225661

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.-Prof. Mag. rer. soc. oec. Dr. rer. soc. oec. Silvia Miksch
Mitwirkung: Univ.-Ass. Roger Leite, BSc MSc

Wien, 29. April 2020

Andreas Scheidl

Silvia Miksch

Informatics

Exploring Networks Over Time and Space Utilizing Visual Analytics

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering/Internet Computing

by

Andreas Scheidl, BSc
Registration Number 00225661

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.-Prof. Mag. rer. soc. oec. Dr. rer. soc. oec. Silvia Miksch
Assistance: Univ.-Ass. Roger Leite, BSc MSc

Vienna, 29th April, 2020

Andreas Scheidl Silvia Miksch

Erklärung zur Verfassung der Arbeit

Andreas Scheidl, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 29. April 2020

Andreas Scheidl

Acknowledgements

I would like to thank my advisor Univ.-Prof. Mag. Dr. Silvia Miksch and Univ.-Ass. Roger Leite, BSc MSc at the TU Wien for their continuous support, enthusiasm, and knowledge through the process of researching and writing this thesis and the motivational words during hard times.

I also want to thank my friends, who accompanied me through my studies, for their encouragement, discussions, and an inspirational evening at the Rathaus Wien, sparking the idea to this thesis.

Finally, I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. Thank you.

Kurzfassung

Durch die voranschreitende Digitalisierung unserer Welt steht uns eine Unmenge an Daten zur Verfügung. Diese Daten erlauben es uns, immer genauere Modelle der Realität zu erstellen, die erforscht und analysiert werden können, um als Entscheidungsgrundlage für die Zukunft zu dienen. Ein ausgesprochen komplexes Konstrukt bildet hierbei das multivariante Netzwerk, das in vielzähligen Bereichen, wie z.B. den Sozialen Medien, der Telekommunikation, dem Transportwesen, der Wirtschaft oder in demografischen Daten zu finden ist. Diese Netzwerke sind oft dynamisch und verändern sich im Laufe der Zeit. Diese Veränderung erschwert die effiziente Darstellung und visuelle Analyse der ohnehin komplexen Zusammenhänge.

Das Ziel dieser Arbeit ist die Definition und Implementierung einer Visualisierung eines solchen multivarianten Netzwerks mit räumlichen und zeitlichen Veränderungen. Als Grundlage dienen die tatsächlichen Wanderungsdaten der Wiener Bürger*innen aus den Jahren 2007 bis 2018, die von der Stadt Wien, MA 23 [MA2], bereitgestellt wurden. Die Wanderungsdaten beinhalten die Wohnsitzwechsel von, nach und innerhalb von Wien.

Um eine geeignete und effiziente Visualisierung der Zusammenhänge zwischen *Raum*, *Zeit* und anderen Attributen der Wanderungsgruppen, wie *Geburtsort*, zu ermöglichen, setzen wir auf eine von Miksch et al. [MA14] beschriebene, benutzerorientierte Entwurfsmethodik. Die Zielgruppen dieser Visualisierung sind sowohl Stadtplaner*innen als auch die Öffentlichkeit. Beide Gruppen könnten Interesse daran haben, die Beziehung der Bezirke untereinander und den Migrationsstrom im Wechsel der Zeit zu verstehen.

In der Entwurfsphase betrachten wir im Speziellen die Stärken und Schwächen diverser Visualisierungstechniken, um eine ausdrucksstarke Visualisierung zu erreichen. Der räumliche Aspekt spiegelt sich in der geografischen Abbildung der Wanderungen wider. Die Art der visuellen Darstellung dieser Bewegungsströme zwischen verschiedenen Regionen ist ausschlaggebend für die Lesbarkeit der Visualisierung. Die zeitliche Veränderung wird aus drei verschiedenen Blickwinkeln beleuchtet und dargestellt. Um die verwendeten Komponenten zu integrieren und eine flexible Analyse zu ermöglichen, sind Interaktivität und Interoperabilität der Komponenten untereinander essenziell.

Der Prototyp wurde durch fünf Expert*innen im Bereich Informationsvisualisierung und einen Laien evaluiert. Die Evaluierung zeigt, dass durch die richtige Kombination vielfältiger Techniken eine Visualisierung entsteht, die den Benutzer*innen wertvolle Erkenntnisse aus den komplexen Daten liefern kann.

Abstract

The digitization of our world provides us with a vast amount of data. This data allows us to construct accurate models of real world situations which are explored and analyzed to get a deeper understanding and eventually draw conclusions for our further actions. Multivariate networks are a particularly complex construct which are ubiquitous in many different subject areas, like social media, telecommunication, transport, finance, and demographics. These networks often have a spatial context attached to them and usually evolve over time. This fact makes it even harder to efficiently visualize the many aspects of such a network.

This thesis aims to define and build a visualization of a multivariate network which changes over time and space. The underlying data network is composed of real-world movement data of citizens of Vienna from 2007 to 2018, provided by the city of Vienna, MA23 [MA2]. This data represents the change of residencies of people moving *to*, *from*, or *within* Vienna.

To tackle the complexity of the many dimensions of this data such as *time*, *space*, and other attributes, like the *country of birth* of the moving people, we follow a user-centered design approach proposed by Miksch et al [MA14]. The implemented prototype of the visualization focuses on two different user groups, which are people from the department for urban development on the one hand and the public on the other hand. Both groups may take interest in the relations between the districts and in understanding the migration flow over the years.

In the design process, we focus on strengths and weaknesses of different visualization techniques to amplify the visual expressiveness of the key aspects of the data. Spatial information is encoded in a geographic map on which flows depict movements between areas. The design choices of these flows are essential to sustain readability. The temporal aspects are depicted with different time-series visualizations. Each of them focuses on the data from a different angle. Interactivity and interoperability between these visualizations ensure determined navigation through the various aspects of the migration data.

We evaluated the visualization prototype with five experts in the field of Visual Analytics and one non-expert. The evaluation showed that the right combination of different visualization and interaction techniques results in an effective and appropriate visualization from which users can draw the desired insight.

Contents

Kurzfassung **ix**

Abstract **xi**

Contents **xiii**

1 Introduction **1**
1.1 Main Research Question . . . 3
1.2 Expected results . . . 3
1.3 Structure . . . 3

2 Related work **5**
2.1 Multivariate networks . . . 5
2.2 Origin-destination flow maps . . . 10
2.3 Time-oriented data . . . 17
2.4 Analyzing changes over time . . . 18

3 Problem definition **27**
3.1 Data . . . 28
3.2 Users . . . 29
3.3 Tasks . . . 30
3.4 Requirements . . . 31

4 Visualization Design **33**
4.1 Composition and Layout . . . 33
4.2 Data aggregation . . . 34
4.3 Main visualization component - Exploration mode . . . 35
4.4 Main visualization component - Analysis mode . . . 44
4.5 General Filtering . . . 48
4.6 Time range selection . . . 52
4.7 Selection of an area . . . 53
4.8 Interaction: Cross-highlighting, Tooltip and Zoom . . . 59
4.9 Statistics component . . . 61

5 Prototype Implementation **65**
5.1 Architecture . . . 65
5.2 Backend . . . 65
5.3 Frontend . . . 67

6 Evaluation **75**
6.1 Expert evaluation . . . 77
6.2 Example solutions . . . 81
6.3 Identified strengths . . . 92
6.4 Identified weaknesses . . . 94
6.5 Additional features . . . 95

7 Future work **97**

8 Conclusion **99**

List of Figures **101**

List of Tables **107**

Bibliography **109**

CHAPTER 1

Introduction

There is no such thing as information overload. There is only bad design.

Eduard Tufte

Open Government Data Initiatives around the world offer a vast amount of data on various topics, like environment, employment, population, economy, finance, education, and many more. Complex datasets make it possible to construct more accurate models of real world situations. These models enable people to acquire a deeper understanding of the relations between facts presented in the data. For these reasons, it is getting harder and harder to retrieve meaning or even finding important insights within this fast growing complexity and sheer amount of data.

Since visualizations support people to solve tasks in a very effective and illustrative way, the art of visualizing such complex relations of data is getting also more and more difficult. The key is to abstract the data onto a visual representation which allow users to explore, analyze, and finally understand it. Data which form a network structure are ubiquitous, they exist in social media, telecommunication, biology, finance, or software engineering, to name a few.

- What is a network?
  A data network consists of nodes which have multiple attributes of various data types, like qualitative, quantitative, temporal, or spatial data. These nodes are connected by edges which represent relations between the nodes and can also have various attributes attached to them. This leads to the definition of a multivariate network (MVN), described in detail by Kerren et. al [KPW14]. Since these networks

can be very large and also given multiple properties of the nodes or edges, it is almost impossible to visualize everything at once.

- Rising complexity by adding time dimension
  The understanding of a complex situation often surges when observed over time. To be able to analyze how the network evolved and which changes it went through, is an important tool to gain the desired insight. It is needless to say, that the the time dimension adds tremendous complexity to the task of presenting the data to the user effectively.

  The challenge when visualizing a network over time is to find a useful and efficient visualization which is able to show the user the important aspects of this change over time while providing tools to query, filter, or select detailed parts of the dataset for specialized analysis. Methods, like creating a visual data-driven story or guiding the user through specific events in time can enhance the expressiveness of the visualization.

- Migration of residents of Vienna
  The proposed visualization aims to show the migration network of people with a residency in Vienna from 2007 to 2018. Furthermore, the data shows the movement between Vienna and the rest of Austria as well as abroad. It should give insight where residents are coming from as well as where they are moving to. Vienna is divided in 23 districts which are divided again into overall 250 sub-districts. These sub-districts represent the nodes in the network and the migration flow represent the edges which connect the nodes to each other. The migration flow is defined by the number of people moving from one sub-district to another in a specific year. Additionally, the data differentiates the movement of people by country of birth so that migration patterns can be analyzed in more detail for different origins.
  The dataset on which the visualization will be based on is provided by the city of Vienna, precisely the department MA 23 (Wirtschaft, Arbeit und Statistik) [MA2]. It contains aggregated movement data of citizens of Vienna gathered by Statistik Austria [Sta] from 2007 to 2018.

The key challenge of this thesis is to find the right visualization techniques to provide insight into the complex combination of a multivariate, spatial network which changes over time.
Since *insight* is ambiguous and has a different meaning to different people, it is crucial to apply a user-centered design approach. The design process is carried out based on the three cornerstones of the design triangle [MA14], *data*, *users*, and *tasks*. By defining the users who will be operating on this visualization and defining the tasks which they will fulfill, the visualization needs to be tailored to maximize expressiveness in the given context.

1.1 Main Research Question

The main research question is

> **How can Visual Analytics support the exploration of multivariate networks over time and space?**

This leads to the main hypotheses of this thesis:

H1: User-centered design leads to an efficient visualization which enables the users to fulfill their tasks.

H2: Combining the right visualization techniques leverages *insight* into the different aspects of networks in time and space.

H3: Interactive methods in the visualization help to overcome the problems with the complexity of time-oriented multivariate networks.

1.2 Expected results

The expected outcome of this thesis is an interactive web visualization which allows a specific target group to explore and analyze the migration data in Vienna over time and space. This interactive prototype is then evaluated by experts, who assess the effectiveness and appropriateness of the chosen design.

1.3 Structure

Chapter 2 describes important concepts and taxonomies in the fields of *Multivariate network visualization* and *Time oriented data visualization*. Furthermore we describe the state of the art on different approaches to visualize Origin-Destination flow maps.

Chapter 3 defines the dominant factors of the problem statement, *data*, *users*, and *tasks*, to follow a user-centered design process. Based on this analysis, we define the functional requirements to the prototype.

Chapter 4 describes the visualization design process. In this chapter we explain the the composition of different visualizations, the interaction possibilities and the design choices we made.

Chapter 5 explains the most important implementation details and architecture of the visualization prototype.

Chapter 6 describes the evaluation process of the visualization prototype through *expert evaluation* and the results.

Chapter 7 outlines the possibilities for extending the prototype.

CHAPTER 2

Related work

In this chapter we describe the basic concepts of time-oriented multivariate networks as well as current solutions to problems which arise when visualizing this kind of data.

The research field *Information Visualization* (InfoVis) embraces techniques to support people in the analysis of visual representations of data. Card et al. [CMS99] defines it as *"The use of computer-supported, interactive, visual representations of abstract data to amplify cognition"*.

Visual Analytics (VA) on the other hand is defined by Thomas et al. [CT05] as *"The science of analytical reasoning facilitated by interactive visual interfaces"*. Furthermore, Keim et al. states that *Visual Analytics* is *"more than just visualization and can rather be seen as an integrated approach combining visualization, human factors and data analysis"* [KAF+08]. From these definitions we derive that the motivation is not only to abstract and present data from the physical world but to analyze the data at hand to help people finding *insight* by supporting their cognitive abilities. This is especially important when visualizing large and complex datasets which may overwhelm the viewer with too much information. This characteristics accurately hold true for multivariate networks.

2.1 Multivariate networks

A high degree of complexity can be found particularly in *multivariate networks* (MVNs) because of their structure, the variety of different attributes and relationships. Visualizing MVNs is a very challenging field in Visual Analytics.

2.1.1 Definition

A (simple) graph $G = (V, E)$ consists of a *finite* set of vertices (or nodes) V and a set of edges $E \subseteq \{(u, v)|u, v \in V\}$.

Such a graph, as well as its nodes and edges, can have a variety of properties. The most relevant with respect to our dataset are:

- *directed:* A directed graph (or digraph) is a graph with directed edges, i.e., (u, v) are ordered pairs of nodes.
- *undirected:* An undirected graph is a graph with unordered pairs of nodes.
- *self-loop:* An edge e = (u, v) with u = v is called a self-loop.

In addition to the above definition of a simple graph, a *multivariate network* N consists of such a graph G plus n additional attributes on edges and/or nodes.

Multivariate networks occur in many variations in the real world, e.g., in telecommunication, social media, transportation, or biology.

The work of Nobre et al. [NSML19a] offers a taxonomy to categorize different methods on multivariate network visualization. There are mainly three different layouts to visualize a network structure, as presented in figure 2.1.

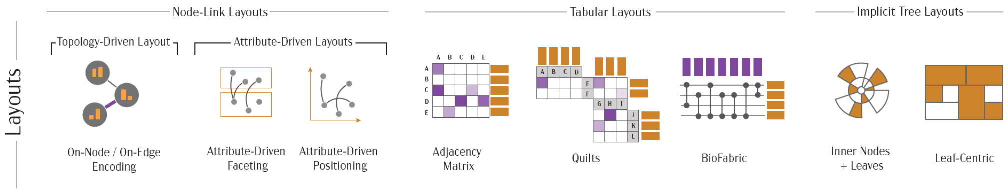


Figure 2.1: Multivariate networks. A taxonomy of different layouts to visualize multivariate networks (MVNs) [NSML19a].

2.1.2 Node-Link Layouts

The *Node-Link Layout* visualizes the network structure, like a graph, positioning nodes and connecting them visually with edges. Each sub-category of the Node-Link Layouts uses a different method of displaying data attributes in the layout.

While *On-Node/On-Edge Encoding* puts the multivariate attributes on the nodes respectively on the edges itself, the *Attribute-Driven Layouts* arrange the network layout based on positioning nodes with regards to its data attributes.

We want to emphasize in this report the *Attribute-Driven Positioning* method because the underlying dataset includes the specific data attributes *latitude* and *longitude*. Therefore, placing the nodes according to these attributes is best achieved when visualizing also the geospatial context in a geographic map.

2.1.3 Tabular Layouts

In tabular layouts, the nodes are encoded as rows and columns and the edges are depicted as cells of a table. The probably most versatile method to visualize multivariate networks is the *Adjacency matrix*. This kind of visualization is widely acknowledged and offers a lot of encoding options for the data attributes. It is also suited to analyse clusters in a big dataset. The downside of using a tabular layout is that the geospatial dimension is disregarded since e.g., the latitude and longitude can only be abstracted to a fixed grid of cells. Due to the limited space of each cell in a big matrix, having many data attributes on edges can be hard to visualize as well. We will show an example of the use of an adjacency matrix for a flow visualization later with OD maps [WDS10]. In contrast to adjacency matrices, *Quilts* can be used for visualizing layered networks. *Biofabrics* encode nodes in rows and edges in columns and offer a solution for when the network has a lot of node- as well as edge-attributes attached. For our problem description, the last two approaches do not offer advantages over the before discussed approaches.

2.1.4 Tree Layouts

Tree layouts rely on the positioning of nodes to encode edges. They excel on visualizing the tree topology but are hard to almost impossible to use if the data forms an either sparse or very dense complex network. A famous example of tree layouts are *TreeMaps* [KPW14].

2.1.5 Operations

Not only the network layout is a key factor for an effective visualization. Different techniques of combining tailored views can leverage the quest for insight even further. Nobre et al. [NSML19a] offer a taxonomy on such operations on three different levels as shown in figure 2.2.

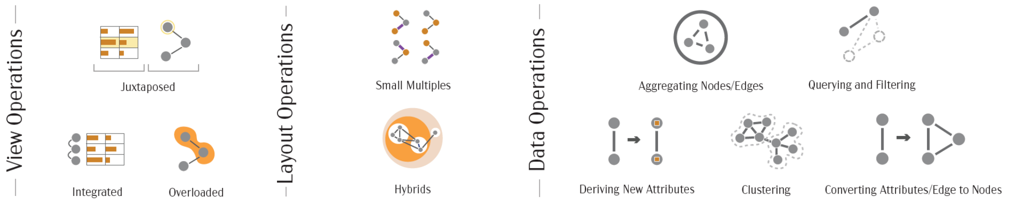


Figure 2.2: Multivariate networks. A taxonomy of operations applied to different levels in a network [NSML19a].

Combining multiple views, where each view contains a visualization which concentrates on its strengths, can improve the data exploration. In a *juxtapositioned* setup, a node-link graph could be arranged next to topology-based visualization to allow the user exploration of both features.

Layout operations combine multiple visualizations to one visual entity by either mixing them as in the *hybrid* approach, or placing the same visualization in small versions as in *small multiplies*. The latter approach will be discussed when analyzing different options to display change over time.

Data operations deal with processing data or aggregate data during analysis to gain new angles and perspectives on the network. These operations include for example deriving new attributes from available ones or aggregating nodes and edges to bigger entities, e.g., aggregating sub-districts to districts. These data operations applied during the analysis task is tightly coupled to the users interactions with the visualization.

2.1.6 Topology

Visualizing topology and change in the structural composition of a network is an important task. Nevertheless this topic is disregarded in terms of the lack of nodes appearing or disappearing over time in the particular dataset of the migration flow in Vienna. Intense studies of the structural change of a network can be found in Kerren et al. [KPW14] as well as Nobre et al. [NSML19a].

2.1.7 Interaction

It is obvious that, due to the size, structure and complexity of these networks, it is almost impossible to visualize everything at once while providing the *insight* we strive for. Therefore, an interactive approach is needed to let the user explore different angles of the data. Kerren et al. [KPW14] categorizes different interaction techniques on three levels. The interaction techniques are briefly described in the following sections with respect to our problem definition:

View-level interactions

View-level interactions deal with *highlighting* interesting data objects and *navigating* through the visualization.

Hovering describes a technique to highlight related data objects when passing over a data point of interest with the cursor. In the migration dataset, hovering over a connection between two districts could highlight the corresponding origin and destination districts.

Brushing and Linking usually involves interacting on multiple views on the same dataset. When the user selects a data point in one view, this entity is also highlighted in all other views related to this data point. This method is very useful to show the selected data in different visualization techniques, e.g., selecting a district on a geospatial map highlights the composition (attributes) of this district in a parallel-coordinates view, in our case this could be the composition of nationalities or the change over time of this connection.

Magic lenses highlight information of interest by either modifying the presentation to reveal hidden information, hide other datapoints which are not in scope of the lens or enrich selected datapoints with detailed information.

Panning and zooming is a navigation technique which enables the user to modify the viewport to enhance the perception of specific data points of interest. There is also to mention the technique *semantic zoom* which changes the amount of detail based on the zoom level. In our map example, panning and zooming could be used to zoom into a certain district, to reveal the hierarchical sub-districts and detailed information about movement within these sub-districts.

View distortion is a technique to also modify the viewport in a way that the original proportions are modified. An example of a view distortion would be a fish-eye lens which distorts the selected area more intense in the middle than on the edges of the lens.

Visual Structure-level Interactions

Selection is a technique to highlight a data point of interest and related data when selected. The difference to highlighting in the view-level is that this effect lasts also when the mouse is moving to another data point. This can be used to compare data points with the selection.

Changing mapping of attributes modifies the visual encoding of attributes. This can be used to change the *perspective*, e.g., the amount of moving people between two districts was encoded as size, by changing it to color may also change the perception of the network.

Network layouts control which aspects of the network are visible to the viewers. Changing network layouts can hide or show attributes of the nodes, focus on the topology or even be combined in multiple views to support exploration. Due to the geospatial relevance of the migration dataset, dragging nodes or links is sub-optimal since it would destroy the geographic composition of the map.

Representation is a technique to let the viewer change how parts of the network are visually represented. It may have an impact on the perception if for example selected nodes are displayed as parallel coordinates view to visualize the multivariate attributes. This technique could be used to apply different representations in place, that means that the original representation is replaced. If multiple views are used in the visualization, some of the views could represent the desired layout, while the original stays the same.

Data-level interactions

Interacting on the data level deals with which data is displayed on the visualization.

Filtering is especially useful for large networks where the scope of the data is filtered to focus on specific details. In the migration data it could be applied to filter only emigration or filtering specific nationalities to be displayed.

Dynamic querying instantly updates the visualization by using an interactive control. A prominent example is a time slider, where the viewers control the time span or period which they are interested. Such a slider could be applied to select a time span to only show the migration from 2016 to 2018.

Adding undisplayed data offers a possibility to the users to navigate through levels of the network by expanding for example neighbouring nodes on demand or drilling down the network to display underlying hierarchies, like expanding a district to its sub-districts in the migration data.

Search is especially useful if the user want to find nodes with certain attributes or properties of the network. It enables a way to navigate through complex and large networks. This could be applied to search for district names or points of interests to show where exactly in the map this area is located.

Editing the network gives the users the ability to add or delete nodes and attributes in order to support the analysis or exploration task. In our case, this will not be possible since the districts represent Vienna as an entity and deleting or adding another district would falsify the data.

Aggregation is used to combine several nodes or edges into one to simplify the network, e.g., to analyze the migration between Vienna West and Vienna East divided by the Danube.

Annotation supports the user to make notes directly in the network. This can be useful in combination with filtering or to create anchors within the exploration.

History and provenance is especially useful if the viewers are allowed to make changes to network or to support the visual exploration of large networks. The ability to fall back to prior filters, views and compositions of the visualization leverages the analysis and exploration efforts.

2.2 Origin-destination flow maps

Origin-destination flow maps are special networks which visualize movement between nodes in the network as edges. These edges usually have attributes, like direction and intensity or volume of movement. Another property of origin-destination flow maps is that the geometry of these flows is not relevant, thus has no encoded information value.

This kind of map qualifies as the main visualization technique to present the migration in Vienna, as the main attributes are the origin- and destination districts as nodes and the number of people who moved between them as edges. Furthermore, the network could be mapped onto the geographic map of Vienna to make use of the geospatial attributes in the migration dataset. It could provide a main overview of the migration in a certain time span or offer analysis in more detail with the use of interaction techniques, such as filtering, selection and zoom. The representation in an origin-destination flow map preserves size and shape of the districts, the position abstracted from the real world and can therefore, reveal patterns, like neighbourhood relations, the impression of distance of the migration and possible regional clusters. Nevertheless, this visualization cannot completely represent a multivariate network on its own since the available space is already taken by the map and its connecting edges.

Since this kind of maps can be very large with a lot of overlapping edges, the probability of visual clutter is very high. This leads to a careful design of the edges to avoid as much clutter as possible.

2.2.1 Design principles of edges in flow maps

The choice of how the edges between the nodes are visually composed has direct impact on the readability of the visualization. These edges can be displayed as straight or curved lines, with or without arrow heads to encode the direction of the flow, and the size and style of the line can be used to encode information, like the volume of the flow. Jenny et al. [JSM+18] identify *design principles* to foster readability and avoid visual clutter. Based on user studies they analyze the effectiveness of different design choices of the edges.

Straight vs. curved lines

Jenny et al. found in their study that curved lines are preferable to straight lines because the viewers had a lower error rate when answering the questions in maps with curved lines. Curved lines also offer the possibility to minimize overlaps which is an important factor when considering the readability and visual clutter. Figure 2.3 shows a flow map with curved and straight lines.

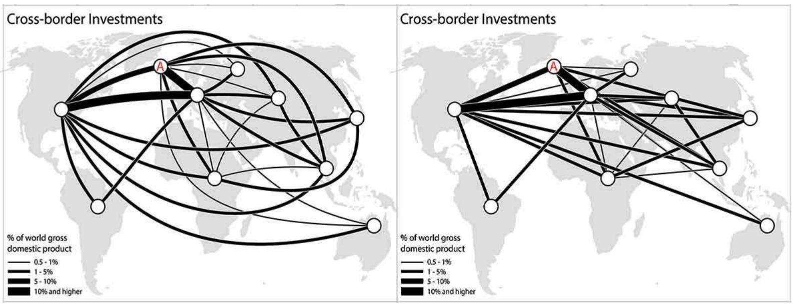


Figure 2.3: Straight vs. curved edges. Flow map shows better readability with *curved*(left) than with *straight*(right) lines [JSM+18].

Encoding direction: arrowheads vs. tapered lines

The authors advise against using tapered lines and prefer the use of arrow heads to encode direction. Figure 2.4 shows a flow map with curved and straight lines. The percentage values in the figure refer to the success rate in which viewers answered the questions correctly. While tapered lines to some extend outperform arrowheads in the studies of Holten et al. [HIvF11] they justify their advise with the following disadvantages of tapered lines:

- Long lines have a smaller gradient than short lines, therefore, they are resulting in different ambiguous gradients.

- Gradients of thin lines are hard to see.

- The direction of incoming flows is hard to be identified due to the very thin end, it could also be a very thin outgoing edge.

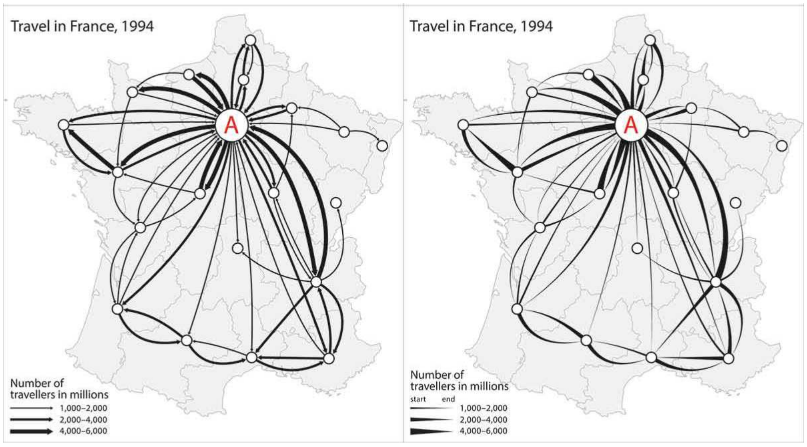


Figure 2.4: Encoding direction. The Map shows better readability with *arrowheads* (left) than with *tapered lines* (right)[JSM+18].

Flows between nodes vs. areas

Introducing a node, usually a circle in the center of the map area, which serves as an anchor for the outgoing and incoming edges showed a lower error rate in answering the questions asked by the study. Therefore, the authors advise to use these anchor-nodes to improve readability. Figure 2.5 shows flow maps with edges connecting nodes vs. areas. The percentage values in the figure refer to the success rate in which viewers answered the questions correctly.

Encoding flow intensity: width vs. color

The volume of the flow is usually encoded in the line width. While it is not necessary to additionally encode the same attribute in another way, the authors argue, that using color intensity in addition to the width can leverage the perception of dense areas and lighter areas in flow maps. Figure 2.6 shows the volume of the flows encoded by the width and color of the edges. The percentage values in the figure refer to the success rate in which viewers answered the questions correctly.

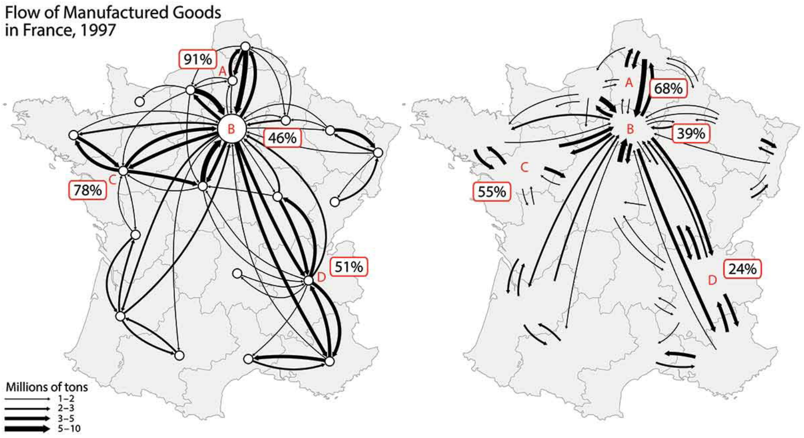


Figure 2.5: Edge termination. Flows terminating at an *anchor point* (left) and somewhere in the *areas* (right)[JSM$^+$18].

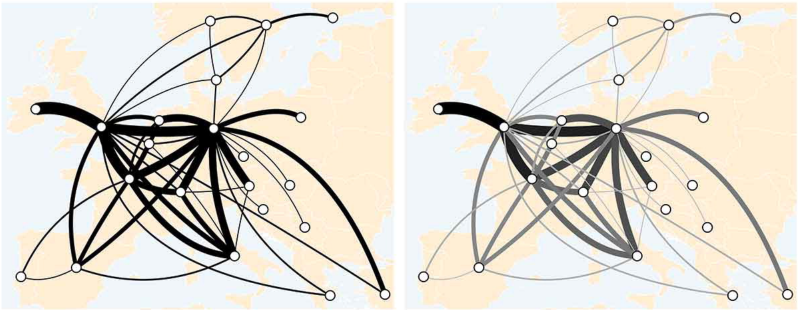


Figure 2.6: Encoding intensity. The flow volume on the left is encoded only in the *line width.* On the right, the volume is encoded with *line width and color* [JSM$^+$18].

2.2.2 Hierarchical edge bundles

Another approach to minimize visual clutter with respect to the connecting edges in a network is presented by Danny Holten [Hol06]. A large network of adjacent edges (in our case district to district movement) and hierarchical edges (in our case district-sub-district relationship) can quickly overwhelm the viewer. Especially a big amount of overlapping connections between the nodes produces a lot of visual clutter. Hierarchical edge bundling assembles adjacent edges into a bigger stream in order to visualize the hierarchical relations in the network.

Figure 2.7 shows an example of the application of hierarchical edge bundling on a software call graph and its impact on the visualization. The green part of the connection represents the *callee* and the red part the *caller* of the software components. The application of the edge bundling technique to the call graph reveals a clear distinction of the systems which are tight coupled and the loose coupled system components. The bigger margins between edge bundles reveal also less intense connections.

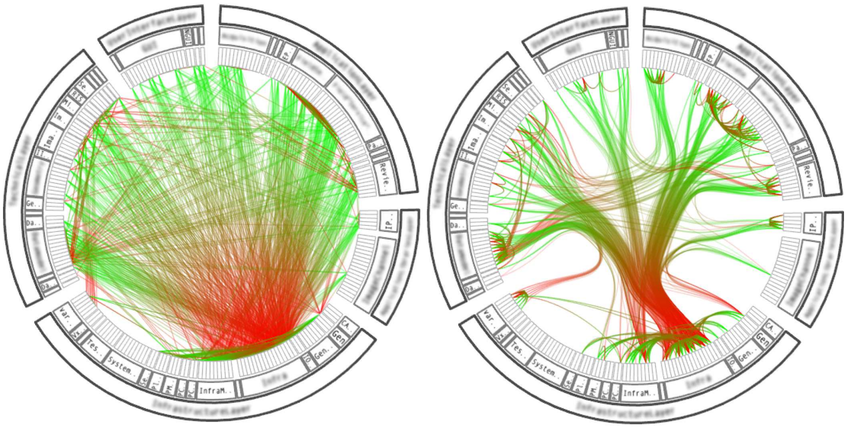


Figure 2.7: Hierarchical edge bundling. Radial network layout without bundling (left) and utilizing hierarchical edge bundling (right) [Hol06].

Additionally to the bundling of the edges, alpha blending helps the user to identify different connections. Alpha blending makes the connection less opaque so that underlying edges are more visible.

Hierarchical edge bundling is not limited to the radial network layout. As figure 2.8 shows, the hierarchical edge bundling method can also be applied to any tree layout. It is clear that edge bundling has an tremendous impact on reducing visual clutter when applying to a geographic map layout as shown in the figure 2.9, which shows an example visualization of flight connections in the United States [Fli].

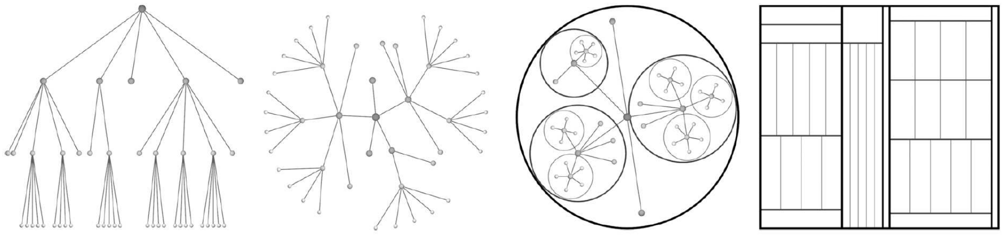


Figure 2.8: Tree layout. Common tree layouts for the use of hierarchical edge bundling. From left to right: rooted tree, radial tree, baloon tree, treemap layout [Hol06].

2.2.3 OD-Maps

Another interesting approach of visualizing origin-destination flows is OD-Maps [WDS10] presented by Wood et al. In contrast to the discussed methods, *OD-Maps* does not use the map as the main visualization component but abstracts the areas into a grid of equally sized cells. The position of the cells is approximated from the real geographic

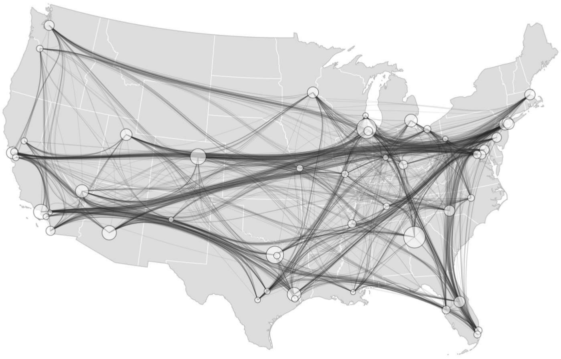


Figure 2.9: Edge bundling example. Edge bundling applied to flight connections in the United states on a geographical map [Fli].

position. Each of the cells includes the whole grid again to show the relation of the cells to each other, similar to the *small multiples* technique. The cells in the inner grid are colored according to the attribute which is chosen to be explored and analyzed.

This visualization avoids the visual clutter caused by many flow lines and therefore, offers an alternative view of very large datasets, but due to the abstraction to the grid, the exact location, size, and shape properties are lost. Figure 2.10 shows the *travel-to-work* volume between districts in the US state Ohio. Although the position is abstracted to the grid, the locality of the travel-to-work relation is clearly visible without any visual clutter introduced from connecting edges.

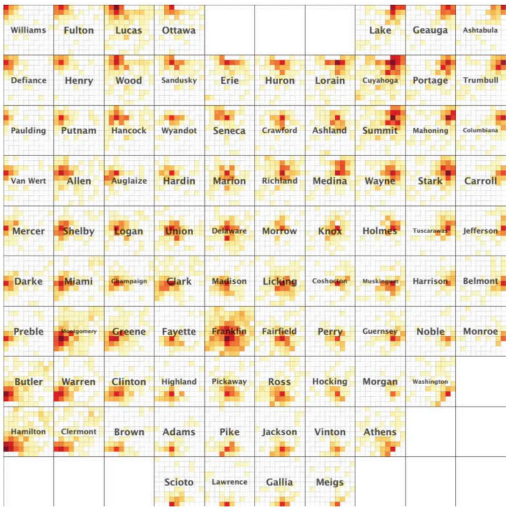


Figure 2.10: *OD-Maps.* Travel-to-work OD-map of the US-State Ohio [WDS10].

2.2.4 Origin-destination flow over time

Additional complexity is introduced if the time dimension is added to these origin-destination-flow-maps. Boyandin et al. [BBBL11] present *Flowstrates*, which is a unique interactive visualization to analyze the change over time of the flow volume. This approach uses two separate maps to display the geographic location of the origin and the destination side by side. In between these two maps, a heat map visualizes the flow volume over the given time frame for each connecting flow between selected origin and destination. Figure 2.11 shows the flow over time from selected African countries to Europe over the time span of 1983 to 2009.

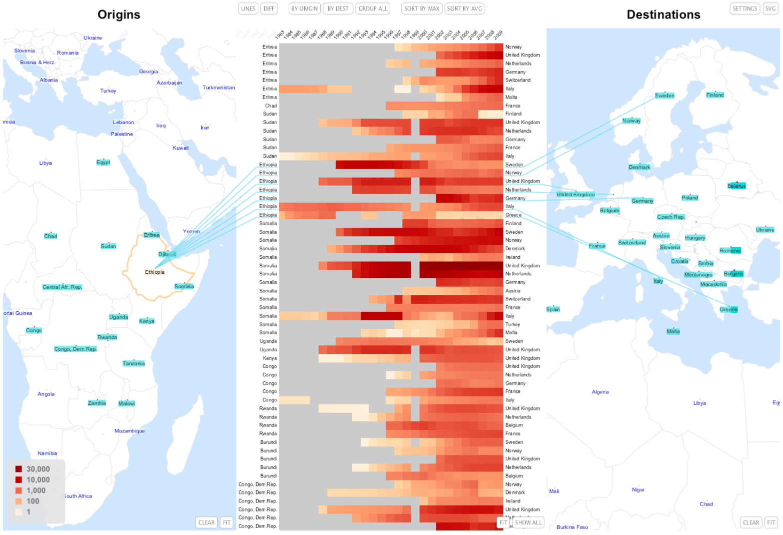


Figure 2.11: *Flowstrates*: Two maps combined with a heat map to display the change over time [BBBL11].

Flowstrates offers multiple interaction techniques to support visual exploration. The users can zoom and pan each of the three parts of the visualization to focus on details or to observe the big picture. Additionally, a selection of countries or areas can be made on the origin as well as the destination side. According to the selection, the heat map is updated to only show the flow between the selected countries. The rows in the heat map itself can be ordered and grouped according to the needs of the user.

This approach combines the advantages of origin-destination maps to preserve the geospatial attributes, like size, shape, and location of the nodes with the support of analyzing the change over time of the time-oriented data attributes. As long as there are not too many lines between the maps and the heat map, a reasonable amount of visual clutter is introduced by the edges. If the number of edges is very high, the heat map in the middle also exceeds the visible space so that the user must scroll to see all rows.

The arrangement with the two maps also complicates a distance-based analysis, because the length of the lines is not according to the actual distance of origin and destination.

Neighbourhood relations are hard to identify since the connecting edges always leave the map.

2.3 Time-oriented data

Adding the time dimension to discussed multivariate networks adds tremendous complexity to the task of supporting users to explore, analyze and understand the given data.
The challenge when visualizing a network over time is to find a useful and efficient visualization which is able to show the user the important aspects of this change over time while providing tools to query, filter or select detailed parts of the dataset for specialized analysis. Methods, like creating a visual data-driven story or guiding the user through specific events in time can enhance the expressiveness of the visualization.
Time itself is a very complex dimension. As the definition of Information Visualization states, the given data is abstracted to amplify cognition. Therefore, we also need a way to abstract time. In our perception, time has many facets whether it is a time span, a concrete point in time or a reoccurring interval. These different characteristics need to be modelled in order to create efficient visualizations tailored to the tasks the users will conduct. Aigner et al. [AMST11] provide such a characterization, which we will use as the foundation of choosing the right visualization techniques. The exact characterization of the data attributes in the migration dataset is listed in section 3.1.

2.3.1 Characterization of time

Scale (ordinal vs. discrete vs. continuous)

The scale refers to how the time attributes are given. While ordinal scale states the relation of events and their timely order (event A happened before event B), discrete and continuous scales refer to concrete points in time. Discrete points in time are modelled as integer whereas continuous points in time are considered to be abstracted to real numbers. With discrete and continuous points in time, calculations can be made, for example how much time elapsed between two timestamps.

Scope (point based vs. interval)

Point-based attributes refer to a single point in time that have no duration attached to them. Interval-based attributes on the other hand last for a given amount of time.

Arrangement (linear vs. cyclic)

The arrangement categorizes the time attributes according to their flow. In a linear time arrangement, time is abstracted to a linear scale from a starting point to some end point, usually past to future. A cyclic arrangement of time involves reoccurring entities, like seasons of a year.

Viewpoint (ordered vs. branching vs. multiple perspectives)

In an ordered domain, events happen one after the other. If no overlapping events are allowed, Aigner et al. are defining the viewpoint as totally ordered, otherwise it is a partially ordered time domain. In contrast to branching and multiple perspectives, there is only one path of occurring events. In some cases it is necessary to model simultaneous paths of occurring events. This can be done with branching, where one of the paths eventually come true (e.g., a project plan) or with multiple perspectives where multiple paths of events are possible (e.g., different eye-witness reports of an accident).

Granularity (none vs. single vs. multiple)

Abstract entities of time entail a hierarchy of time which is categorized with the granularity of the time domain. Such abstract entities can be days, weeks, fiscal months and so on. A calender uses multiple abstractions of time and is therefore, categorized as *multiple*. If no abstraction is given, e.g., simple abstract ticks of a CPU, the granularity is called *none*. In this manner, *single* refers to if only one of the abstractions is given (every time value is given in milliseconds).

Time primitives (instant vs. interval vs. span)

Instant time primitives represent fixed points in time and can be modelled in a point-based scope as well as in an interval-based scope where an *instant* can also have a duration based on the granularity. *Interval* time primitives have a start point (instant) and an end point (instant) or a time span (duration). They are modelled as closed intervals where the start and end points are included. These two time primitives are absolute primitives which always can be mapped to a point in time, whereas *span* time primitives are relative primitives. They denote a portion of time which has no start or end point, only a duration, for example *7 days*.

Determinacy (determinate vs. indeterminate)

The last characterization of the time domain is about certainty or uncertainty of the time aspects in the data. For example, a fuzzy time span (2-3 days or during the afternoon) would be an indeterminate time specification. In contrast, determinate time aspects are concrete time values, where complete knowledge of all temporal aspects are present (e.g., from 2:00 pm to 2:34 pm). Indeterminacy can also occur, when changing the time values' granularity (e.g., switching from days to hours with no exact knowledge of the hourly aspects of the event).

2.4 Analyzing changes over time

When exploring time-oriented data, it is of great interest to identify possible patterns or trends which evolve over the observed time period. The following sections give an overview on methods which can be applied to foster Visual Analytics.

There are many visualization techniques which help the user to identify trends and patterns over a time-series data. Aigner et al. [AMST11] present an excessive survey of visualization techniques for time and time-oriented data. To showcase the diversity of these visualizations, the most relevant ones with respect to our problem definitions are discussed in the next section. Additionally, *Time Curves* [BSH$^+$16] is presented as another Visual Analytics approach to identify patterns over time.

2.4.1 Visualization of time-oriented data

LinePlot and BarGraph

It seems natural to include the basic *LinePlot* and *BarGraph* in the list of time-series visualizations, since they are both well known to all kinds of viewers. In both visualizations, users can easily identify differences between the points in time and observe possible trends which might occur in the data as shown in figure 2.12. An example would be comparison of the migration volume of two districts over a period of time, where the time is represented on the x-axis and the number of people migrating on the y-axis.

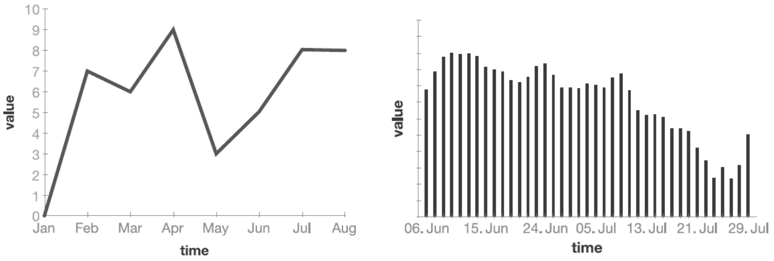


Figure 2.12: Visualizing univariate data. Examples of a LinePlot (left) and a BarGraph (right) [AMST11].

Sparklines

Visualizations of complex datasets often require a combined view of multiple techniques. *Sparklines* are used to enrich text or areas of interest by using these small visualizations as word-like components to show change over time of *univariate, linear, and instant* data attributes. Figure 2.13 shows the integration of a LinePlot and a BarGraph in order to enrich stock market data.

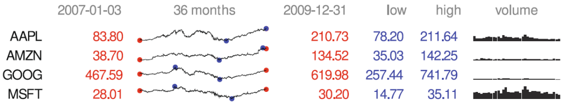


Figure 2.13: *Sparklines.* Enrichment of stock marked data by integrating small visualizations into text [AMST11].

ThemeRiver

A method of visualizing *multivariate, linear, and instant* data is presented by Havre et al in 2000. ThemeRiver [HHN00] uses a metaphor of a flowing river to visualize the relevance of topics in media over time. The width reflects changes in thematic strength of temporally associated documents. This method provides a visual narrative due to the annotations of important events and the possible correlation with the data. Figure 2.14 shows the topics in the media during December 1959 and July 1961, annotated by important events in history during this time. This method can be applied for example to visualize the rate of change of the migration of different nationalities between districts over time.

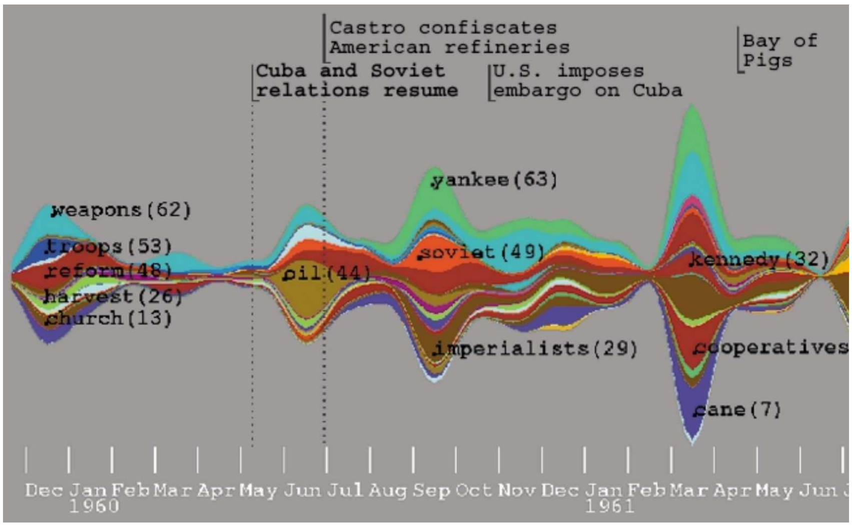


Figure 2.14: *ThemeRiver*. Multivariate data attributes visualized in form of a width-changing river. The graph shows the topic relevance in the media from 1959 to 1961 [HHN00].

Data Tube

Another approach to visualize *multivariate, linear, and instant* time-dependent attributes is called Data Tube, presented by Ankerst [Ank01]. This visualization makes use of the 3-dimensional space in which a tube is constructed. The viewer looks into the tube along the time axis, the multiple attributes are aligned along the wall of the tube. Each slice of the tube represents an *instant*. Figure 2.15 shows an illustration of Data Tube with stock market data for different stocks. Data Tube is an interactive visualization as the user can move through the tube and explore the different attributes. This visualization technique could be used in a detail view of an edge to visualize different nationalities or emigration-immigration relation over time. Since the geospatial aspect of the migration dataset is lost in this kind of representation, it would only serve as an addition to other representations.

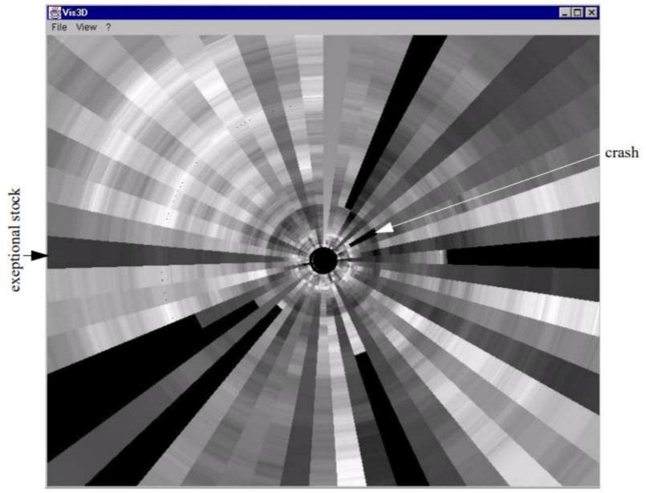


Figure 2.15: *Data Tube.* 3D visualization technique where the viewer walks along the timeline and observes the change of data attributes along the walls. This example shows stock market development [Ank01].

Trendalyzer

Another important method to visualize changes over time is called dynamic visualizations. In contrast to the before discussed static visualizations, the time dimension is abstracted to the animation of transitioning from and to static visualizations of each *instant*. The interactive visualization called Trendalyzer by the Gapminder Foundation [Gap] visualizes different datasets and its changes over time. Trendalyzer offers multiple static visualization techniques, like ScatterPlots, LinePlots, BarCharts, or Maps. Selected attributes are representing the axes and their change over time is visualized by animating the plots along the time dimension. Figure 2.16 shows a snapshot of the evolution of life expectancy relative to the income per person from 1800 to 2018. Through *Selection* the countries Austria and China were highlighted.

This dynamic representation of the temporal data could also be applied to the migration dataset to visualize the transition of edges in a map as time goes by.

Small multiples

The concept of *Small multiples* described by Edward R. Tufte [Tuf83] encodes the time dimension by stringing multiple small versions of the same static visualization together, like picture frames of a movie. Each miniature hereby presents a different point in time. An example of a small multiples arrangement of migration data is shown in figure 2.17.

Since the static visual representations of the dataset can be of each kind, small multiples are a very flexible way to present change over time. This method can be applied to every kind of time-dependent attributes. The downside of this concept is that each miniature

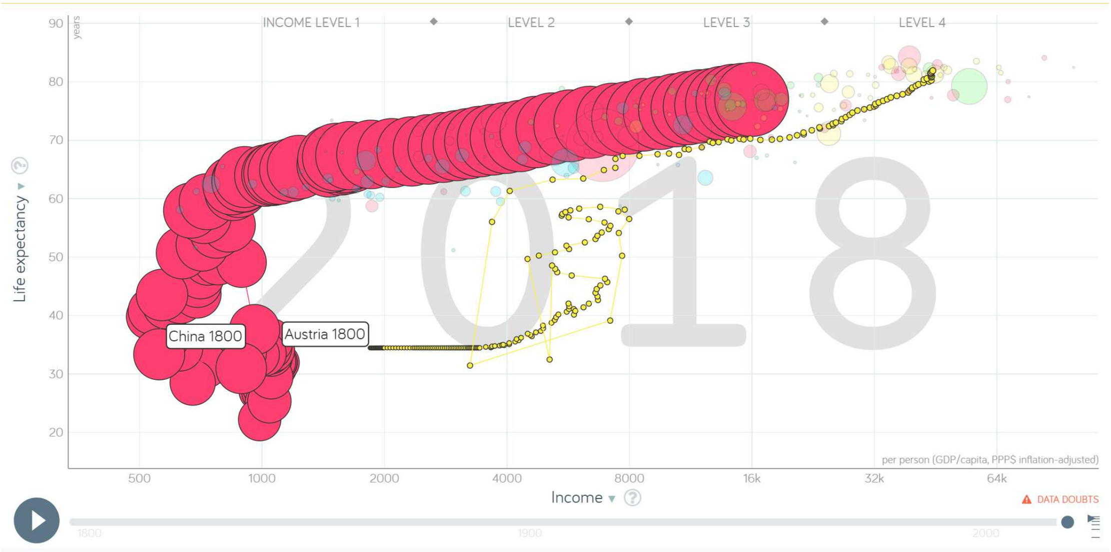


Figure 2.16: *Trendalyzer.* Dynamic visualization through animation of the life expectancy vs. income per person over time by Gapminder [Gap].

has very limited space and the number of time steps is rather limited to the overall available space. Due to the complexity and sheer amount of edges in the migration dataset, this concept has limited potential for this thesis because of the lack of detail of the miniatures.

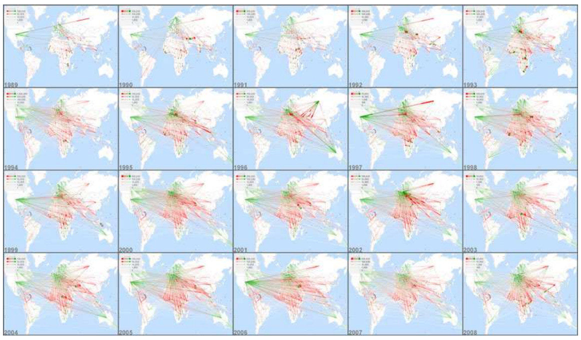


Figure 2.17: *Small multiples.* Visualizing change over time with the arrangement of miniature versions of the visualization of migration data on a map [AMST11].

Data Vases

Introduced by Thakur et al., Data Vases [TR09] represents a visualization technique which allows many time-dependent quantities to be displayed in a relatively dense space. The main concept of one data vase is a *LinePlot* which is mirrored along the time-axis and rotated by 90° counter-clockwise. An example of many data vases representing the number of crimes in counties in the U.S. state North Carolina is shown in figure 2.18. The width represents the quantitative value of the attribute of the vase, the time is represented from bottom to the top of the vase. This representation elaborates trends as well as outliers. Missing data is also represented in form of dots.

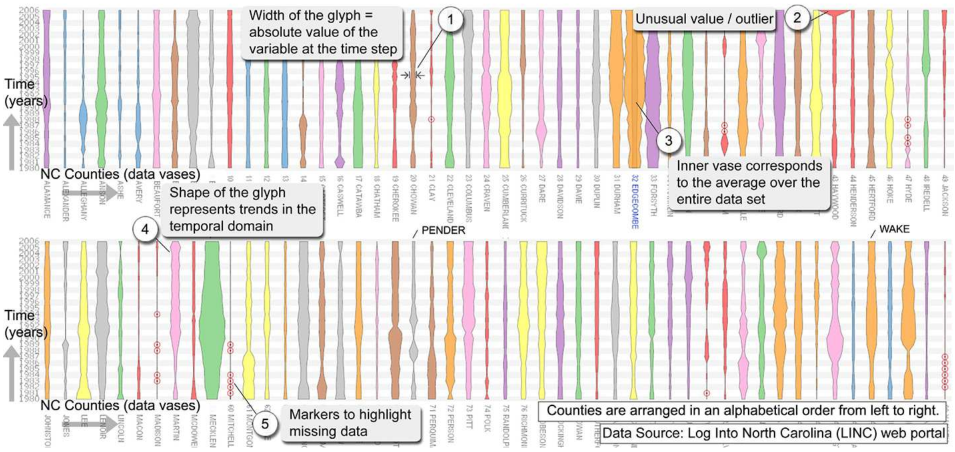


Figure 2.18: *Data Vases*. Vertical plots for each attribute show the time series data of crime rates in counties of North Carolina U.S. [TR09].

These data vases can also be expanded to the 3-dimensional space to put them onto a map, for example to locate them in a geographical area. Data vases could also be used for our problem to visualize the change over time of different nationalities migrating between two districts.

TimeCurves

Finally, analyzing multivariate data over time can be tedious, especially when facing many time-dependent attributes. Bach et al. [BSH^{+}16] describe in their paper a way to visualize and analyze patterns in time-oriented datasets.

Basically, as shown in figure 2.19, TimeCurves maps a color to each *instant* which encodes the similarity between these instants, which means that similar time-dependent variables have similar colors. The distance between two instants is depending on how much time elapsed between these instants. In proceeding steps, the straight line is folded. Similar instants are attracted to each other, so that similarity is represented by its spatial proximity. The resulting curve offers the possibility to analyze the evolution of the time-dependent attributes through the geometry of the curve.

Figure 2.20 shows a constructed time curve of the history of the Wikipedia article on Palestine. Each revision of the article represents an instant where the color of the dots represents how recent the article was revised (the darker, the more recent).

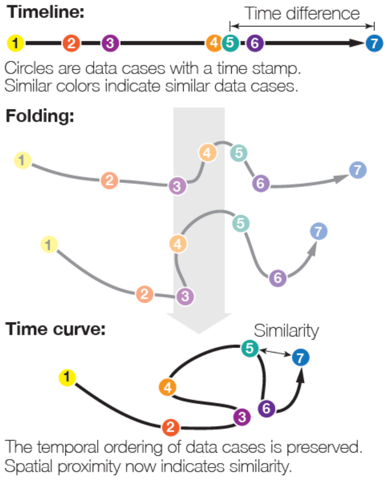

Timeline:
Time difference
1
2
3
4
5
6
7
Circles are data cases with a time stamp.
Similar colors indicate similar data cases.
Folding:
Time curve:
Similarity
The temporal ordering of data cases is preserved.
Spatial proximity now indicates similarity.

Figure 2.19: *Time curves.* Folding a timeline of data cases according to similarity. Similar cases are drawn to each other [BSH+16].

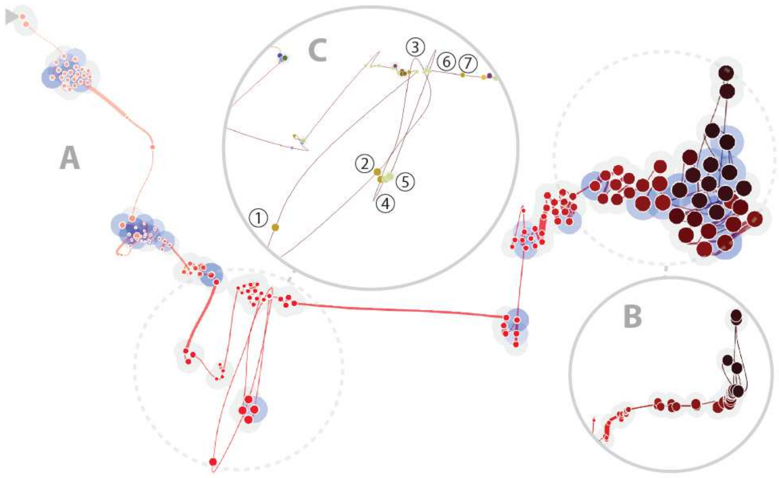


Figure 2.20: *Time Curves example.* Evolution of the Wikipedia article on Palestine with an edit war at point C [BSH+16].

In section A of the time curve, the article progressed steadily, with minor changes in the more dense areas. In section C one can see a controversy on the article as it was revised multiple times with opposing content. After finding consent, the article changed a lot, eventually coming to section B where the article reached a relatively stable version with only minor changes.

Bach et al. describe different patterns which can be identified in a time curve. Straight or smooth lines mean progression, whereas a dense cluster represents only minor changes or stagnation. The degree of oscillation describes the stability of progress, whether it alternates between very different states, for example in an edit war in Wikipedia articles, or progresses steadily with no oscillation at all. Other patterns may have a form of repetition, like cycles, short distances between dots indicate changes in a short period of time, and outliers are easily detectable since they are single points farther away from the rest.

Another advantage of TimeCurves is its domain independence. Bach et al. show a variety of domains where this method can be applied, like weather measurements, video recordings, networks, bio-science and many more. In our migration dataset, trends and patterns could be analyzed for the movement between two districts or how the migration of certain area is evolving over time.

CHAPTER 3

Problem definition

In this chapter we define the requirements for our prototype by characterizing the cornerstones of the *Data-Users-Tasks-Design triangle* described in the work of Miksch et al. [MA14] shown in figure 3.1. *Data* is referred to the question of *What* has to be visualized whereas the *Users* and *Tasks* define the question of *Why* this visualization is needed. The question of *How* we are going to present the data is discussed in the next chapter.

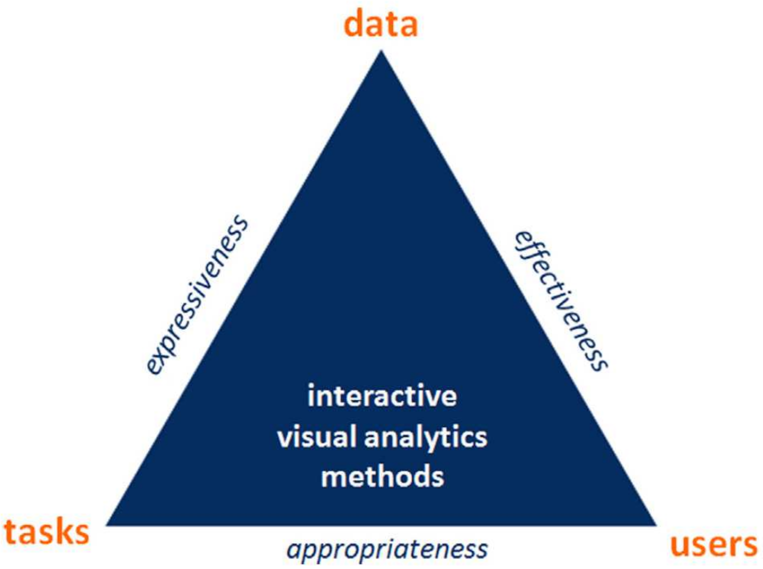


Figure 3.1: *Design triangle*: Data, Users, and Tasks as the dominant factors in the design of interactive visualizations [MA14].

3.1 Data

The dataset consists of real-world movement data of people who immigrated *to*, emigrated *from*, or changed their residency *within* Vienna from the years 2007 to 2018. The movement from and to Vienna is called *external migration*, while the movement within Vienna is called *internal migration*. The source and target regions in the movement data are divided into 250 sub-districts of Vienna and two regions for movement from/to the *rest of Austria* and from/to *abroad*. Furthermore, the movement data contains the information about the *country of birth* of the moving people.

Data structure

The city of Vienna, in particular the department MA23 [MA2], provided the dataset as follows:

The dataset consists of 194 files. Each file contains a matrix of the 250 sub-districts and the regions for the external migration. Each cell of this matrix defines the movement count for one specific year and country of birth between two regions. For example, one file contains the movements in 2010 of people born in Austria and another one contains the movements in 2013 of people born in Africa. A characterization of the data attributes is shown in table 3.1.

Attribute	Type
Year	ordinal
Origin Sub-district code (Zählbezirk)	nominal
Destination Sub-district code (Zählbezirk)	nominal
Country of birth (geopolitical entity)	nominal
Number of people who moved	ratio
Shape of each sub-district	polygon values

Table 3.1: Data attributes

The structure of the data spans a multivariate, temporal-spatial network, consisting of the following parts:

Nodes

The 250 sub-districts, the *rest of Austria*, and *abroad* represent the nodes of the network graph. Each movement has one source and one target area of these entities.

Edges

Each cell in the data matrix represents one edge of the network graph. One record contains the movement volume of people born in a particular country, who were moving from one particular region to another.

Time

The time dimension is characterized according to the taxonomy of properties defined in the work of Aigner et al. [AMST11] as shown in table 3.2.

Attribute	Type	Description
Scale	discrete	Every year has its movement data.
Arrangement	linear	The time proceeds from the past to the future, without cycles.
Viewpoints	ordered	There is only one perspective from past to future.
Time primitives	instants	Values are fixed at the end of the year.
Determinacy	determinate	We have complete knowledge of temporal attributes.

Table 3.2: Time attributes

Limitations

Due to privacy restrictions, not every country of birth is explicitly shown in the data. If the movement data of a certain country goes below the count of 2000 people a year, the corresponding movement data is aggregated to a higher geopolitical entity. For example, in 2008 only 728 people born in Afghanistan moved, so for this year the movement data is added to the file for Asia.

3.2 Users

The targeted group consists of employees of the city of Vienna respectively the department of the city development (Stadtentwicklung). These urban planners design the scheme of the city and they analyze where and how to develop new regions with parks, residential areas, as well as public transportation. They also analyze how the city evolves over time and derive possible trends.

The second group is the public which may take interest in the development of the city. This group is expected to be more focused on exploratory search.

Both user groups are assumed to be familiar with maps and how to read a map. Since every district has a geographic location and a shape, this kind of representation would be suitable. Basic line graphs and point plots are also assumed to be familiar to everyone. Other visualizations, like parallel coordinates or Time curves [BSH$^+$16] may need some introduction to the public. The application domains are mostly sociology and urban development.

3.3 Tasks

The visualization aims to enable the defined users to carry out the following tasks. These tasks are divided into high-level tasks, which reflect the motivation of the users, and more technical low-level tasks, defined by Aigner et al.[AMST11]. Furthermore, concrete scenarios are defined as questions, which the users pose to the system for specific insight. These questions are part of the evaluation criteria.

The defined tasks are valid for both of our target groups. Nevertheless, we assume that each target group has a different focus. While urban planners focus more on finding patterns and conducting a more detailed analysis of the evolution of the migration, which is also supported by their background knowledge of prior or future development projects in their field of work, the general public is assumed to be more playful and to focus only on areas of their personal interest.

High-level tasks

- Understanding the migration to and in Vienna.
- Analyzing how the migration changed over time.
- Understanding the relationship between country of birth and the migration flows.
- Evaluating the impact of certain events in the data (e.g., refugee crisis, urban development, like Seestadt Aspern)
- Analyzing the migration flows for specific attributes.
- Finding patterns (over time) in the migration flows.

Low-level tasks

During the analysis of the migration data, the users need to perform different tasks which can be defined additionally on a lower, more technical level, as the taxonomy presented in the work of Adrienko et al. [AA06] describes. The following tasks are split into *elementary* and *synoptic* tasks which need to be addressed in the prototype. *Elementary* tasks define actions on values whereas *synoptic* tasks define actions on sets which enable the analysis of patterns.

Elementary tasks:

- Lookup: How many people moved from *Donaustadt* to *Floridsdorf* in 2011?
- Comparison: Compare the immigration to and emigration from *Hernals* to *Währing* in 2018.
- Relation seeking: Did people move far away from *Landstrasse* or moved rather to the direct neighbourhood in 2009?

Synoptic tasks:

- Lookup: What is the movement trend of people born in Austria for *Leopoldstadt*?
- Patterns: Compare the movement diversity (regarding the country of birth) for *Favioriten* over time.
- Connections: Which districts are highly correlated to each other and does the country of birth influence this relation?

3.4 Requirements

The combination of the findings in defining the cornerstones of the design triangle leads to the main requirements of this prototype:

R1: District and sub-districts geographical distribution

The system shows the map of Vienna with the (sub-)districts. Since the data network contains spatial information of the sub-districts, it is obvious to facilitate the understanding of the locality of data points and relations. Therefore, it is required to visualize the data within the spatial representation of Vienna and its sub-districts. This leads to many possible advantages, like observing the *distance* of movement and *neighbourhood* relations. For example, the 12th district and the 23rd district of Vienna are adjacent districts which may be overlooked when just looking into the datapoints.

R2: Detailed view of (sub-)districts

The user is able to select a specific (sub-)district to see details. As described in the problem statement, visualizing every aspect of the data at once is not instrumental in getting insight or even not possible based on the complexity and the multidimensionality of the data. Therefore, it is required to *zoom* into the data in order to retrieve details of a selected node for specific analysis avoiding distraction by the visual clutter the overview may have.

R3: Movement

The system shows the movement data. The migration flow which is represented by the edges connecting the nodes in the network has to be presented visually. This enables the user to observe patterns of movement and the intensity of the relations between the areas.

R4: Temporal analysis

The user is able to select a specific time span. The time dimension plays a very important role in the process of understanding and analyzing the migration flow in the city and how it develops over the years. Therefore, it is required to select a specific year or a time span to show the movement data based on this criteria. This enables the user to observe the migration flow in a specific time frame and thus may correlate this data to events which occurred in the selected time frame, e.g., development of the Seestadt Aspern area in the 22nd district.

R5: Immigration rate

The system shows the immigration of a selected (sub-)district over time. Observing the immigration of a specific (sub-)district enables the user to understand from where people are moving to the selected (sub-)district. This may gain *insight* about the *composition* of a (sub-)district, e.g., from where do people move to the sub-district *Seestadt Aspern*, from districts close to the city center, from outer districts, or is there no pattern at all.

R6: Emigration rate

The system shows the emigration of a selected (sub-)district over time. Observing the emigration of a specific (sub-)district enables the user to understand where people are moving to. This may gain *insight* to certain events that may have caused intense emigration or correlation patterns with other (sub)-districts.

R7: Internal migration

The system shows the internal migration of a selected (sub-)district over time. Observing the migration flow within a selected (sub-)district enables the user to understand the stability of a (sub-)district. If there was no movement at all, this attribute may be an indicator for the satisfaction of people within a (sub-)district.

R8: Geopolitical entities

The system enables to filter by geopolitical entities. Observing the migration flow of certain *countries of birth* respectively geopolitical entities, may allow *insight* into various attributes about the (sub-)districts, like clustering or the degree of diversity. Therefore, it is required to filter by the geopolitical entities which are extracted from the data.

CHAPTER 4

Visualization Design

To fulfill the defined requirements and tasks, we chose a combination of different visualization techniques to leverage their strengths. In the following sections, the general layout design is described in detail and each of the chosen visualizations is linked to the requirements respectively tasks which they are used to satisfy.

4.1 Composition and Layout

The visualization layout is composed of various sections which fulfill a specific need to gain insight into the complex migration flows and its different data attributes. Each of these sections is part of a grid layout to arrange the different components as shown in figure 4.1.

The main part, located at the center of the screen, shows one of two available *Node-Link-Layouts*, depending on the *mode* the user has selected. In the *exploration mode*, the migration data is presented as a geographical map of Vienna. In the *analysis mode* the user can observe the relationship between districts disregarding their geographical position, displayed as a Force-Directed-Node-Link-Layout.

The section in the upper right corner of the screen serves as an information panel for the user. This section, which is from now on referred to as *Statistics component*, shows a summary of the currently displayed mode as well as statistical values about the data selection and different aspects of the data aggregation.

The left grid column represents the UI component for selecting the mode and filtering the migration data, which is handed over to the visualization components. Besides choosing the mode, the selections and filters control the granularity for showing either districts or sub-districts, the information encoding of the nodes and edges, the countries of birth of the migration data, and the number of edges which are shown in the main visualization components.

Right below the main visualization the *Timeline component* is located, which acts as selection tool for the desired time period. The user may choose a single year or a longer period ranging from 2007 to 2018.

The section beneath the Timeline Component, as well as the section in the bottom right corner of the screen, contain visualization components for exploring the movement of a selected area of interest. The *Time series data component* shows the change over time for various aspects of a selected region. It consists of three different time series visualizations in which the user can explore and analyze the changing movement intensities, countries of birth and relationships between other areas of the selected (sub-)district. The switches in the *Special filter component* control, which of these three visualizations is displayed and allow further filtering of the time series data regarding the selected (sub-)district's immigration or emigration.

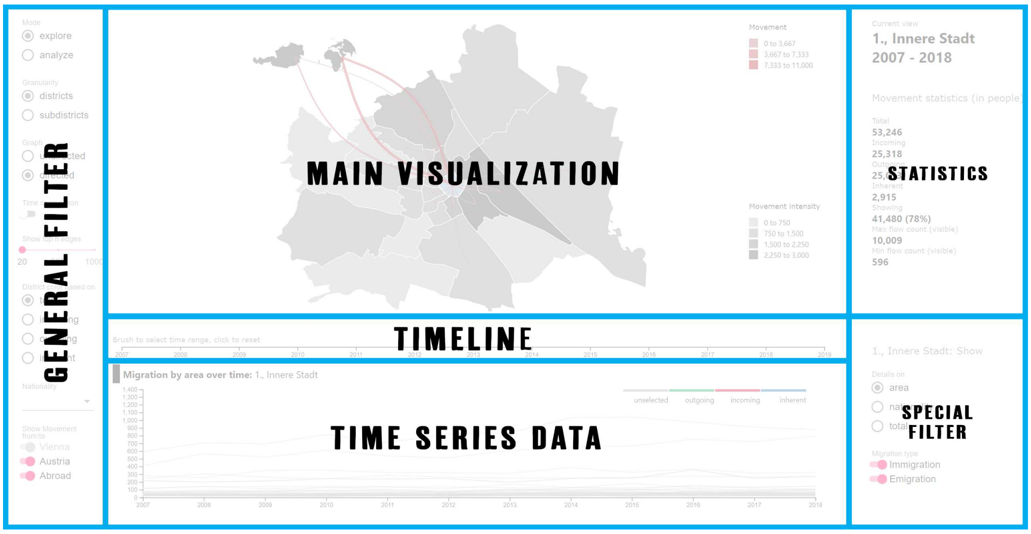


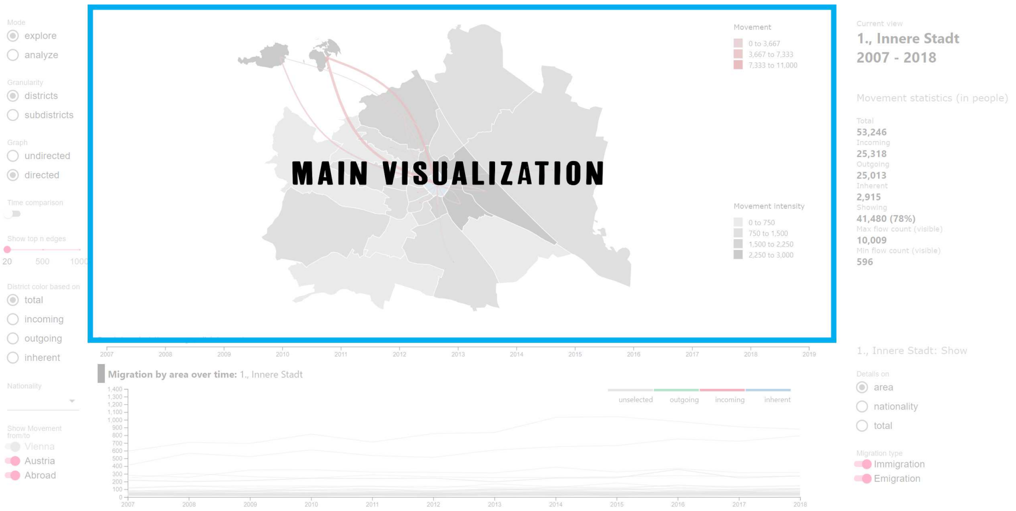
Figure 4.1: *Visualization Layout*: Arrangement of the various visualization components.

4.2 Data aggregation

To provide a visualization satisfying requirement **R1** and **R3** it is necessary to aggregate the raw data in a way that summarizes the migration flow over the desired time span and every country respectively the higher level geographical entities, like *Africa* and *Rest-EU*. This aggregation serves as the data source for the main visualization components. Additionally, movement intensities are aggregated for each (sub-)district, which is used to encode this information on the nodes. The time series data is aggregated for each year of the selected time range only if a specific region is selected in the main visualization component.

4.3 Main visualization component - Exploration mode

The main part of the visualization is located in the center of the screen as shown in figure 4.2. We chose a *Node-Link-Layout* with *Attribute-Driven Positioning*, where the positioning of the nodes is based on the geographical location of the (sub-)district. This results in a map representation of Vienna. The migration flows are depicted as lines, connecting the migration source to the migration target region. In the following sections we discuss the reasons for choosing this representation as well as design details on the nodes and edges of the migration network.



Mode
● explore
○ analyze

Granularity
● districts
○ subdistricts

Graph
○ undirected
● directed

Time comparison

Show top n edges
20 500 1000

District color based on
● total
○ incoming
○ outgoing
○ inherent

Nationality

Show Movement from/to
Vienna
Austria
Abroad

Current view
1., Innere Stadt
2007 - 2018

Movement statistics (in people)
Total
53,246
Incoming
25,318
Outgoing
25,013
Inherent
2,915
Showing
41,480 (78%)
Max flow count (visible)
10,009
Min flow count (visible)
596

1., Innere Stadt: Show

Details on
● area
○ nationality
○ total

Migration type
Immigration
Emigration

Figure 4.2: *Main visualization component*: Geographical representation of Vienna.

4.3.1 Design decisions

The geographical representation of Vienna has many advantages over e.g., a tabular representation, like an adjacency matrix. The user benefits from the geographical attributes, having the possibility to observe real distances, neighbourhood relations, and geographical clusters. Another advantage over a tabular layout is that a geographical representation is very well known to humans as we are accustomed to using maps in atlases in school, car navigation, touristic city maps or even in more abstract public transport maps. While adjacency matrices are very versatile with grouping, sorting and displaying multivariate data, it lacks the described geographical attributes which are natural for us humans to observe in maps.

Nobre et al. [NSML19b] state in their paper that the attribute-driven-positioning method is well suited for cases where the relationships between nodes are the most important criteria in the dataset, while the topology of the network has less importance. This criteria is very true for the visualization of the migration flows as the flows itself represent weighted relationships of the nodes.

The network topology on the other hand has less impact on the visualization as the districts do, if ever, change very rarely. Although topology changes seem very unlikely for Vienna, there is an ever ongoing discussion in the city government about possibly merging smaller inner-city-districts, like the 7th, 8th, and 9th or dividing larger districts, like the 21st or 22nd to simplify or to split the administration. In 2015, some Styrian districts were merged which changed the political division of Austrian districts. In our case, merging areas would only effect the districts view, as the raw data with its fine grained sub-districts would still be valid. Nevertheless, visualizing changes in the topology of the network is an important and challenging use case but out of scope in this thesis, since the topology of Vienna did not change.

4.3.2 Node design: Geographical areas

The nodes represent the districts as well as the sub-districts in the network. The underlying data attributes which determine the shape and position of the nodes are *Latitude* and *Longitude*-coordinates of districts and sub-districts.

The center of these areas define the source and target points of the edges which connect the nodes. This *Node-Link Diagram* is mapped onto the geo-spatial representation of the city of Vienna, based on the *GeoJSON* data of the areas.

The *GeoJSON* data format [Geo] is based on the JavaScript Object Notation Data Interchange Format [JSO] and specifies geographic features, their properties and the spatial extents. Listing 4.1 shows the definition of the 17th district *Hernals* in geoJSON format and figure 4.3 shows the resulting shape of this district.

The two geographic datasets for the prototype were extracted from the open data portal of Austria [OGDc].

- **Bezirksgrenzen Wien** [OGDa]: This dataset contains the geoJSON data for the 23 districts of Vienna.
- **Zählbezirksgrenzen Wien** [OGDb]: This dataset contains the geoJSON data for the 250 sub-districts of Vienna.

```
{
    "type": "Feature",
    "geometry": {
        "type": "Polygon",
        "coordinates":[[
        [16.28841069735892,48.250976316295585],
        [16.288375139016654,48.250937246517736],
        [16.288337210118243,48.25082980462866]
        [16.288292169551376,48.25071259529511],
        ...
        },
        "properties": {
            "ID": "900700",
            "NAMEK_NUM":"17., Hernals",
            "NAMEK_RZ":"XVII. Hernals",
            "NAMEG":"HERNALS",
            "LABEL":"XVII.",
            "BEZ":"17",
            "DISTRICT_CODE":1170,
            ...
        }
    }
```

Listing 4.1: Example GeoJSON entry of 17th district, *Hernals*.

Figure 4.3: geoJSON feature: Geographical shape of the 17th district, *Hernals*.

Movement from/to outside of Vienna

As we described in section 3.1, the data not only contains the migration within Vienna but also the external migration from and to the rest of Austria and abroad. *Austria* and *abroad* therefore, represent two additional nodes in the network as they are connected to the other nodes.

In contrast to the districts and sub-districts, these two special nodes, which are not part of Vienna, do not have geographical attributes. Nevertheless, they need to be placed somewhere on the visualization to include their migration flows. To deliver a consistent mental model to the user, we chose a similar representation of these nodes by depicting them in an abstract geographical shape. Figure 4.4 shows the abstract representation of the additional nodes. *Austria* is depicted by the country's shape, *abroad* is represented by the shape of the globe. This should help the user to identify the different areas accurately.

Color coding

The districts and sub-districts occupy a large amount of space in the visualization. This space is used to encode additional information about the nodes. We derive additional attributes for each node which show the movement intensity as fill color to the areas. There are four different metrics for the calculation of the movement intensity for a node a:

Figure 4.4: Special nodes: Geographical shape of the additional nodes representing the rest of Austria (left) and the rest of the world (right).

- **Incoming** represents the number of people moving from other areas to the region a. That means, the incoming metric is the sum of the movement of all edges targeting this area.
  Given an area a with
  $$a \in V$$
  , the formula to calculate the incoming movement intensity is as follows:
  $$a_{incoming} = \sum_{i=0}^{n} i_{movement}$$
  with
  $$i \in \{[u, v] \in E, v = a \land u \neq v\}$$
- **Outgoing** represents the number of people moving from region a to other areas. That means, the outgoing metric is the sum of the movement of all edges leaving this area.
  Given an area a with
  $$a \in V$$
  , the formula to calculate the outgoing movement intensity is as follows:
  $$a_{outgoing} = \sum_{i=0}^{n} i_{movement}$$
  with
  $$i \in \{[u, v] \in E, u = a \land u \neq v\}$$
- **Inherent** movement represents the number of people moving within an area a. There are two different cases on how the metric is calculated:
  - The inherent movement of a sub-district a is simply the sum of people moving from a to a.
  - The inherent movement of a district on the other hand, includes, in addition to the described movement sum above, also the movement between sub-districts which are part of the same district. For example, the inherent movement

intensity for the *10th* district also includes the movement of people between the sub-districts *10., Hauptbahnhof - Sonnwendviertel* and *10., Quellenplatz*.

Since the sub-district-data is aggregated according to the hierarchical relation to their district, the general valid formula to calculate the inherent movement metric is as follows: Given an area a with

$$a \in V$$

then

$$a_{inherent} = \sum_{i=0}^{n} i_{movement}$$

with

$$i \in \{[u, v] \in E, u = v = a\}$$

- **Total** movement is the sum of above described metrics. Given an area a with

$$a \in V$$

then the *total* movement intensity is calculated with:

$$a_{total} = a_{incoming} + a_{outgoing} + a_{inherent}$$

Figure 4.5 shows an example of the map in which the district's colors are based on the *total movement intensity*. This example shows the highest movement in *Favoriten (10th district)* as well as *Austria* and *abroad*. *Innere Stadt*, the first district and city center of Vienna, has the fewest movements, which results in the light gray color code. This view enables the user to quickly identify areas which have high or low migration volume.

4.3.3 Edge design: Movement

The choice on how to design the edges is crucial to the readability of a *Node-Link Layout*. Visual clutter should be avoided, even with many visible edges. The following sections describe the design choices made for the edges and the visual encoding of migration flow data.

Visual encoding of data

The edges encode the information of the migration flow by connecting two nodes with each other. Besides the connection itself, such a flow has additional attributes, which can be encoded visually to add value to the representation of this edge.

Direction represents the information if people were moving from node A to node B or the other way around. In contrast to an undirected edge, which has two equally shaped endpoints, an arrowhead at the target's endpoint depicts the movement direction from the source to the target area. Figure 4.9 shows an edge representing the migration flow

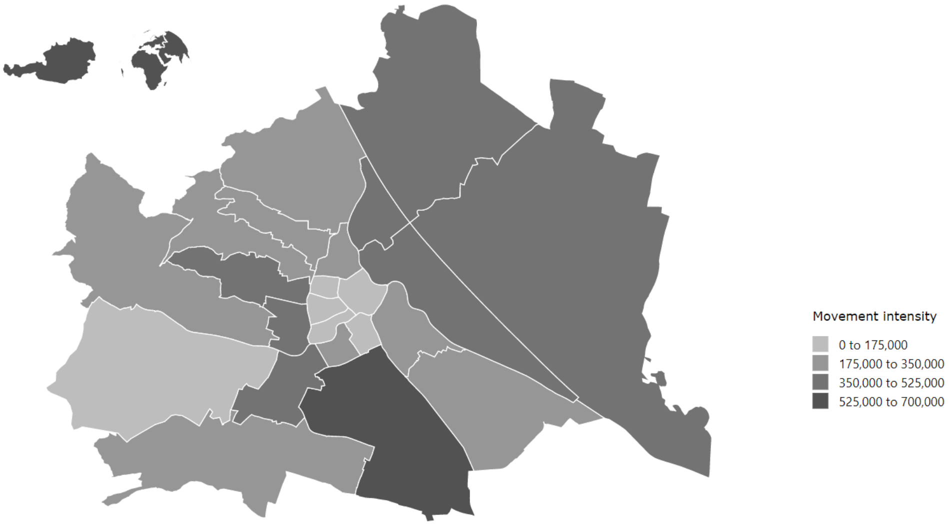


Figure 4.5: Node color coding: Districts are filled based on their total movement metric. The darker a district, the more people are moving from, to, and within the district.

of 11.142 people moving from *22., Donaustadt* to the neighbouring district *21., Floridsdorf*. The arrowhead is located in, and pointing to the target district *21., Floridsdorf*. Interaction techniques, like *Highlighting* of the connection when hovering over the edge is discussed in further detail in section 4.8.

In the migration dataset, this information is especially useful to analyze the composition of districts. It may be of interest to the user to understand where a district's people are immigrating from or whereto they are moving. Some districts may have more immigration from *abroad* while some districts may attract people equally from all other areas.

Flow volume represents the amount of people migrating from one node to another. We identified two ways of encoding the flow volume in the edge design. The first approach is to apply a different stroke width according to the flow volume, so that a bigger flow volume results in a thicker edge.

If many edges overlap each other in a very dense area, it can be difficult to distinguish the different flow amounts solely by their width. Therefore, we chose to apply an additional design element to emphasize important edges by coloring the stroke according to the flow amount. We chose three different values which are chosen by binning the flow volume into three sections. The highest third relative to the biggest amount in the data uses the most intense dark red color value. This color should reflect the intensity of the migration flow, while lower flow intensity results in a lighter, less intense red color. Additionally, the edges are drawn in ascending order regarding their flow volume. The smaller, lightweight edges are drawn first, the higher movements on top of the other ones. This ensures the visibility of important flows even in a dense area with many low flow volume edges.

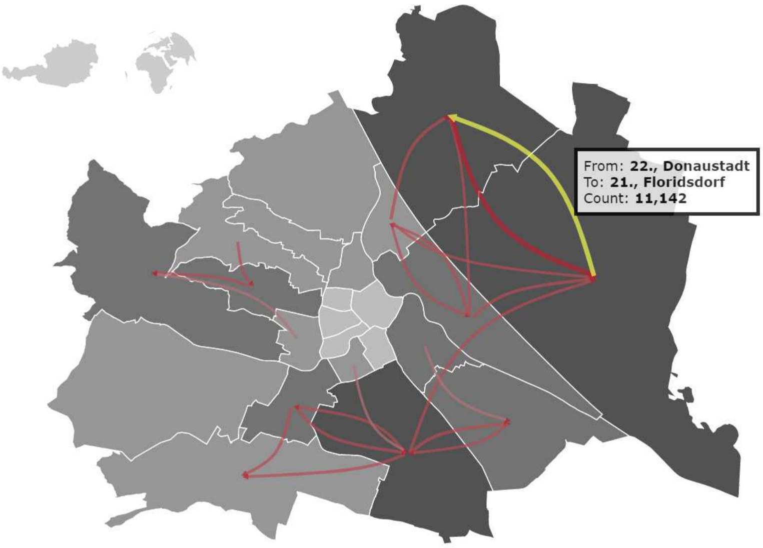


Figure 4.6: *Encoding direction*: 11.142 people moving from the 22nd to the 21st district. The edge is highlighted when hovering over with the mouse. The arrowhead displays the direction.

Curvature

As described in the paper of Jenny et al. [JSM+18] the participants of their user study had a lower error rate when analyzing maps with curved lines over straight ones. Therefore, we also decided to use curved lines to represent the edges. The amount of curvature applied to the edges is static for simplicity reasons, depending on the coordinates of the starting and the ending point.

Figure 4.7 shows a map with migration flows depicted as straight lines, in contrast to the curved lines as shown in figure 4.8. Besides the impression of a more dynamic flow of the curved edge, a curved long distance flow is visually more traceable than its straight counterpart, if the edge is intersecting another one.

Start and termination points

The exact position which an edge starts from or terminates at, is a very important visual aspect. In (sub-)districts, where a lot of edges originate or terminate, visual clutter can be too high to e.g., differentiate the edges' direction because the arrowheads may be covered by other edges. Therefore, we decided to introduce a circular buffer area around the center of each district or sub-district which serves as an offset from the center. The circle's border represents the position, on which edges are starting from or terminating at, according to the angle of the line. The additional space, especially when having edges spreading in all directions on the map, reduces the visual clutter at the center of the district as well as helps the user to distinguish different flows.

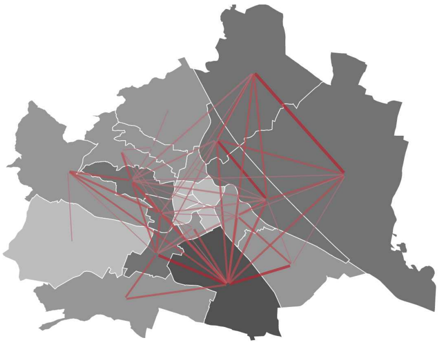


Figure 4.7: *Curvature:* Straight edges.

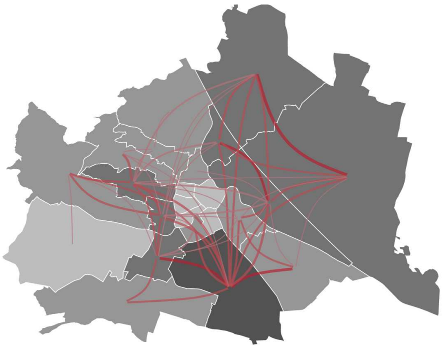


Figure 4.8: *Curvature:* Applying curved edges.

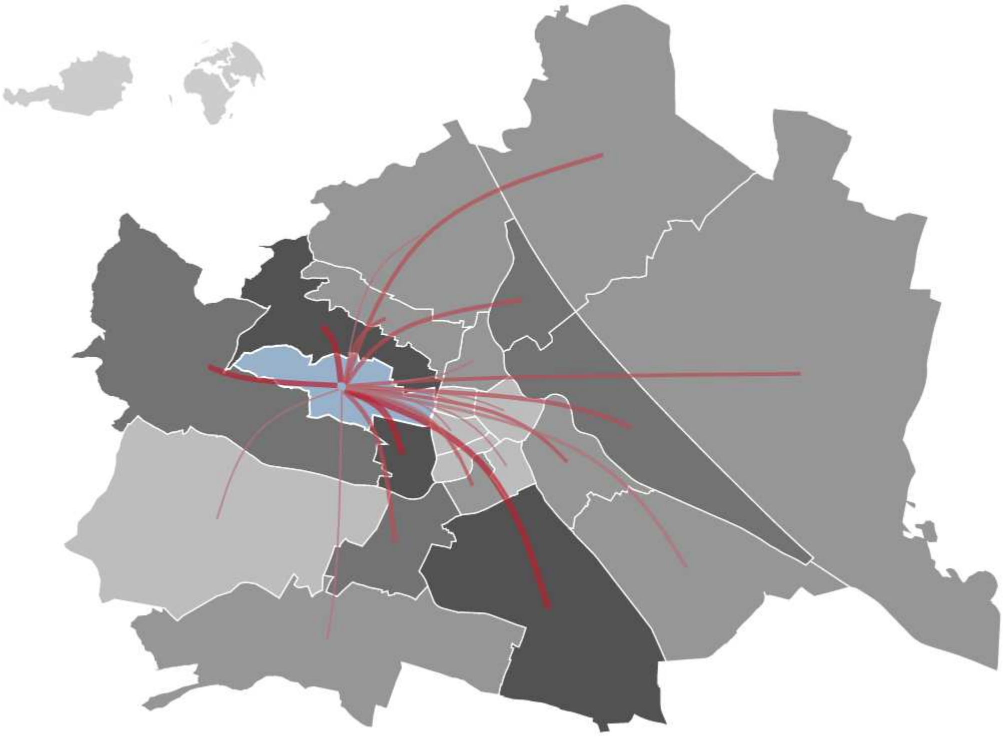


Figure 4.9: *Edge design*: Circular offset at the source and target points improve readability.

Intersections

Intersecting edges often introduce problems regarding readability and visual clutter, especially when displaying a large amount of edges. It gets more difficult to distinguish the edges from each other if they overlap. They could hide source and target points or cover the arrowhead of another edge to hide the direction.

Nevertheless, for reasons of simplicity, we renounced to introduce a collision detection or other avoiding techniques for edges and node centers, but instead reduced the opacity depending on the flow volume of each edge to make underlying lines visible. Another benefit of this effect is that dense, short areas with many overlapping lines produce a visually more intense cluster than a single line, which may be of interest to the user when looking for migration patterns.

4.3.4 Time comparison

The time comparison feature shows the direct comparison of movement intensities of each district or sub-district between two years. This feature allows the user to identify the areas which had a massive change in the movement intensity compared between the two years. Since the movement intensities are visually encoded on the nodes, the edges are hidden. The change over time of each movement metric is displayed as linear gradient on the nodes. Figure 4.10 shows the change over time of the incoming movement intensity of 2015 to 2016.

This mode, in contrast to the overview, observes only the lower and upper bound of the selected time range. This means, that instead of aggregating over every year of the selected time range, only the first and the last are compared, disregarding every movement in the years between them. E.g., given a time range selection from 2008 to 2012 would compare only the movement intensity of the areas which happened in the year 2008 to the movement intensity of the year 2012. This mode is only available in the exploration mode.

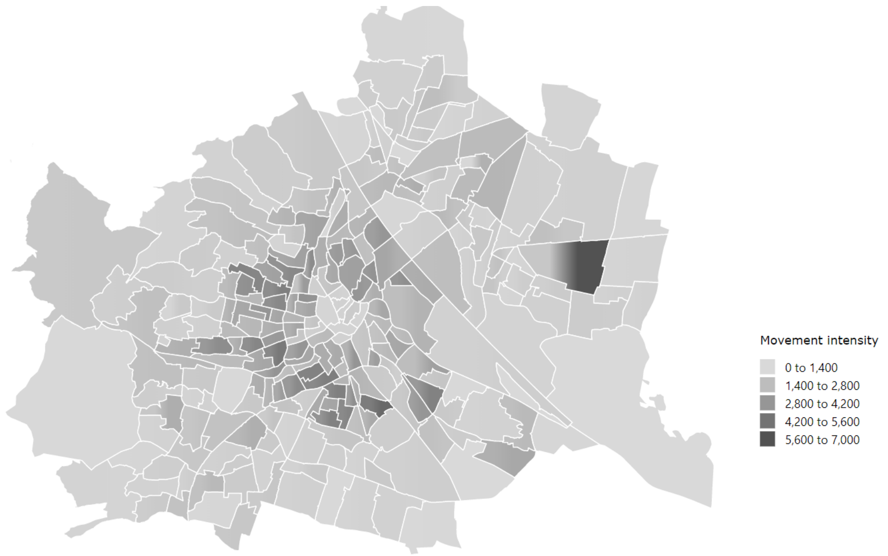


Figure 4.10: *Time comparison feature*: The incoming movement intensities for each sub-district is shown as a gradient fill from left for 2015 to right for 2016.

The node encoding of the movement intensities is based on the absolute numbers of moving people. The darker color an area is filled with, the more people moved, depending on the selection of the movement intensity (total, incoming, outgoing, or inherent). To observe the biggest changes in the movement between the selected years, the *Statistics component* on the right side shows the 15 biggest relative changes in the selected movement intensity.

4.4 Main visualization component - Analysis mode

We already discussed the problem of visual clutter in the geographical map, introduced when drawing many edges in a dense area and the decrease of readability as its consequence. Figure 4.11 shows the geographical map used in the exploration mode, showing the top 1000 migration flows in Vienna. In this example, the distinction of the different flows and their volume is almost impossible, even with a reduced opacity of the edge color. Another problem when displaying a lot of edges on this map, is that the district's color, depicting the movement intensity of the respective area, is hidden by the dense network of lines, therefore, delivering no information to the user.

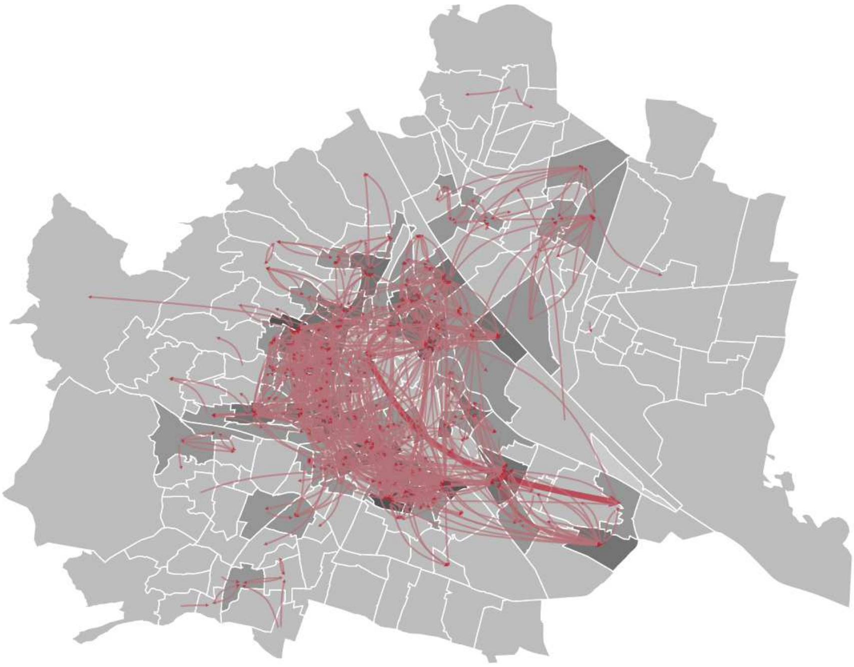


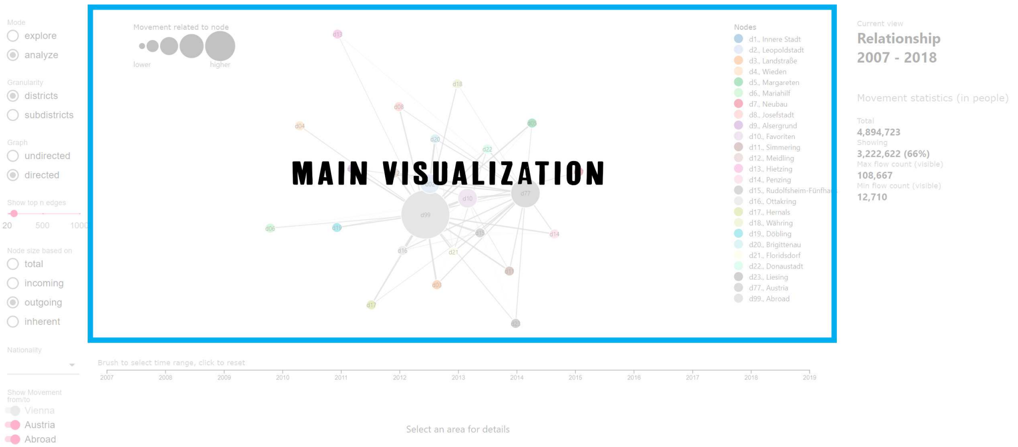
Figure 4.11: *Visual clutter problem*: Displaying many edges on the map reduces the information gain of edges and nodes.

Relations between nodes

A migration flow represents the relationship between two districts or sub-districts. The more people are moving from one area to another, the stronger this relation is. In the geographic map layout, these relations can be hard to observe under certain circumstances. e.g., for very long distance edges of equal intensity which intersect each other. In order to satisfy **R3** of the requirements defined in section 3.4, we want to visualize these relations for network as a whole.

The visualization presented in the analysis mode should display relations between nodes in the network. These relations can reveal clusters of nodes with an intense movement between them or show sparse areas for less connected nodes. Another interesting aspect in the analysis of sub-districts is the distribution and distance between nodes which are part of the same district. Long distances between such sub-district-nodes would mean that they do not correlate to each other although they are geographically very close. Close distances of these sub-district-nodes on the other hand would more or less reflect the geographic contiguity.

In contrast to the *exploration mode* described in section 4.3, the analysis mode displays the migration network in a *Force Directed Node-Link Layout*. This visualization is located in the main visualization component and replaces the map layout as shown in figure 4.12.



Mode
○ explore
● analyze

Granularity
● districts
○ subdistricts

Graph
○ undirected
● directed

Show top n edges

Node size based on
○ total
○ incoming
● outgoing
○ inherent

Nationality

Show Movement from/to
Vienna
Austria
Abroad

Current view
Relationship
2007 - 2018

Movement statistics (in people)

Total
4,894,723
Showing
3,222,622 (66%)
Max flow count (visible)
108,667
Min flow count (visible)
12,710

Brush to select time range, click to reset

Select an area for details

Figure 4.12: *Analyze mode*: The Force-Layout replaces the map layout.

4.4.1 Node positioning

Instead of positioning the nodes based on their geographic location, this graph uses *attraction* and *repulsion forces* on the nodes to determine the position in the network.

The force layout uses a *Verlet integration*, an iterative algorithm to simulate physical movement of particles. These particles in our case are the nodes in the network. Each node of the network is subject to different forces which either attract or repel other nodes. These forces blend together to determine the position of the node on each iteration. This means, that districts with many people moving between them are closer to each other than districts with a low flow or no relation at all. The most important nodes are positioned closer to the center of the visualization as they are pulling more nodes towards them than being pulled away from other ones.

4.4.2 Node design: Districts and Sub-districts

Displaying the nodes in their geographical shape, as used in the map design, is not applicable in this visualization, especially if the geographical context is removed. The areas would not be recognizable only based on their shape, thus not delivering additional information value. Instead, the nodes are simplified to circles, which are connected by the edges.

Node coloring

The fill space of the circles can be used to encode information. Since the nodes represent categorical data, we chose to color them based on a categorical color scale to be easily differentiated by the users.

Since we have 23 different districts plus the two additional special nodes for *Austria* and *abroad* in our dataset, this introduces a problem. Differentiating 25 areas simply based on their color code is very hard, because the colors would need to have a high perceptual distance to be distinguishable. The maximum number of colors usually used to encode categorical data is much lower then 25. In *Colorbrewer* [Col], a tool for selecting color palettes for maps, the maximum count of available colors is 12. Therefore, in addition to the colors and to avoid an implicit mental relation to the colors used in the other parts of the prototype, the nodes are labelled with the district number and a leading 'd'. The label for the 5th district *Margareten* is *d05*, the label for the 22nd district *Donaustadt* is *d22*. The labels for the special nodes Austria and abroad are *d77* and *d99*. Figure 4.13 shows the districts with their corresponding color and label.

Another way to distinguish nodes from each other would be to apply different textures to the nodes. The advantage of this approach is that the same color can be used multiple times, because the texture would modify the appearance of the node sufficiently. However, since the combination of colors, textures, and the labels would lower the data to ink ratio and lower the readability of the labels, we decided against this additional distinction.

The colors and labels of the districts are also applied to the 250 sub-district-nodes. Finding 252 distinguishable colors is impossible, even with the use of textures. Therefore, we chose to retain the colors, just like we did in the districts view, and to encode the hierarchy of districts and their sub-districts. Figure 4.14 shows the sub-district-nodes in the *Force-Directed-Graph*, showing the color grouping for all sub-districts being part of a district.

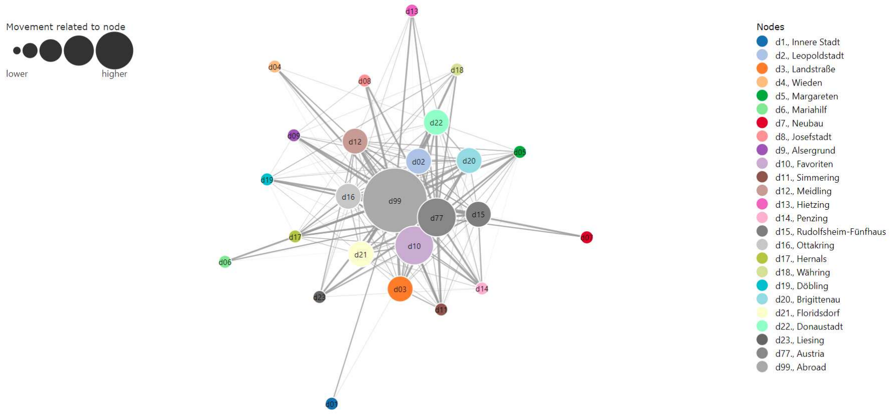


Figure 4.13: *Node Labelling and Coloring*: Color scheme and labelling for the 23 districts plus *Austria* and *abroad*.

With this *grouping* of the fine grained sub-district-nodes, the user is able to identify homogeneous or heterogeneous clusters with regards to the district they are part of. Heterogeneity would mean, that the formed clusters are mixed with nodes of different districts, whereas homogeneity maintains the districts hierarchy and its geographical relation.

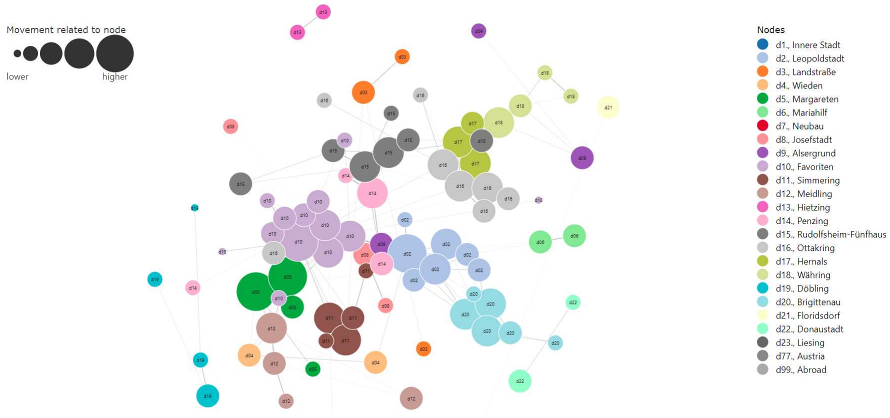


Figure 4.14: *Color coding of sub-districts*: Sub-districts of the same districts have the same color and label.

Node size

The size of the nodes in this visualization represents the movement intensity, similar to the color scale for the districts in the *exploration* mode. This means, that based on how many people are either moving out of this area, moving to this area, or moving within this area, the size of this particular node changes. The more people are related to this node, the larger is its size. Generally, the number of people related to a certain area of Vienna proportionally influences the importance of this node in the network. In figure 4.14 the legend in the upper left corner shows the scale of importance of a node. Figure 4.13 shows the node *d99 - abroad* as the most important node related to the movement intensity metric in the network.

4.4.3 Edge design: Relationship between nodes

In contrast to the map design, the edges in the force directed graph are represented as straight lines. The curvature applied to the edges in the map was chosen due to cartographic aesthetics. In the analysis mode however, the edge display has minor priority, as the edges serve primarily the calculation of the attraction forces. Nevertheless, the line width of the edges is calculated based on the flow volume. The more people were moving between two areas, the thicker the line becomes. This ensures that the information about a single high movement is not lost simply because many other smaller movements would pull the respective node in other directions.

4.4.4 Use cases

- Migration correlation between sub-districts
- Most important node for people born in Germany to immigrate
- Clustering of nodes

4.5 General Filtering

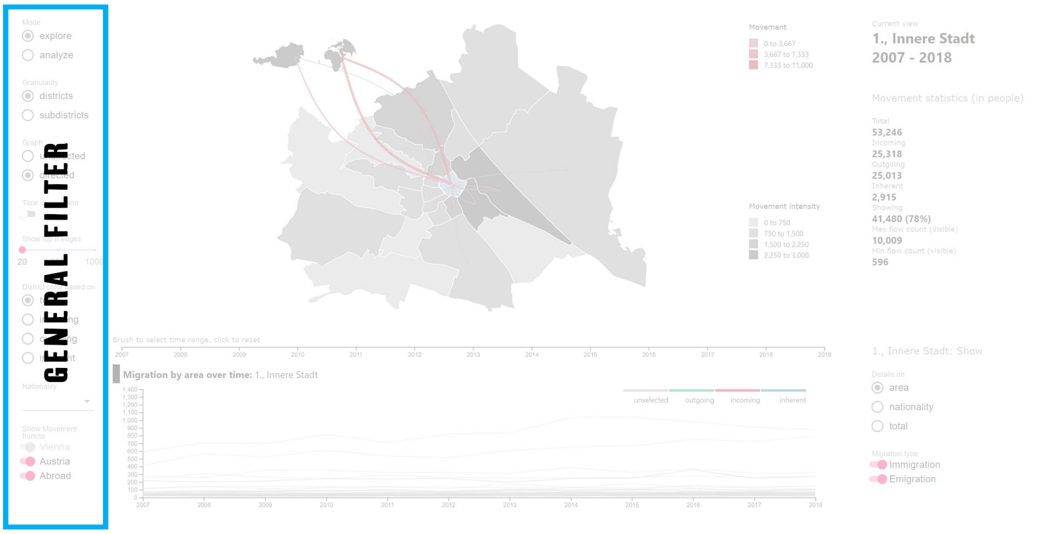
The downside of displaying the network as a node-link-layout in contrast to an adjacency matrix is, that this representation is very susceptible to visual clutter. Especially a very dense network with a large amount of edges may overwhelm the viewer.

Data-level interactions are very common to reduce the complexity of networks with multiple data attributes. Visualizing every aspect of the data at once is not feasible due to already discussed reasons. One of the most applicable approaches to letting the viewer focus on details of the data is filtering for specific values of these attributes.

This visualization component consists of common user interface controls to select, choose, and filter multiple aspects of the data and is positioned on the left side of the screen as shown in figure 4.15.

4.5.1 Mode

This selection switches between the *exploration mode* and the *analysis mode*. The exploration mode displays the main geographic map of Vienna while the analysis mode displays a *Force-Directed-Node-Link-Layout*, where the positioning of the nodes no longer



Mode
◉ explore
○ analyze

Granularity
◉ districts
○ subdistricts

Graph
○ undirected
◉ directed

Time comparison

Show top n edges
20 500 1000

District color based on
◉ total
○ incoming
○ outgoing
○ inherent

Nationality

Show Movement from/to
Vienna
Austria
Abroad

Movement
▨ 0 to 3,667
▨ 3,667 to 7,333
▨ 7,333 to 11,000

Movement intensity
▨ 0 to 750
▨ 750 to 1,500
▨ 1,500 to 2,250
▨ 2,250 to 3,000

Current view
1., Innere Stadt
2007 - 2018

Movement statistics (in people)

Total
53,246
Incoming
25,318
Outgoing
25,013
Inherent
2,915
Showing
41,480 (78%)
Max flow count (visible)
10,009
Min flow count (visible)
596

Migration by area over time: 1., Innere Stadt

1., Innere Stadt: Show

Details on
◉ area
○ nationality
○ total

Migration type
Immigration
Emigration

Figure 4.15: *General filter component*: Filtering the data is positioned in the left grid column.

depends on the geographical data attributes but on the correlation force of the nodes and its weighted edges.

4.5.2 Granularity

The granularity controls the hierarchy of the network. It enables the user to change the network based on either the districts or the fine grained sub-districts as nodes. In the districts mode, the movement data of each sub-district is aggregated to its corresponding district.

Based on the mantra of Ben Shneiderman *“Overview first, zoom, then details-on-demand”* [Shn96], zooming in from the more aggregated data of the districts down to the fine grained sub-districts data allows a detailed analysis of areas of interest.

4.5.3 Direction

The direction shows the network as undirected or directed graph. The directed graph shows two distinct edges for the movement between two areas. The movement between two districts can have a tendency of people moving only in one direction, which is an interesting fact to visualize in the network. In the undirected graph, these two edges are aggregated to one edge by adding the movement values. This aggregation can be used to leverage the display of the relationship of two districts, e.g., if two medium intense flows with lower visual impact on the network represent a strong connection when aggregated.

4.5.4 Edge Limit

Since the visual clutter is direct proportional to the number of edges displayed in the graph, a simple way to reduce this clutter is to limit the number of visible edges.

In the migration dataset, there are over 63.000 possible edges between the 252 sub-districts including the *rest of Austria* and the *rest of the world*. Displaying all 63.000 edges on this limited, very dense space makes it almost impossible to gain any insight, apart from a possible performance issue. On the other hand, hiding too much migration flows in the visualization may lead to false conclusions or even lead to wrong decisions if the visualization is used as a decision making tool.

Obviously, edges with a high migration volume are more important then edges with a very low volume. Therefore, we chose to offer a selection of the top n migration flows in descending order of the flow volume.

It may be the case that the top 20 migration flows represent e.g., 90% of the total migration for the filtered data. In this case, it is better to only show these 20 edges in order to avoid cluttering the view with hundreds of very thin edges with very low movement values. The other extreme would be if every flow was equally important corresponding to its flow amount. Hiding some of these edges would lead to misinterpretation and wrong representation of the migration network. For that reason, it is very important to provide transparency to the user. This transparency is achieved by displaying some numerical values of the presented data. Based on the filter criteria, the number of the total migration volume and the actual visible number of the migration volume should be displayed. By providing also information about the maximum flow volume and the minimum of the displayed flow volume the user should be supported in his decision on how many edges are to be drawn.

4.5.5 Time comparison

This selection activates or deactivates the *Time comparison* feature. Activating this feature will hide the edges and enables the user to compare the movement intensities for each area between the first year and the last year of the selected time range.

4.5.6 District color/size based on movement intensity

This selection controls which movement metric is used to define the *fill color* of a district in the *explore mode* or the *node size* in the *analyze mode*. The values are calculated based on the incoming, outgoing, inherent, or the total movement intensity as described in 4.3.2.

4.5.7 Country of birth

The country of birth of the moving people is a very interesting attribute to understand the migration in Vienna and the cultural relationships. The data can be filtered to show the migration network only for the citizens who were born in the selected country respectively geopolitical entity.

By focusing on this particular attribute, the user can look for patterns in the migration dataset, specific to the country of birth. This may reveal regions of Vienna which are more attractive to certain origins then others. Such cases are an important factor when planning a district's development to prevent ghettoizing or support projects to foster diversity. There are many scenarios where such insight may lead to effective planning, for example planning new schools which usually thrive for a high diversity or cultural projects to support the integration process of migrants of other cultures. From a marketing perspective, it can support the decision of where to advertise something in various branches. It can be used in the real estate market, where agents can tailor their portfolio specifically to these conditions.

In section 3.1 we described the privacy restriction for the *country of birth* attribute. If the number of moving people, who were born in a specific country, do not exceed 2.000 in one year, the country of birth is merged into the next higher geopolitical entity. This restriction introduces some inconsistencies when observed over time. We consider following example of the migration flows of people born in *Syria*: Table 4.1 shows the migration count for each year in the dataset. The data in the colored cells are falling short of the 2.000 people limit. Therefore, the movement data of Syria for the years 2007 to 2014 is aggregated to the next *geopolitical entity* provided by MA23 [MA2], which in this case is *Asia*. From 2015 on, there exist standalone movement data for people of Syrian origin.

Year	Migration count
2007	206
2008	201
2009	212
2010	241
2011	247
2012	373
2013	576
2014	1.407
2015	5.851
2016	11.351
2017	10.949
2018	9.410

Table 4.1: Migration count for Syrian people. The movement data in the colored cells are aggregated to *Asia* due to privacy restrictions.

When filtering for *Syria*, the visualization only shows data about the available years from 2015 to 2018. Filtering for the geopolitical entity *Asia* includes only the movement of people born in *Syria* in the years 2007 to 2014. This fact can lead to misinterpretation.

One approach to overcome this problem would be, to set up a hierarchy of geopolitical entities and countries. Then it would be possible to aggregate the data of all movements for a selected higher geopolitical entity and every country which is part of it. Unfortunately, this approach would introduce new problems, which is illustrated in the following example:

We consider the political entity *EU (European Union)* and the political entity *Ex-Yugoslavia*. *Croatia*, is a part of Ex-Yugoslavia and since 2013 a member of the European Union. The movement data for Croatia would be included in the entity Ex-Yugoslavia as well as in the entity EU, but only from 2013 on. Since the movement data of people born in Croatia falls into the privacy restriction, the provided data for Croatia is included only in the entity *Ex-Yugoslavia*. From there, it is impossible to query movement data from 2013 only for Croatia to aggregate it to the entity selection of EU.

We propose, for reasons of simplicity, the selection of countries and geopolitical entities as they are provided by MA23. The countries as well as the entities are treated as distinct values with no data aggregation between them.

4.5.8 Migration from/to

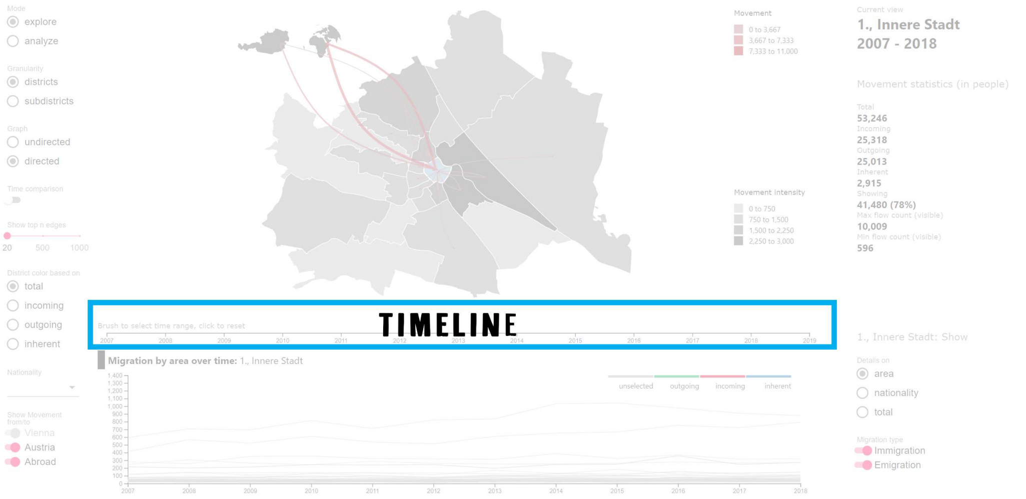
A basic analysis of the migration dataset revealed, that about 50% of the total migration in Vienna is either from or to the special nodes *Austria* and *abroad*. This selection offers the flexibility to analyze only the migration within Vienna, or to include the external migration from or to Austria and abroad. This enables the user to focus his analysis on different angles and aspects of the data while maintaining the full analytical possibilities of the visualization.

4.6 Time range selection

The timeline is located right below the main visualization component as shown in figure 4.16.

By brushing over a timeline, the user is able to select a specific year or a time range, with a start date and an end date. By default, the complete time range is selected to display the overall migration from 2007 to 2018.

In general, the migration data is aggregated over all the selected years, to be displayed in the main visualization component. When activating the *time comparison*, the time range represents the first year of the selection to be compared to the last year of the selection. For example, if the selected area on the timeline ranges from 2010 to 2014, the migration data from 2010 is compared to the data from 2014. In case that the time comparison is disabled, the migration data is aggregated for 2010, 2011, 2012, 2013, and 2014. Figure 4.17 shows the selection of the time range from 2010 to 2014. The title of the current view in the statistics component in the upper right corner displays, which time range is selected and if the years are aggregated or compared. 4.9



Mode
● explore
○ analyze

Granularity
● districts
○ subdistricts

Graph
○ undirected
● directed

Time comparison

Show top n edges
20 500 1000

District color based on
● total
○ incoming
○ outgoing
○ inherent

Nationality

Show Movement from/to
Vienna
Austria
Abroad

Movement
0 to 3,667
3,667 to 7,333
7,333 to 11,000

Movement intensity
0 to 750
750 to 1,500
1,500 to 2,250
2,250 to 3,000

Current view
1., Innere Stadt
2007 - 2018

Movement statistics (in people)

Total
53,246
Incoming
25,318
Outgoing
25,013
Inherent
2,915
Showing
41,480 (78%)
Max flow count (visible)
10,009
Min flow count (visible)
596

1., Innere Stadt: Show

Details on
● area
○ nationality
○ total

Migration type
Immigration
Emigration

Figure 4.16: *Selecting the time range*: Brushing to select a time range.

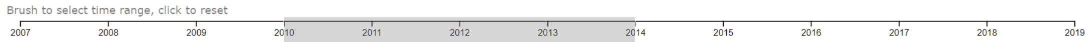


Figure 4.17: Filtering the time dimension. A brush on a timeline offers the selection of the desired time range, in this particular example 2010 - 2014.

The selection can be expanded, shortened or moved along the timeline. As the short description on top of the brush explains, the user can deselect the brush by clicking on an unselected area. This leads to the initial setting, which is the complete time range (from 2007 to 2018).

The time range selection enables the user to interactively browse through time, either by selecting one year and moving the selection year by year to the end or by slicing the data into smaller time ranges. The latter can be used to examine for example the migration before and after certain events, e.g., the world depression in 2008, the refugee crisis in 2015 and 2016, or big city development projects, like *Seestadt Aspern* from 2010 to 2017. Especially with big city development projects, which are usually long running projects, it can be very interesting to see, which milestones had an effect on the migration, e.g., the completion of public transport connection, or industry settlement in an area.

4.7 Selection of an area

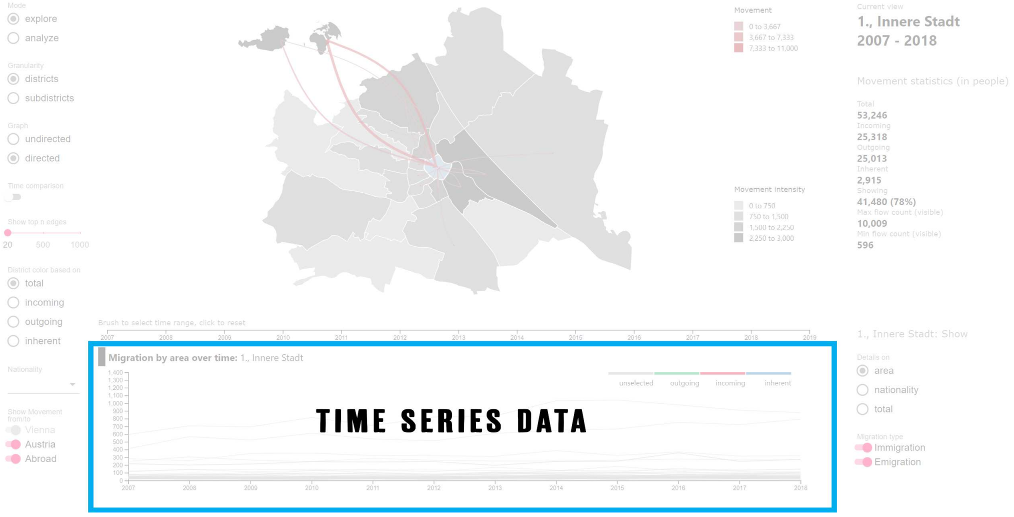
In order to fulfill requirements **R2** and **R4**, it is necessary to enable the user to *drill down* into the data by selecting a specific area of interest. By clicking on a geographic area in the map or a node in the Force-Layout, the migration data is filtered so that only movements from, to, or within the selected district or sub-district are shown.

The analysis of a single district or sub-district allows the user to explore migration patterns over time as well as the composition of a district related to the country of birth of the migrants, and to observe possible trends. By selecting for example the sub-district *Seestadt Aspern* in the 22nd district, the user is able to see where people were immigrating from when the project was completed and in which countries the people who preferred moving there were born in.

The following sections describe the additional visualizations which are available for a selected area.

4.7.1 Time-Series-Data-Component

In addition to the filtering of the movement data with regards to the selected area, special visualization components are displayed for further analysis. These visualization components are positioned in the *Time-Series-Data-Component* as shown in figure 4.18.



Mode: ● explore ○ analyze

Granularity: ● districts ○ subdistricts

Graph: ○ undirected ● directed

Time comparison

Show top n edges: 20

District color based on: ● total ○ incoming ○ outgoing ○ inherent

Nationality

Show Movement from/to: Vienna, Austria, Abroad

Current view
1., Innere Stadt
2007 - 2018

Movement statistics (in people)

Total 53,246
Incoming 25,318
Outgoing 25,013
Inherent 2,915
Showing 41,480 (78%)
Max flow count (visible) 10,009
Min flow count (visible) 596

1., Innere Stadt: Show

Details on: ● area ○ nationality ○ total

Migration type: Immigration, Emigration

Migration by area over time: 1., Innere Stadt

Figure 4.18: *Time series visualizations*: Three different visualizations let the user analyze the change over time by different aspects.

There are three types of visualizations the user can analyze:

- Migration by area over time
- Migration by country of birth over time
- Total migration over time

These visualization components do not use the aggregated data as the main visualization components do, but they display the *time-series-data* related to each time instant in the selected time range.

4.7.2 Migration by area over time

This visualization shows the temporal relation of the movement to or from the other areas as a multiple line graph. For each district, which has any migration relation with the selected area, a line is plotted along the time instants on the x-Axis, showing the change over time of the flow amount for this relation on the y-Axis. Figure 4.19 shows the line graph for the selected 10th district *Favoriten* and its migration relations with other districts over time.

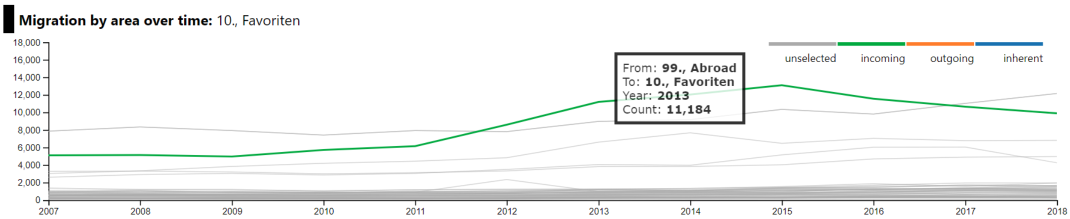


Figure 4.19: *Time series: Migration by area over time.* The line graph shows the change over time of the movement volume between *Favoriten* and other districts. Each line represents a migration relation with another district. The highlighted line shows the immigration from abroad.

Color coding

There are three different colors encoding the migration type in the time series visualizations as shown in figure 4.20.

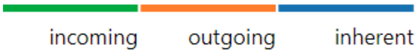


Figure 4.20: *Line coloring:* Three different colors depict the migration types: immigration (incoming), emigration (outgoing), and inherent (migration within the selected area).

Use cases

With this visualization, the user is able to observe trends in the migration data regarding the movement between the selected area and other areas. Interesting questions the user could answer with this visualization would be:

- How does the immigration from *abroad* evolve over time?
- Is there any year in which the migration from a specific district is much higher than in others?
- Is there a visible trend in the migration from/to the selected district?

- How does the migration relation between two districts change over time?

Figure 4.21 shows the migration to *Seestadt Aspern* and the sub-districts the people are moving from. The line chart clearly shows a peak in 2015 where a big milestone of the project was reached and people finally moved into this area. The user can observe from which areas the people immigrated the most. It shows, that the majority of people who moved into the Seestadt came from *the rest of Austria.* The second highest immigration in 2015 was from *abroad* and while the immigration from Austria was stagnating for the last two years, the line for abroad shows an upward trend.

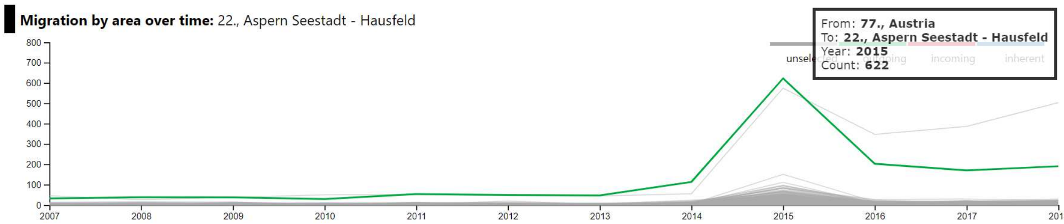


Figure 4.21: *Time series: Immigration to Seestadt Aspern.* The immigration peaked in 2015, when a milestone of the project was reached, resulting in a high immigration from Austria, stagnating from 2016 on.

4.7.3 Migration by country of birth over time

When trying to understand the migration in a certain area, it is interesting to look at the *composition* regarding the country of birth of the migrating people. This migration helps to understand the cultural diversity of an area of interest and may support the user in e.g., developing tailored integration strategies for this area.

This visualization shows a *parallel coordinates graph* to satisfy the multivariate attributes of the migration flow. The dimensions on this parallel coordinates graph are the different countries or geopolitical entities and the temporal dimension. Figure 4.22 shows the composition of the 6th district *Mariahilf* over time, filtering only the immigration data. The visualization shows, that the majority of the immigrants in Mariahilf were born in Austria, in Germany, or in the Rest of the European Union. In contrast to the 6th district, the same graph for the 20th district *Brigittenau*, shows an almost equally high immigration of people who were born in *Asia* in 2015, which could be connected to the refugee crises. The composition overall clearly differs from the one shown in figure 4.23, having more immigrants born in *Ex-Yugoslavia, Serbia, Afghanistan, Syria*, and so on.

Use cases

Various user groups can benefit from the analysis of a region's composition over time. It is interesting to explore, if certain cultures are more attracted to particular areas than others. It may also lead to understanding the diversity of a certain region and how it

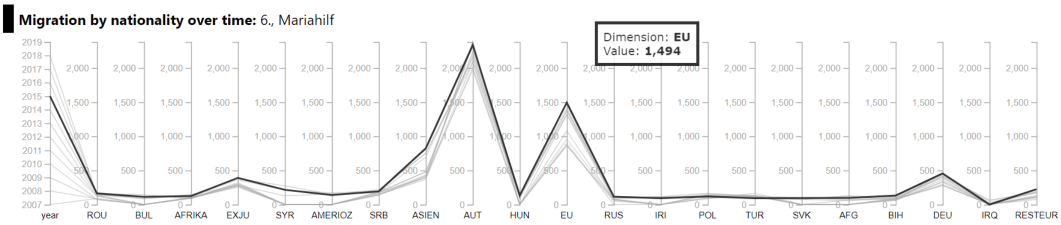


Figure 4.22: *Composition of countries of birth.* A parallel coordinates graph shows the migration volume of each country or geopolitical entity migrating to *6., Mariahilf.*

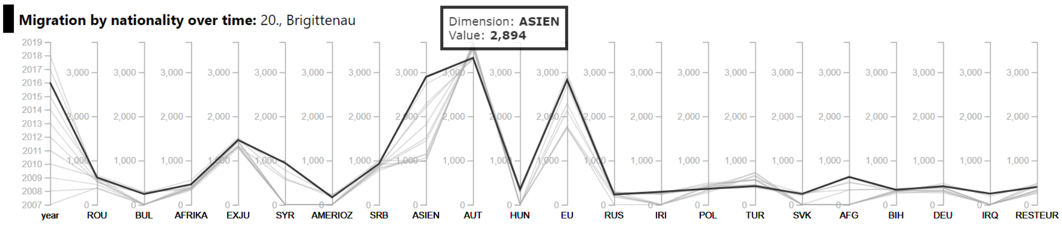


Figure 4.23: *Composition of countries of birth.* A parallel coordinates graph shows the migration volume of each country or geopolitical entity migrating to *20., Brigittenau.*

changes over time or how it correlates to certain events, like the refugee crises in 2015 and 2016. Interesting questions the user could answer with this visualization would be:

- Does the composition of the immigration to a district change over time?
- What is the composition of a specific sub-district, does it differ from the corresponding district's composition?
- Are certain events visible in the composition, e.g., the refugee crisis?

4.7.4 Total migration over time

The visualization components described so far do not reflect the overall migration change over time of a district. The third time-series visualization shows a simple grouped bar chart, displaying the sum of people who immigrated, people who emigrated, and people who moved within the region for each time instant. In this view, the user can compare the volume of each migration type within a year, or observe trends over time. Interesting questions the user could answer with this visualization would be:

- Are there years, in which more people left the region than immigrated?
- Are there trends in the immigration- does it stagnate, increase or decrease over time?

- Which events had a certain effect on the overall migration statistics in a certain area?

Figure 4.24 shows the evolution of the migration volume for the selected 17th district from 2007 to 2018. The color scheme for the bars corresponds with the migration types: immigration, emigration, and inherent migration. In the year 2018, more people left the district than immigrated.

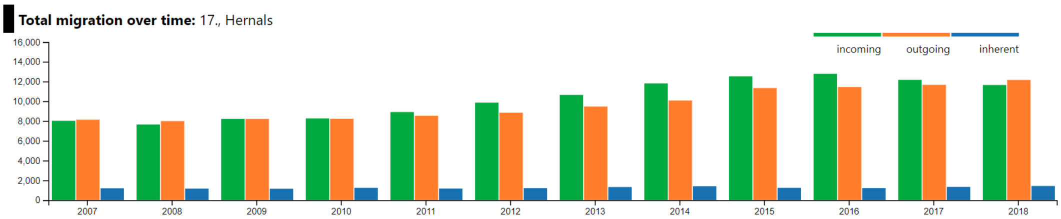
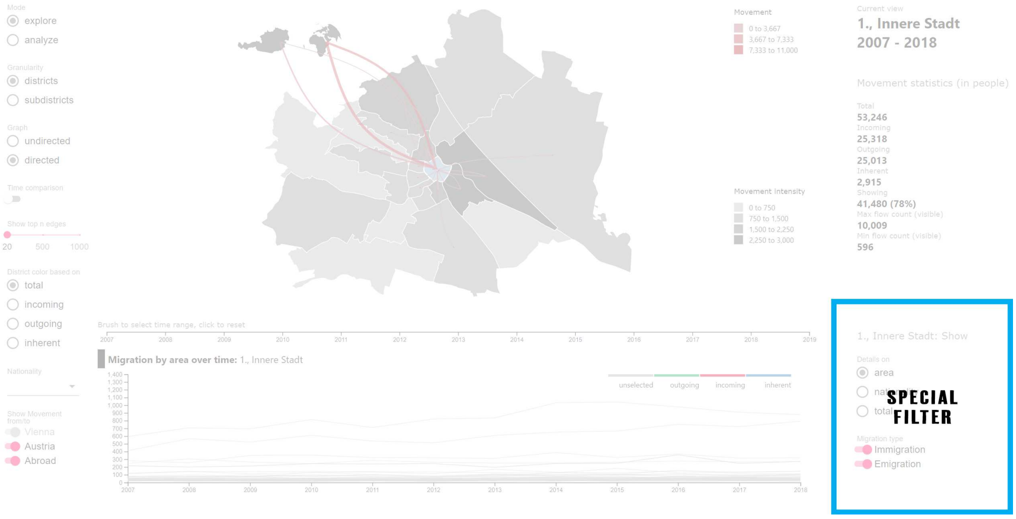


Figure 4.24: *Migration volume over time.* The grouped bar chart shows the time-oriented data of the migration volume of *17., Hernals*.

4.7.5 Special Filtering

In the bottom right corner of the screen, as shown in figure 4.25, the user can switch between the three time-series-data-visualizations defined in 4.7.1 as well as filter the time series data.



Mode
◉ explore
○ analyze

Granularity
◉ districts
○ subdistricts

Graph
○ undirected
◉ directed

Time comparison

Show top n edges

District color based on
◉ total
○ incoming
○ outgoing
○ inherent

Nationality

Show Movement from/to
Vienna
Austria
Abroad

Movement
0 to 3,667
3,667 to 7,333
7,333 to 11,000

Movement intensity
0 to 750
750 to 1,500
1,500 to 2,250
2,250 to 3,000

Current view
1., Innere Stadt
2007 - 2018

Movement statistics (in people)
Total
53,246
Incoming
25,318
Outgoing
25,013
Inherent
2,915
Showing
41,480 (78%)
Max flow count (visible)
10,009
Min flow count (visible)
596

Brush to select time range, click to reset

Migration by area over time: 1., Innere Stadt

1., Innere Stadt: Show
Details on
◉ area
○ nationality
○ total

Migration type
Immigration
Emigration

Figure 4.25: *Special filtering*: If a node is selected, the user has additional possibilities to filter the time series data and to choose from different time-series-visualizations.

Based on the selected district or sub-district, the user is able to show and hide data of the following migration types:

- Immigration
- Emigration

The selected filters in this area have an impact on the time-series-data-visualizations as well as on the main visualization.

4.8 Interaction: Cross-highlighting, Tooltip and Zoom

Interaction is a vital component of the visualization of complex multivariate networks. We already introduced interactions on the *Visual-structure level (Selection of areas*) and on the *Data level (Filtering the data)*. The following methods describe interactions on the *View level*, which relates to the visual emphasis of interesting objects.

4.8.1 Cross-highlighting

Although the layout composition with its different visualization components has its advantages in focusing on the strength of each component, it increases the difficulty of building and preserving the mental model throughout the whole visualization. Therefore, it is important to cross-reference certain data cases which are observed in a particular component and to highlight or emphasize them also in the other components. This is particularly true for the same visual entities presented in different styles. These entities in our case are the main constituent parts of the network: nodes (districts, sub-districts) and edges (migration flows).

We chose to apply the concept of cross-highlighting to the main visualization components (the *geographical map* and the *force-layout* as well as to the time-series visualization: *Migration by area over time*). Whenever a region is selected, hovering over another district or edge highlights the corresponding line in the line chart. Likewise, if hovering over a line in the time-series-chart, the corresponding areas are highlighted. Figure 4.26 shows the cross-highlighting of the corresponding lines in the time-series visualization when hovering over a related area or edge on the map.

4.8.2 Tooltip

Every visualization component consists of a number of visual entities, like a node or an edge in the network layouts, a line in a graph, or a bar in a barchart. These entities have different attributes attached to them. If the user is interested in one entity, for example a specific edge, the attached attributes should be displayed. In combination with *Hovering*, a Tooltip next to the entity displays these attributes based on the user's interest. Figures 4.27, 4.28, and 4.29 show different tooltip attributes based on the entity of interest.

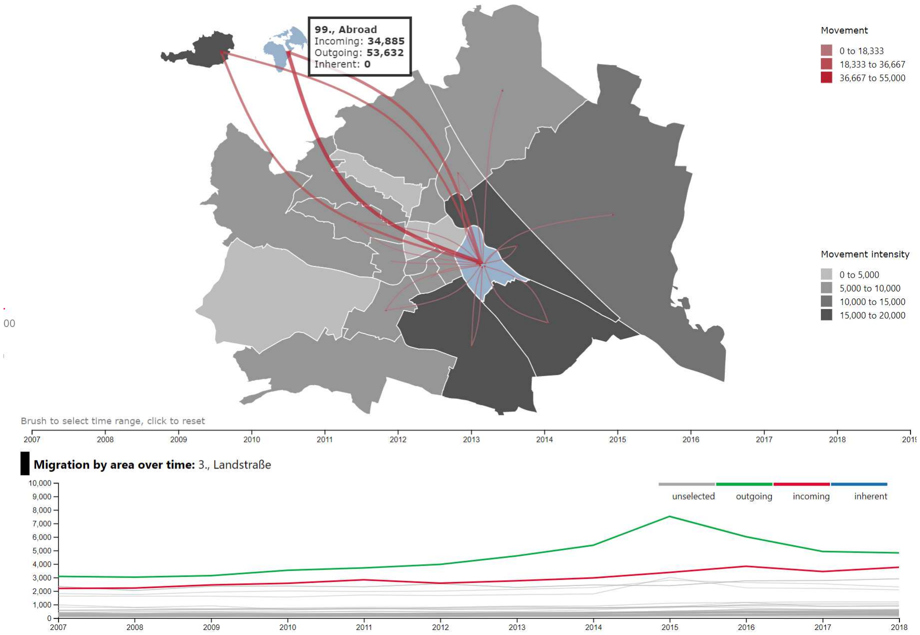


Figure 4.26: *Cross-highlighting.* The screen shows the migration in the selected 3rd district *Landstraße*. Hovering over the region *abroad* highlights the migration flows corresponding to both regions in the line chart.

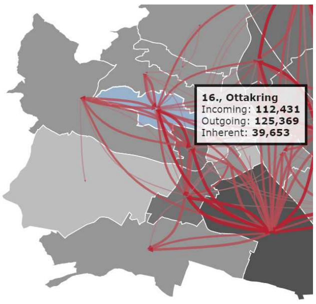


Figure 4.27: *Node tooltip.* Showing the attributes of a node.

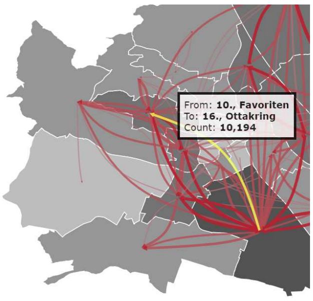


Figure 4.28: *Edge tooltip.* Showing the attributes of an egde.

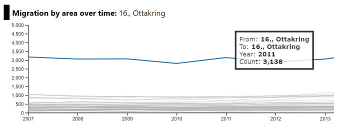


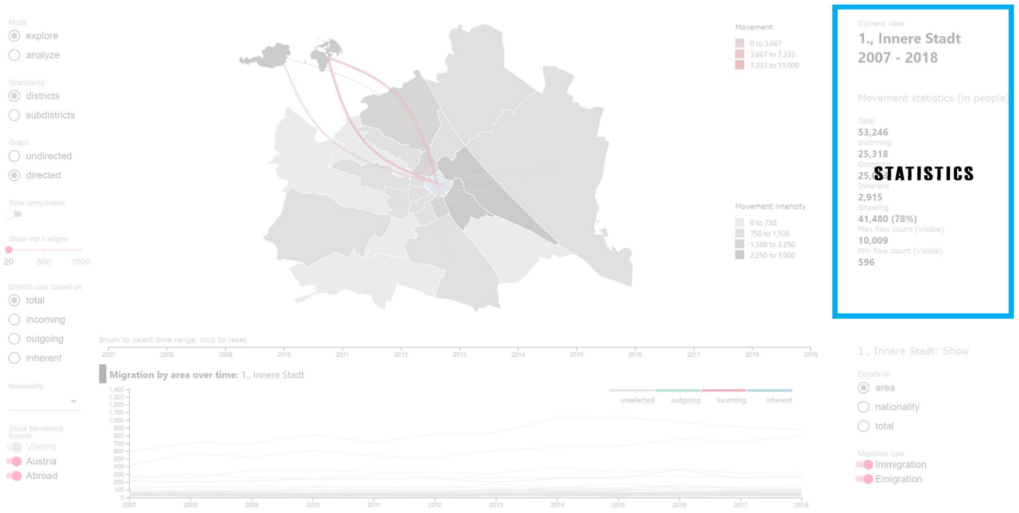
Figure 4.29: *Time series tooltip.* Showing the attributes of a data point.

4.8.3 Zoom

The main visualization components show a lot of complex information. There are a lot of nodes and edges drawn onto the map and complex relationships in the force-directed-graph. Zooming into the diagrams changes the viewport and enlarges the interesting area, therefore, supports the user to distinguish even the tiniest details.

4.9 Statistics component

Due to the many possibilities to view, filter, and select the migration data, it is easy to get lost in details, loosing sight of the big picture. Therefore, it is important to offer the user a summary of what is currently displayed as a visual anchor. The upper right corner of the screen as shown in figure 4.30, shows a short summary of the current view and the corresponding data details currently displayed in the visualization.



Current view
1., Innere Stadt
2007 - 2018

Movement statistics (in people)

Total
53,246
Incoming
25,318
Outgoing
25,0[illegible]
Inherent
2,915
Showing
41,480 (78%)
Max flow count (visible)
10,009
Min flow count (visible)
596

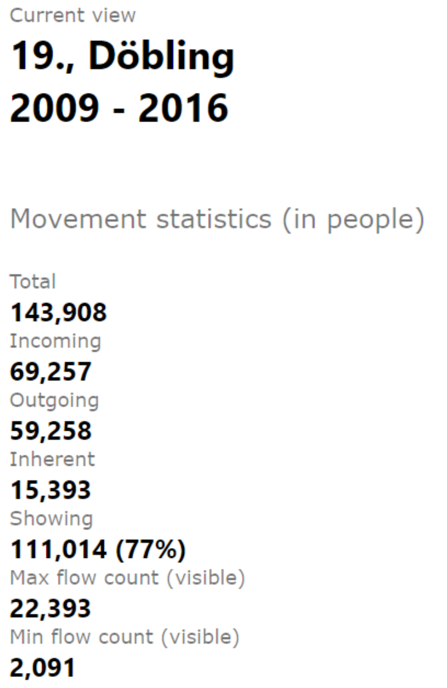
Mode: ● explore ○ analyze
Granularity: ● districts ○ subdistricts
Graph: ○ undirected ● directed
District color based on: ● total ○ incoming ○ outgoing ○ inherent
Details on: ● area ○ nationality ○ total

Figure 4.30: *Show me the numbers*: Visual anchor for the user to analyze numbers.

As we already described in section 4.5, limiting the visible edges on the map hides a part of the data. This fact has to be made transparent to the user, avoiding misinterpretations and wrong conclusions. Figure 4.31 shows the statistics component in the *exploration mode*, a specific selected time range and a few visible edges. The statistics show that currently 77% of the total migration volume are shown in the edges. Figure 4.32 on the other hand shows the Overview map of the complete time range where only 25% of the total migration volume is visible.

Current view

The *current view* could be seen as a *title* of the visualization which displays a short summary of what the user is looking at: whether it is an overview of the migration data in the exploration mode, the relationships in the analysis mode or details for a specific area. Along with the title, the time span gives the hint on which years are included in the data. If the *time comparison* feature is enabled, it shows that two years (range start vs. range end) are being compared, disregarding the time in between, where in other modes, the selected time range (range start - range end) is displayed.



Current view

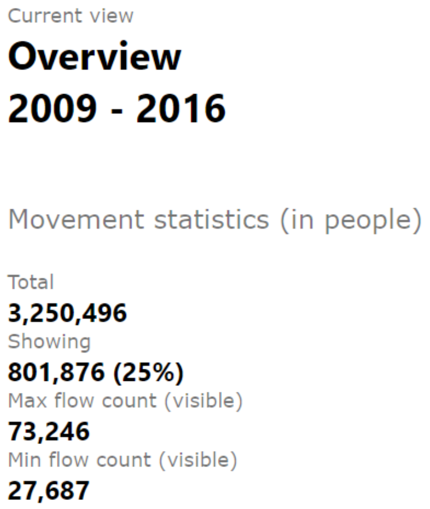
19., Döbling

2009 - 2016

Movement statistics (in people)

Total
143,908
Incoming
69,257
Outgoing
59,258
Inherent
15,393
Showing
111,014 (77%)
Max flow count (visible)
22,393
Min flow count (visible)
2,091

Figure 4.31: *Selected district.* The component shows the currently selected district *19., Döbling* along the selected time range, as well as detailed information about how many people were migrating and what parts of this migration are visible.



Current view

Overview

2009 - 2016

Movement statistics (in people)

Total
3,250,496
Showing
801,876 (25%)
Max flow count (visible)
73,246
Min flow count (visible)
27,687

Figure 4.32: *Overview.* Currently, the overview is displayed in the visualization. Detailed information about how many people were migrating and what parts of this migration are visible is shown below.

Total movement (in people)

This number represents the aggregated total of the movement for the filtered data, disregarding the edge limit.

Showing

These numbers represent the aggregated movement volume currently depicted by the visible edges.

Flow counts

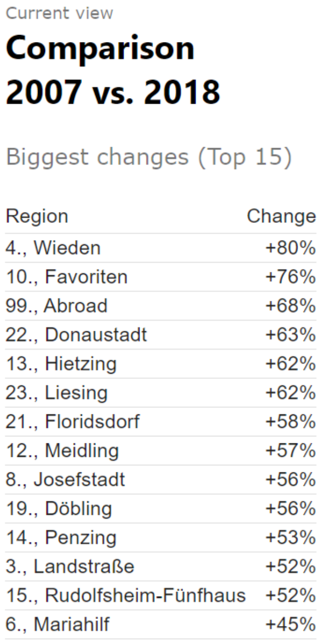
The *Max flow count (visible)* represents the highest flow volume of a single edge visible in the main visualization, while *Min flow count (visible)* shows the smallest single visible flow volume. These attributes may be useful to decide if more visible edges will add significant changes or not. In a fictive example, 40 edges represent 60% of the total migration of the filtered data. The *Min flow count* is already very low, which means that every edge that is shown additionally, will have the same or lower flow volumes. In this example it is the case, that showing more edges will only introduce visual clutter without significantly increasing the *visible flow volume*.

Movement metrics

The values of *Incoming*, *Outgoing*, and *Inherent* display the number of people who migrated regarding a selected area.

Relative changes

In the *time comparison* view, the statistics component shows the top 15 regions with the highest relative change of the selected movement intensity. These numbers support the gradient visualization to highlight big changes in the migration as shown in figure 4.33.



Current view

Comparison
2007 vs. 2018

Biggest changes (Top 15)

Region	Change
4., Wieden	+80%
10., Favoriten	+76%
99., Abroad	+68%
22., Donaustadt	+63%
13., Hietzing	+62%
23., Liesing	+62%
21., Floridsdorf	+58%
12., Meidling	+57%
8., Josefstadt	+56%
19., Döbling	+56%
14., Penzing	+53%
3., Landstraße	+52%
15., Rudolfsheim-Fünfhaus	+52%
6., Mariahilf	+45%

Figure 4.33: *Time comparison*: The statistics component shows the top 15 regions with the biggest relative change between the first and last year of the selected time range.

CHAPTER 5

Prototype Implementation

This chapter describes the architecture and implementation details of the interactive prototype. We chose to only include parts of the code which have a relation to the design choices described in chapter 4. The basic use of various functions of the d3 API is extensively documented on the project's github page [D3A] and therefore, not part of this section.

5.1 Architecture

We use a client-server architecture for the prototype. For data persistence, there is a PostgreSQL database running on the server. The Node.js server acts as an HTTP server for receiving RESTful HTTP requests from the client. Furthermore, it queries the database for the requested data which is transferred to the client in the JSON format.

The client is implemented with the use of the React JavaScript framework [Rea] for the state management. It is handling the communication and control of the User Interface Components, e.g., Cross-highlighting between components. The visualization components use D3.js [D3J], the de-facto state of the art JavaScript data visualization library for the web. The following sections dive deeper into the different approaches and challenges of the implementation of this prototype.

5.2 Backend

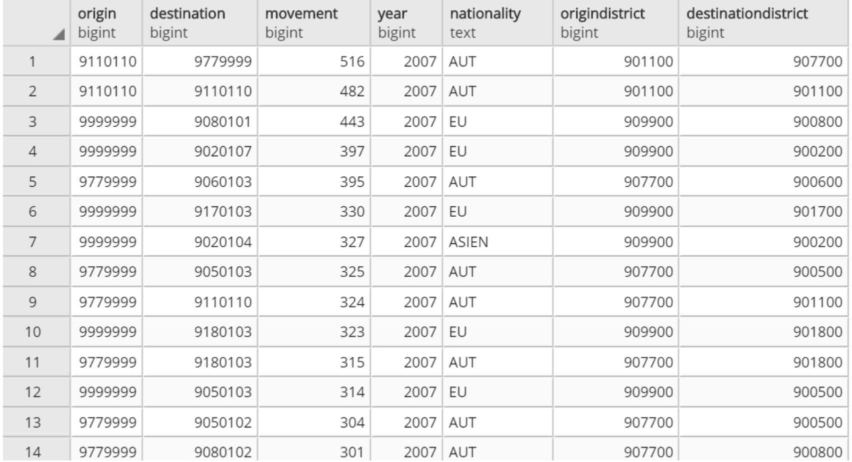
One part of the backend is used for parsing, preprocessing and loading the data into the PostgreSQL database [Pos]. The over 190 csv files provided by the City of Vienna, department MA23 [MA2], are parsed, restructured and stored in the database in the initial data load.

Each csv-file is structured as a matrix of the *origin* sub-districts as *rows* and the *destination* sub-districts as *columns*. Each cell's numerical value represents the flow volume from the origin to the destination. There is one file for each year and geo-political entity. For the data aggregation part, and to be flexible on the number of columns (respectively sub-districts), the data is restructured into a very flat format. Listing 5.1 shows the structure of the database table which stores the movement records. To accelerate the data aggregation for the districts, the data load extracts the *origin district code* and the *destination district code*.

```
CREATE TABLE data
(
        origin BIGINT,
        destination BIGINT,
        movement BIGINT DEFAULT 0,
        year BIGINT,
        nationality TEXT,
        origindistrict BIGINT,
        destinationdistrict BIGINT
)
```

Listing 5.1: Create Table statement for structure of the movement records

Figure 5.1 shows sample data records of the migration table.



	origin bigint	destination bigint	movement bigint	year bigint	nationality text	origindistrict bigint	destinationdistrict bigint
1	9110110	9779999	516	2007	AUT	901100	907700
2	9110110	9110110	482	2007	AUT	901100	901100
3	9999999	9080101	443	2007	EU	909900	900800
4	9999999	9020107	397	2007	EU	909900	900200
5	9779999	9060103	395	2007	AUT	907700	900600
6	9999999	9170103	330	2007	EU	909900	901700
7	9999999	9020104	327	2007	ASIEN	909900	900200
8	9779999	9050103	325	2007	AUT	907700	900500
9	9779999	9110110	324	2007	AUT	907700	901100
10	9999999	9180103	323	2007	EU	909900	901800
11	9779999	9180103	315	2007	AUT	907700	901800
12	9999999	9050103	314	2007	EU	909900	900500
13	9779999	9050102	304	2007	AUT	907700	900500
14	9779999	9080102	301	2007	AUT	907700	900800

Figure 5.1: *Migration records*: Excerpt from the table which stores the migration data.

There are many cells in the data matrices with a movement volume of 0, therefore, no migration happened between these origin and destination sub-districts. For reasons of efficiency and to reduce the payload which is transferred from the server to the client, records with a zero value movement were excluded during the data load. These records are generated on demand if the zero-value movements are needed, for example in the time series data component, which we will discuss in the frontend section of this chapter.

The Node.js server which receives RESTful HTTP requests uses the Express.js framework as the RESTful WebAPI Middleware. Although the REST routes offer different data aggregated by the server on demand, we chose to transfer the complete dataset to the client and to do data aggregation and data manipulation exclusively on the client. This design decision was made during implementation of the filter component because of the vast amount of filter options and different data structures required by the many visualization types.

5.3 Frontend

The user interface is built with the React JavaScript framework. This framework, developed and maintained by Facebook, is widely used because of its simplicity regarding the creation of reusable user interface components, its flexibility, state management, and high performance. D3.js is a data manipulation and visualization library for JavaScript. It offers a very big set of functions for manipulating data, creating charts and animating them.

Both technologies are emphasizing a declarative way of building web applications. While React makes use of an virtual DOM, which decides which components and when to re-render, D3 offers routes for *directly* manipulating the DOM. Since both of them want to rule the DOM, it can be very tedious to integrate them, without a clear separation.

5.3.1 React and D3

The main strategy on integrating React and D3 is to separate them by components. The pure React components render the user controls, e.g., in the *Filter component* utilizing Material UI. React also handles the state management for updating the data, tooltips and the cross-highlighting as well and the decision when to update the visualization components. The visualization components, which use D3 as rendering engine, take control of a DOM container, an *svg* element, passed from React to D3. From there, D3 creates, updates, and destroys the visualization elements, like edges, lines in the line chart or nodes in the force-directed graph.

At first, we took another approach on integrating React and D3. React should handle every rendering and have exclusive control of the DOM. This approach used D3 only for data manipulation, calculations of the visualization elements, like paths, circles, and so on, which were then rendered by React. This approach was considered to be the more *maintainable*, and from a software engineering perspective the *cleaner* way. Listing 5.2 shows a simplified version of React rendering everything for the districts on the map. However, as the complexity of the visualization increased, this strategy was not scaling very well. We observed significant loss of performance on the cross highlighting as well as the force-directed graph. As some of the functions of D3 need direct access to the DOM, they were not applicable. Since this is not a thesis about the integration of React and D3, we changed to the previously described method, to utilize the full power of D3, carefully separating the access to the DOM. Listing 5.3 shows a simplified version of our chosen approach, separating React and D3.

```
function ViennaMap({ geoData }) {
  // effect hook, for update the map on data changes
  useEffect(() => {

    // D3 calculating paths
    const districts = geoData.features
       .map((feature,i) =>
          <path
            key={'path' + i}
            d={pathGenerator(feature)}
            className='districts'
          />)

    ....

    }, [ geoData ]);

  // React redering
  return (
    <svg ...>
      <g id='districts' />
        { districts }
      <g id='edges' />
      ...
      </g>
    </svg>);
}
```

Listing 5.2: *React renders.* D3 calculates the svg paths for the districts. React renders everything.

```
function ViennaMap({ geoData }) {
  // reference to the svg container
  const svgRef = useRef();

  // effect hook, for update the map on data changes
  useEffect(() => {
    const svg = select(svgRef.current);

    ...

    // D3 rendering
    svg.select('#districts')
      .selectAll('.district')
      .data(geoData.features)
      .join('path')
      .classed('district', true)
      .attr('d', feature => pathGenerator(feature));

  }, [ geoData ]);

  // React redering
  return (
    <svg ref={ svgRef } ...>
      <g id='districts' />
      <g id='edges' />
      ...
      </g>
    </svg>);
}
```

Listing 5.3: *D3 renders.* React only renders the svg container and then passes the control over to D3, which renders the districts.

5.3.2 Data aggregation

To utilize the strengths of the various visualization components, we need to aggregate the data in multiple ways. Those records, which are satisfying the selected filter conditions, are aggregated depending on the visualization they address.

Migration flows

For the *Main visualizations*, the migration data is aggregated over the complete dataset for every ordered/unordered pair of districts, respectively sub-districts to show the total movement between them. Listing 5.4 shows the *reducer* for aggregating the undirected migration data for the main visualizations for each unordered pair of districts. Listing 5.5 shows the data aggregation for the directed migration flows, taking the ordering of the districts (source, target) into account. The reducers create a dictionary for each edge which is then converted to an array to process them in the main visualizations.

```
const aggregateDistrictsUndirected = (acc, cur) => {
  const minDistrict = Math.min(cur.origin, cur.destination)
  const maxDistrict = Math.max(cur.origin, cur.destination)
  acc[[minDistrict, maxDistrict]] = {
    point1: minDistrict,
    point2: maxDistrict,
    movement: acc[[minDistrict, maxDistrict]] ?
      acc[[minDistrict, maxDistrict]].movement + cur.movement
        :
      cur.movement,
  }
  return acc;
}
```

Listing 5.4: *Data aggregation of undirected flows.* The reducer to aggregate over every year disregarding the ordering of source and target node.

```
const aggregateDistrictsDirected = (acc, cur) => {
  acc[[cur.origin, cur.destination]] = {
    point1: cur.origin,
    point2: cur.destination,
    movement: acc[[cur.origin, cur.destination]] ?
      acc[[cur.origin, cur.destination]].movement+cur.movement
        :
      cur.movement,
  }
  return acc;
}
```

Listing 5.5: *Data aggregation of directed flows.* The reducer aggregates over every year for directed migration flows.

Movement intensities

Furthermore, the movement intensities are calculated for every area based on the filtered dataset and the direction of the movement flow as shown in listing 5.6.

```
const aggregateFillData = (acc, cur) => {
  // initialize dictionary
  ...
  // calculate metrics
  ...
  if(cur.origin === cur.destination) {
    // calculate internal movement
    acc[cur.origin] = {
      ...actualOrigin,
      internal: actualOrigin.internal + +cur.movement
      };
  } else {
    // count outgoing amount in origin
    acc[cur.origin] = {
      ...actualOrigin,
      outgoing: actualOrigin.outgoing + +cur.movement
    };

    // count incoming amount in destination
    acc[cur.destination] = {
      ...actualDestination,
      incoming: actualDestination.incoming + +cur.movement
    };
  }
  return acc;
}
```

Listing 5.6: *Data aggregation of movement intensities.* The reducer aggregates the different movement intensities for each area.

Time series data

The data for the *Time series visualizations* is aggregated, if an area is selected by the user. Depending on the type of visualization, the data are aggregated either by the areas, the countries of birth, or the movement intensities for each year included in the selected time range.

5.3.3 Force directed graph layout

The force directed graph layout module in d3 implements a *velocity Verlet* numerical integrator for simulating physical forces on particles [D3F]. This iterative algorithm uses different forces to set particles in motion in combination with a velocity decay in each iteration, to determine the final position of the nodes.

Repulsion force

In general, every node in the network has a repulsing force to the other nodes. In a network without edges the nodes would be equally distributed over the visible space.

This repulsion force ensures some distance between unconnected nodes and increases the readability by preventing one big *hairball*.

Attraction force

An attraction force applies to nodes by changing the edge length according to the flow volume. This pulls the nodes from their original position through increasing the velocity of the motion towards the target node. In every iteration of this simulation, this velocity is decreased due to friction or velocity decay until it finally stops moving.

The initial length of an edge is calculated based on the movement intensity. This length constraint is relaxed indirect proportional to the flow volume, which means that the length of a high flow volume edge may change more than the length of an edge with a low flow volume. This method is called *constraint relaxation* and it ensures, that low flow volumes encode more distance between nodes than high volumes.

Every edge has a maximum length to prevent nodes from shooting too far out of the visible space. Displaying every edge in the network has therefore, limited expressiveness, because the weight of high flow volumes has less impact in relation to the many low volume edges, which also apply some force. We observed high expressiveness and a good impression of the clustering when the minimum flow volume is about 10% of the maximum flow volume. This is no general rule, but since the highest migration flows are the most influencing ones, the analysis focuses on them more often.

Furthermore, the user is able to see the impact of the force by dragging one node away from the network. High correlation relation between districts force the dragged node to immediately bounce back if released. Low correlation relations have less impact on the other nodes when dragged. Highly connected, and thus very important nodes, pull other nodes with them when dragged.

Force simulation configuration

Listing 5.7 shows the configuration of the different forces applied to the graph. The weight of each edge ranges from *0.05* to *0.4* which is the base for calculating the edge length and the relaxation constraint.

- **Link** sets the attraction force for the edges where distance is indirect proportional to the movement weight, and the relaxation constraint is direct proportional to the movement weight.
- **Charge** sets the repulsion force for the nodes, whereas districts have a higher repulsion force setting then sub-districts.
- **Collision** sets the radius for each node to prevent too much overlapping of the nodes.

- **Center** sets the force center to the center of the visualization to prevent the nodes bouncing too far off the visible space.

```
// basic repulsion forces for the nodes based on granularity
const nodeForceStrengthSubdistrict = -100;
const nodeForceStrengthDistrict = -2000;
...
const weight = scaleLinear()
  .domain(extent(links, (d) => +d.value))
  .range([0.05, .4]);

...

const simulation = forceSimulation(nodes)
  // strength of the pulling force
  .force('link', forceLink(links).id(d => d.id)
    .distance((d) => 1/weight(+d.value))
    .strength((d) => weight(+d.value)))

  // strength of the repulsion force of the nodes
  .force('charge', forceManyBody()
    .strength(nodeForceStrength)
    .distanceMax(height / 2))

  // do not let the nodes overlap
  .force('collision', forceCollide()
    .radius((d) => radius(d)))

  // center the force to center of the svg
  .force('center', forceCenter(+width / 2, +height / 2))
```

Listing 5.7: *Force directed graph.* Configuration of the force simulation with the force parameters.

5.3.4 Tooltips in Time series visualizations

The line chart as well as the parallel coordinates chart show different lines related to the migration data. In the multiple line graph, each line represents the change over time of the movement of the selected district with another district. In the parallel coordinates graph, one line shows the composition of a district for a specific year and the migration volume for each country of birth.

We implemented a tooltip which shows the details of the data point in the chart the user is interested in. Based on the mouse position, we calculate the closest data point, so that the user does not need to hover exactly over the thin lines to observe the details. Listing 5.8 shows the calculation of the closest data point in the line chart, based on the location of the mouse pointer. First, the year on the x-Axis is determined by inverting the mouse coordinates to the x-Axis scale and calculating the closest year. Within the data array for this year, the element with the closest movement value is chosen to be displayed.

```
// highlighting on mouse events
svg.select('#linechart')
  .on('mousemove', (d) => {
    ...
    // invert mouse positions to scale values
    const x0 = x.invert(mouse(event.currentTarget)[0]);
    const y0 = y.invert(mouse(event.currentTarget)[1]);

    // index of data array where the point would be inserted
    const i0 = bisectDate(timeRange, formatTime(x0), 0);

    // determine closest neighbouring year
    ...
    i = leftDistance > rightDistance ? ++i0 : i0;

    // get closest movement value in the calculated year
    const s = sumstat.reduce((a, b) =>
      Math.abs(a.values[i].movement) <
      Math.abs(b.values[i].movement) ?
          a : b
     );

    // highlight the closest path
    const currentPath = paths
      .classed('selected', false)
      .filter(d => s === d)
      .classed('selected', true)
      .raise();

    // tooltip update
    ...
  })
```

Listing 5.8: *Line chart component.* Calculation of closest data point from the mouse pointer to show the data detail.

CHAPTER 6

Evaluation

In the evaluation phase, the implemented visualization prototype is evaluated against the requirements defined in section 3.4.

The evaluation of techniques in the field of Information Visualization and Visual Analytics is a very hard task. The applied tools and techniques, to support users in gaining insight and deeper understanding into the underlying data, are complex and the effectiveness is hard to quantify.

Heuristic evaluation is a very common approach to identify problems in the usability of a user interface, in which the evaluators rate the UI based on a defined set of established usability principles named *heuristics*. The selection of the right set of heuristics is important to identify problems which are relevant for the type of user interface under evaluation.

One of the best known approaches is presented in the work of Nielsen [Nie94] which contains 10 general usability principles.

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use

- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

These very general principles can be applied to any interactive user interface and are not specific to Visual Analytics. Therefore, they are useful to identify problems on the interaction level but not problems related especially to the challenges of Visual Analytics.

The work of Zuk et al. [ZC06] presents heuristics which were compiled to evaluate uncertainty visualizations based on general principles outlined in the work of Bertin, Tufte, and Ware. Furthermore, Forsell et al. [FJ10] created a set of heuristics based on 6 different previously available sets of varying levels. In a user study, they analyzed how well the 63 different heuristics, including the work of Nielsen and Zuk et al., would identify 74 usability problems, which were derived from previously conducted evaluations. They integrated the 10 heuristics with the highest explanatory value into a new set, containing the following:

- Information coding
- Minimal actions
- Flexibility
- Orientation and help
- Spatial organization
- Consistency
- Recognition rather than recall
- Prompting
- Remove the extraneous
- Data set reduction

Santos et al. [SFD15] compared these three sets of heuristics in a user study in which the evaluators had to choose a specific set and apply it to an example visualization. This study investigated the understandability of the heuristics as well as the correct application of them. The study concluded, that the use of heuristic evaluation as an evaluation method is useful to identify common problems in interactive systems. Another result was that the work of Nielsen was easy to understand and to apply but it was more suitable to identify problems in the general user interface while the works of Zuk et al. and Forsell et al. were more appropriate to identify problems related to the challenges of Visual Analytics. The authors also stated, that heuristic evaluation produces useful results even if the evaluators are less experienced. Therefore, they suggest to include this method in the developer's evaluation toolkit.

Since many of the heuristics were already integrated in the iterative design process, we decided to let our prototype be evaluated by visualization experts, which is described in the following section.

6.1 Expert evaluation

The visualization is assessed by five visualization experts and one UI expert, who work on a series of questions. They have to answer, if the defined tasks can be achieved, in more detail, if the visualization enables the user to answer the following questions, which reflect the main requirements as shown in table 6.1.

6.1.1 Methodology

The evaluation is divided into five parts.

- The first part assesses the background about the evaluators confidence in Visual Analytics and the geography of Vienna.
- In the second part, the evaluators are given a brief introduction to the implemented prototype, as well as the aim of the visualization and the target audiences.
- The third part is a ten minutes session, in which the evaluators make themselves familiar with the prototype and explore it on their own.
- The forth part involves specific tasks, which the evaluators should solve. Since the design followed a user-centred-design, they should be able to achieve these tasks with support of the prototype. Each task is rated accordingly to their difficulty and general suggestions and remarks are noted.
- In the last part, the evaluators reflected on the visualization design in general, the user interface, the tasks they had to solve, and how the visualization prototype was supporting them.

The average time to conduct the evaluation was two hours.

6.1.2 Tasks

The following list presents the tasks, which the evaluators were given to solve. Table 6.1 shows the mapping of the analysis tasks to the requirements defined in section 3.4.

T1.1: Which district has the highest immigration overall?

T1.2: How does that immigration change over time?

T2.1: Which district has the highest immigration of Austrian people overall?

T2.2: How does that immigration change over time?

T3.1: Between the districts *Favoriten* and *Donaustadt*: Which district has a higher diversity of nationalities?

T4.1: Explore the movement of Austrian people in Vienna for *11., Simmering*. What is your impression of the movement behaviour? (far, chaotic, near, follows a pattern, ...)

T5.1: In 2017, the extension of the underground line *U1* to *Oberlaa* was finished. Did something change in this sub-district, and when?

T6.1: Which nationality (or geo-political entity) has the highest immigration from abroad to Vienna? Did something significantly change over time?

T7.1: Which district had the highest immigration from abroad of Syrian people?

T7.2: Is this immigration distributed over the whole district equally, or does it focus on some area? What could be the reason?

T8.1: Internal migration: Analyze the relationship of districts for people born in Turkey. Are there visible clusters?

T8.2: Internal migration: Analyze the relationship of districts for people born in Austria. Are there visible clusters?

T8.2: Analyze the relationship of sub-districts for Austria and Turkey. Are there other clusters? What relation do the sub-districts have with the districts?

T9.1: Where do people from the 7th district mostly move to, is there a preference?

T9.2: Where do people from the 8th district mostly move to, is there a preference?

T9.3: Where do people from the 9th district mostly move to, is there a preference?

T9.4: In T9.1 - T9.3, is there a pattern visible?

T10.1: From where do citizens immigrate to the Seestadt Aspern? How diverse is this immigration related to *Country of birth*?

T11.1: Compare the immigration intensity of 2010 and 2018. Is there a pattern visible?

Task	Related requirements
T1	R1, R2, R4, R5
T2	R1, R2, R4, R5, R8
T3	R8
T4	R1, R2, R3, R8
T5	R1, R2, R3, R4
T6	R1, R2, R4, R5, R8
T7	R1, R2, R3, R8
T8	R5, R6, R7, R8
T9	R1, R2, R3
T10	R2, R3, R5, R8
T11	R3, R4, R5

Table 6.1: Task-Requirement mapping

6.1.3 Results

In the first step, we explained the problem definition and the cornerstones of the design triangle: *data*, *users*, and *tasks*. After that, the evaluators were introduced to the prototype by presenting the user interface components, their functionality, and how they are working together.

Due to the large amount of different functions and filtering possibilities, the evaluators were given ten to fifteen minutes to get themselves familiar with the user interface. Generally, we observed during the familiarization phase, that the evaluators quickly got used to the different functions and filters. Most of them instantly explored the migration data of the district they are living in. They found it intuitive to start with the overview and then drill deeper into details. All of the evaluators found the user interface and the arrangement of the UI components intuitive but stated, that an introduction to the prototype is necessary to understand the different functionalities and options. Especially the complex aggregation regarding to the geopolitical entities and the privacy restrictions needed a more extensive explanation.

After getting familiar with the user interface, the evaluators were asked different questions about their background knowledge in the field of Visual Analytics and the geography of Vienna.

Five of the six evaluators are visualization experts with great confidence in the field of Visual Analytics. Most of them were familiar with the geography of Vienna on the district level to identify where certain areas are located geographically. Table 6.2 shows the confidence scores of each evaluator on Visual Analytics and the geography of the city of Vienna.

Along with this background information, first impressions of the prototype were asked to be scored. Table 6.3 shows the evaluation scores of different aspects of the prototype,

Topic	E1	E2	E3	E4	E5	E6	AVG score
Visual Analytics	2	5	5	5	5	5	4.5
Geography	4	4	4	3.5	2.5	3	3.5

Table 6.2: The evaluators confidence scores.

such as usability or performance. These ratings were revised at the end of each session, where the evaluators were asked, if the impression rating still holds or if they want to change anything. Each evaluator confirmed the rating given after the familiarization phase.

- The *Usability* rating is scaled from *1: not usable* to *5: consistent and intuitive*.
- The *Features* rating is scaled from *1: missing features* to *5: rich/complete set*.
- The *Performance* rating is scaled from *1: very slow* to *5: very fast*.
- The *Overall impression* reflects if the prototype was complete in the sense of how useful the visualization of the explained data was. *1: useless* to *5: useful*.

Impression	E1	E2	E3	E4	E5	E6	AVG score
Usability	3	5	5	4	4	4	4.17
Features	5	5	5	4.5	4	3	4.42
Performance	4	4	4	4.5	4.5	5	4.33
Overall	5	5	5	5	4	3	4.50

Table 6.3: The evaluators impressions of the prototype, taken after step 1 and revised at the end of the session. The prototype was rated to be useful, quite easy to use, rich on features and fast.

In the main part of the evaluation, the evaluators used the prototype to solve the tasks defined in section 6.1.2. Each evaluator was able to solve all the given tasks. Table 6.4 shows the solution rate of each task.

During the tasks we sometimes gave assistance, if a evaluator did not notice, that a filter was set from before. After the solution, we discussed alternative ways to solve the task as well as the difficulty to achieve the goal with the help of the prototype. This difficulty rating is shown in table 6.5 for each task where the rating scales from *1: impossible* to *5: easy to solve* with respect to the support, the prototype provided.

We also observed a learning effect during the tasks. The users quickly got more confident in using the different controls. Nevertheless, that learning effect had only minor impact or none at all on the ratings of the latter tasks.

Task #	E1	E2	E3	E4	E5	E6	AVG score
T1	y	y	y	y	y	y	100.00%
T2	y	y	y	y	y	y	100.00%
T3	y	y	y	y	y	y	100.00%
T4	y	y	y	y	y	y	100.00%
T4	y	y	y	y	y	y	100.00%
T6	y	y	y	y	y	y	100.00%
T7	y	y	y	y	y	y	100.00%
T8	y	y	y	y	y	y	100.00%
T9	y	y	y	y	y	y	100.00%
T10	y	y	y	y	y	y	100.00%
T11	y	y	y	y	y	y	100.00%

Table 6.4: Solving rate of the tasks.

Task #	E1	E2	E3	E4	E5	E6	AVG score
T1	5	4	5	5	4	4	4.50
T2	4.5	5	5	5	4	5	4.75
T3	3	4.5	5	5	4	5	4.42
T4	5	5	5	5	4	4	4.67
T5	5	4	5	4	4	3	4.17
T6	4	4	5	5	4.5	4	4.42
T7	5	5	5	5	4	5	4.83
T8	5	3	4.5	5	4.8	4	4.38
T9	4	4	5	5	4	3	4.17
T10	3	5	5	5	4	5	4.50
T11	4	4	4	4	4	3	3.83

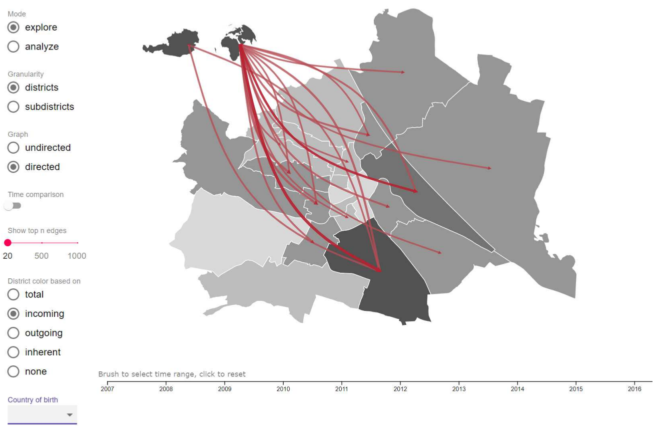
Table 6.5: Difficulty rating of the tasks. It shows, that the task T11 was rated the hardest with regards on how much the prototype supported the user.

6.2 Example solutions

In this section we outline example solutions of the evaluators to some of the defined evaluation tasks. While every task has multiple ways to be fulfilled, this section should showcase, how the elementary and synoptic tasks, defined in section 3.3, are performed.

6.2.1 T1 - Highest immigration volume

Figure 6.1 shows a solution to the task *T1.1*, in which the evaluators should identify the district with the highest immigration volume overall. To observe the immigration intensity of each district, the evaluator applied the display filter to color the districts based on *incoming* movement. This setting reveals *Favoriten* as the district with the highest immigration overall, as it is the darkest area.



Mode

● explore

○ analyze

Granularity

● districts

○ subdistricts

Graph

○ undirected

● directed

Time comparison

Show top n edges

20 500 1000

District color based on

○ total

● incoming

○ outgoing

○ inherent

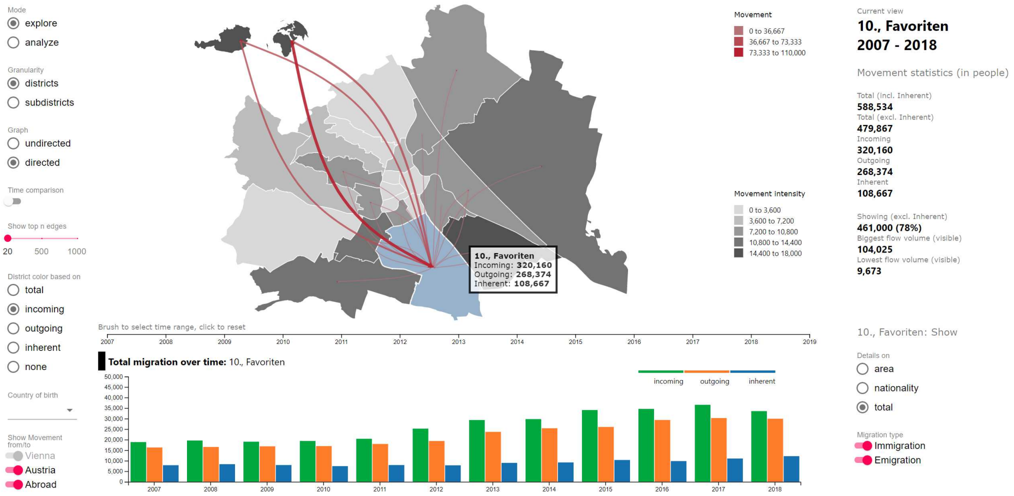
○ none

Country of birth

Brush to select time range, click to reset

Figure 6.1: *Elementary comparison*: The specific district coloring, based on the immigration count of each district, reveals *Favoriten* as the district with the highest immigration.

Figure 6.2 shows a solution to task *T1.2*. To observe the change over time of the immigration to *Favoriten*, the evaluator selected the district and chose the time series data visualization to show the total movement intensity. The bar chart shows a steady increase of the immigration to this district (green bars). In 2018, the immigration volume decreased slightly in comparison to 2017.



Mode

● explore

○ analyze

Granularity

● districts

○ subdistricts

Graph

○ undirected

● directed

Time comparison

Show top n edges

20 500 1000

District color based on

○ total

● incoming

○ outgoing

○ inherent

○ none

Country of birth

Show Movement from/to

Vienna

Austria

Abroad

Brush to select time range, click to reset

10., Favoriten
Incoming: 320,160
Outgoing: 268,374
Inherent: 108,667

Movement

🟥 0 to 36,667

🟥 36,667 to 73,333

🟥 73,333 to 110,000

Movement intensity

▨ 0 to 3,600

▨ 3,600 to 7,200

▨ 7,200 to 10,800

▨ 10,800 to 14,400

▨ 14,400 to 18,000

Current view

10., Favoriten

2007 - 2018

Movement statistics (in people)

Total (incl. Inherent)
588,534

Total (excl. Inherent)
479,867

Incoming
320,160

Outgoing
268,374

Inherent
108,667

Showing (excl. Inherent)
461,000 (78%)

Biggest flow volume (visible)
104,025

Lowest flow volume (visible)
9,673

Total migration over time: 10., Favoriten

10., Favoriten: Show

Details on

○ area

○ nationality

● total

Migration type

Immigration

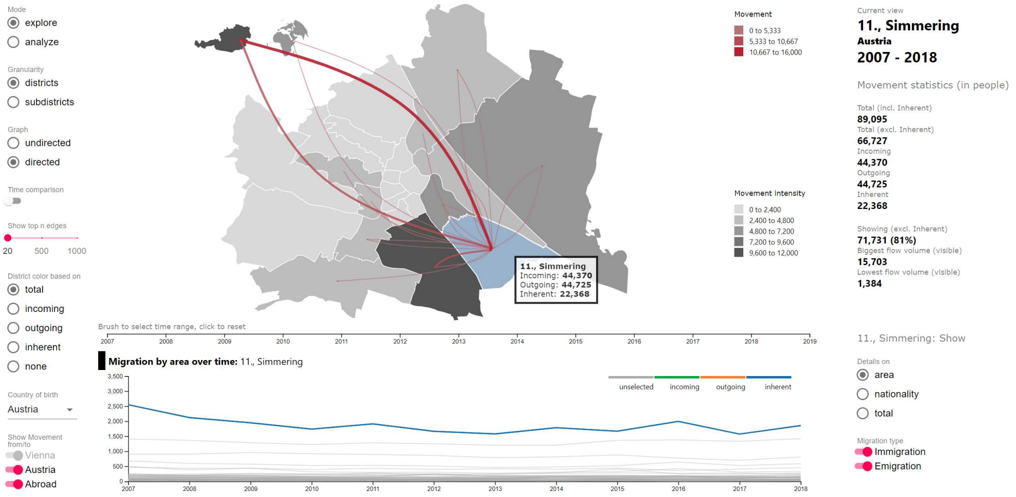
Emigration

Figure 6.2: *Elementary comparison*: The bar chart for the total migration over time of *Favoriten* shows a steady increase of the green bars until 2018, when the number of immigrating people slightly decreased.

6.2.2 T4 - Movement patterns for *11., Simmering*

In task *T4.1* we asked for the analysis of the movement behaviour of people born in Austria for the district *11., Simmering*. This *synoptic comparison task* asks for finding patterns in the movement behaviour.

Figure 6.3 shows the visualization with *Simmering* as the selected district for more detailed information on the time series data. The data is filtered by the *country of birth* to analyze only people born in Austria. The evaluator observed, that most of the people are moving within the district itself consistently over the whole time range. During this time period, this number decreased by about 30%.



Mode
◉ explore
○ analyze

Granularity
◉ districts
○ subdistricts

Graph
○ undirected
◉ directed

Time comparison

Show top n edges
20 500 1000

District color based on
◉ total
○ incoming
○ outgoing
○ inherent
○ none

Country of birth
Austria

Show Movement from/to
Vienna
Austria
Abroad

11., Simmering
Incoming: 44,370
Outgoing: 44,725
Inherent: 22,368

Current view
11., Simmering
Austria
2007 - 2018

Movement statistics (in people)

Total (incl. Inherent)
89,095
Total (excl. Inherent)
66,727
Incoming
44,370
Outgoing
44,725
Inherent
22,368

Showing (excl. Inherent)
71,731 (81%)
Biggest flow volume (visible)
15,703
Lowest flow volume (visible)
1,384

11., Simmering: Show

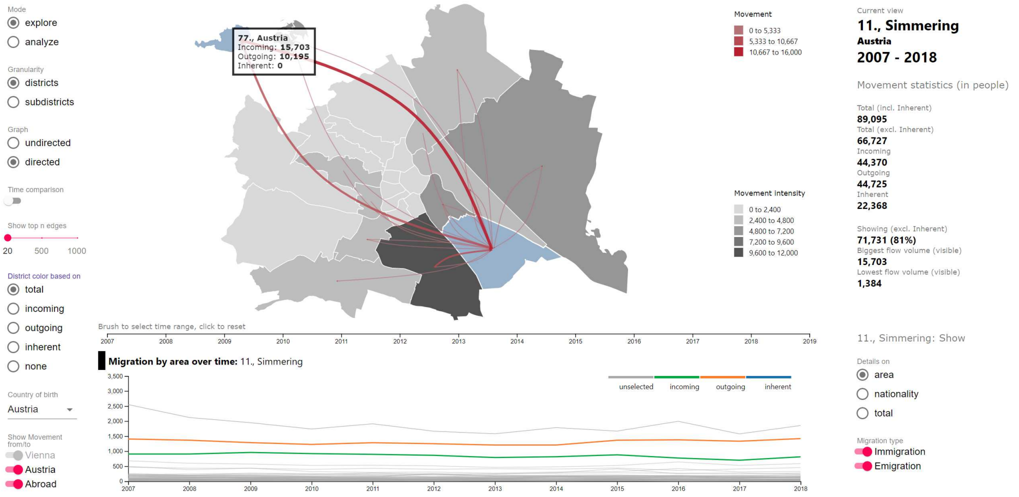
Details on
◉ area
○ nationality
○ total

Migration type
Immigration
Emigration

Figure 6.3: *Inherent movement over time:* Hovering over the selected district highlights the movement within *Simmering* over time.

By hovering over the depiction for *Austria*, the evaluators observed the change over time of the migration between *Simmering* and the *rest of Austria*. Figure 6.4 shows, that more people are leaving the district for *Austria* every year compared to the other way around.

Figures 6.5 and 6.6 show further analysis of the migration related to *Simmering* only *within* Vienna. Therefore, the evaluator filtered out the *external movement*. While in figure 6.5 the data were filtered furthermore, to only show the emigration from *Simmering*, figure 6.6 shows only the immigration to *Simmering*. The analysis of the evaluator showed a strong movement between neighbouring districts which are located in the south and in the east of Vienna, especially to the directly adjacent district *10., Favoriten*. The evaluators also mentioned, that more people are leaving for the eastern districts *21., Floridsdorf* and *22., Donaustadt* than the other way around.



Mode
◉ explore
○ analyze

Granularity
◉ districts
○ subdistricts

Graph
○ undirected
◉ directed

Time comparison

Show top n edges
20 500 1000

District color based on
◉ total
○ incoming
○ outgoing
○ inherent
○ none

Country of birth
Austria

Show Movement from/to
Vienna
Austria
Abroad

77., Austria
Incoming: **15,703**
Outgoing: **10,195**
Inherent: **0**

Movement
🟥 0 to 5,333
🟥 5,333 to 10,667
🟥 10,667 to 16,000

Movement intensity
▨ 0 to 2,400
▨ 2,400 to 4,800
▨ 4,800 to 7,200
▨ 7,200 to 9,600
▨ 9,600 to 12,000

Current view
11., Simmering
Austria
2007 - 2018

Movement statistics (in people)

Total (incl. Inherent)
89,095
Total (excl. Inherent)
66,727
Incoming
44,370
Outgoing
44,725
Inherent
22,368

Showing (excl. Inherent)
71,731 (81%)
Biggest flow volume (visible)
15,703
Lowest flow volume (visible)
1,384

Brush to select time range, click to reset

Migration by area over time: 11., Simmering

11., Simmering: Show

Details on
◉ area
○ nationality
○ total

Migration type
Immigration
Emigration

Figure 6.4: *External migration*: Hovering over Austria highlights the movement between *Simmering* and the *rest of Austria*. The biggest flow indicates the highest movement volume of people leaving *Simmering*.

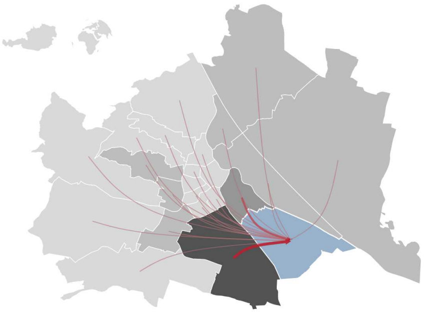


Figure 6.5: *Internal immigration:* Movement behaviour of people moving to *11., Simmering*. The biggest flow shows the high immigration from *Favoriten*.

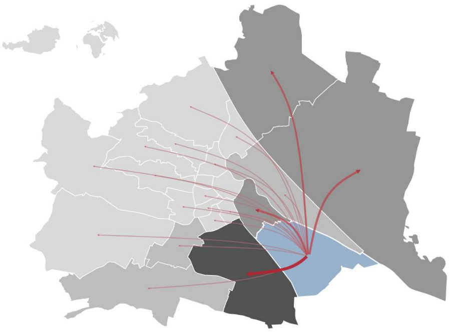
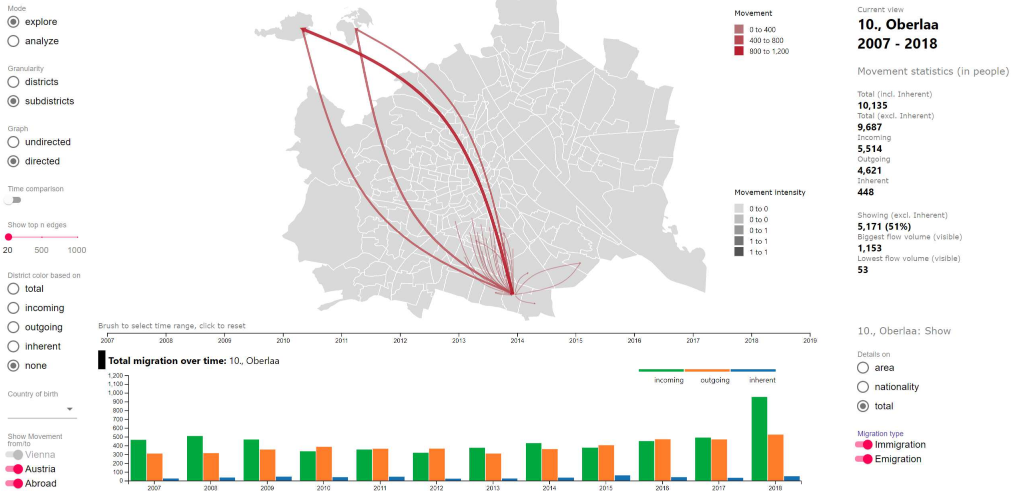


Figure 6.6: *Internal emigration:* Movement behaviour of people leaving *11., Simmering*. The biggest flow shows the high emigration towards *Favoriten*.

6.2.3 T5 - U1 extension to *Oberlaa*

This *elementary lookup* task asked the evaluator to observe a possible impact of an event. In 2017, the extension of the underground line U1 was finished in the sub-district *Oberlaa*. Figure 6.7 shows the map visualization on sub-district level. To focus the analysis on *Oberlaa*, the evaluator selected the requested sub-district. After analyzing the movement patterns by looking at the flows, the evaluator switched to the bar chart to observe the total migration count and its evolution over time. The visualization shows a strong peak of immigration in the year after the completion of the underground.



Mode
● explore
○ analyze

Granularity
○ districts
● subdistricts

Graph
○ undirected
● directed

Time comparison

Show top n edges
20 500 1000

District color based on
○ total
○ incoming
○ outgoing
○ inherent
● none

Country of birth

Show Movement from/to
Vienna
Austria
Abroad

Movement
0 to 400
400 to 800
800 to 1,200

Movement intensity
0 to 0
0 to 0
0 to 1
1 to 1
1 to 1

Current view
10., Oberlaa
2007 - 2018

Movement statistics (in people)

Total (incl. Inherent)
10,135
Total (excl. Inherent)
9,687
Incoming
5,514
Outgoing
4,621
Inherent
448

Showing (excl. Inherent)
5,171 (51%)
Biggest flow volume (visible)
1,153
Lowest flow volume (visible)
53

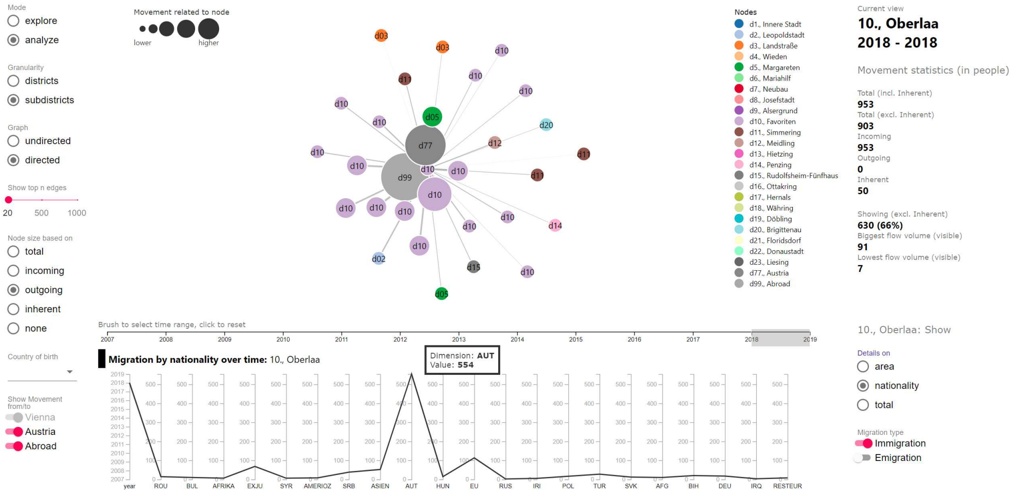
10., Oberlaa: Show

Details on
○ area
○ nationality
● total

Migration type
Immigration
Emigration

Figure 6.7: *Analyzing Oberlaa*: The bar chart shows an immigration peak one year after the completion of the U1 station and the most intense migration flows.

The evaluator decided to analyze the immigration behaviour to *Oberlaa* especially for the year 2018. By choosing the *force directed layout*, the evaluator wanted to analyze, where people in this year mostly immigrated from. The graph shows the high external immigration from abroad and the rest of Austria as well as from neighbouring sub-districts. The *parallel coordinates graph* shows the diversity regarding the country of birth. 2018, immigrants were mostly born in Austria, followed by EU, Balkan, Asia, and Turkey.



Mode
○ explore
● analyze

Granularity
○ districts
● subdistricts

Graph
○ undirected
● directed

Show top n edges
20 500 1000

Node size based on
○ total
○ incoming
● outgoing
○ inherent
○ none

Country of birth

Show Movement from/to
Vienna
Austria
Abroad

Nodes
d1., Innere Stadt
d2., Leopoldstadt
d3., Landstraße
d4., Wieden
d5., Margareten
d6., Mariahilf
d7., Neubau
d8., Josefstadt
d9., Alsergrund
d10., Favoriten
d11., Simmering
d12., Meidling
d13., Hietzing
d14., Penzing
d15., Rudolfsheim-Fünfhaus
d16., Ottakring
d17., Hernals
d18., Währing
d19., Döbling
d20., Brigittenau
d21., Floridsdorf
d22., Donaustadt
d23., Liesing
d77., Austria
d99., Abroad

Current view
10., Oberlaa
2018 - 2018

Movement statistics (in people)

Total (incl. Inherent)
953
Total (excl. Inherent)
903
Incoming
953
Outgoing
0
Inherent
50

Showing (excl. Inherent)
630 (66%)
Biggest flow volume (visible)
91
Lowest flow volume (visible)
7

10., Oberlaa: Show

Details on
○ area
● nationality
○ total

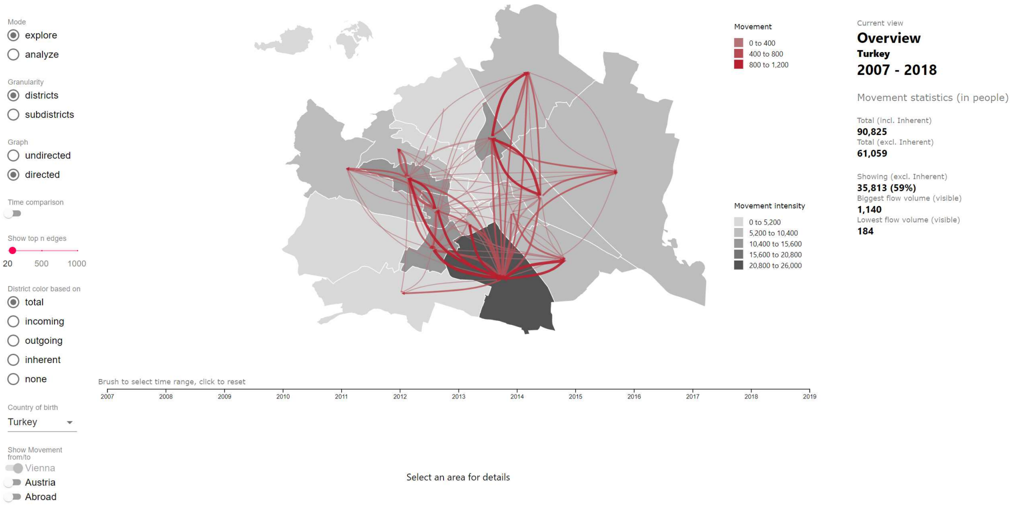
Migration type
Immigration
Emigration

Figure 6.8: *In-depth analysis*: The most correlated areas in 2018 and the diversity regarding the *country of birth*.

6.2.4 T8 - Internal migration patterns

In task *T8*, the evaluator was asked to analyze the correlation of districts and sub-districts for the *internal* migration of people born in *Austria* and *Turkey*. This complex task combines multiple elementary and synoptic tasks during the analysis.

First, the evaluator observed the migration flows on district level for Turkey. As figure 6.9 shows, the most frequent district regarding the movement is *Favoriten*, followed by *Meidling*, *Margareten*, *Rudolfsheim-Fünfhaus*, *Ottakring*, and *Brigittenau*. There are several districts, which are hardly related to the migration: Inner city districts and the districts *Währing*, *Döbling*, *Liesing*, and *Hietzing*. The evaluator observed, that high volume flows are mostly short, while far movements are less intense.



Mode
● explore
○ analyze

Granularity
● districts
○ subdistricts

Graph
○ undirected
● directed

Time comparison

Show top n edges
20 500 1000

District color based on
● total
○ incoming
○ outgoing
○ inherent
○ none

Country of birth
Turkey

Show Movement from/to
Vienna
Austria
Abroad

Current view
Overview
Turkey
2007 - 2018

Movement statistics (in people)

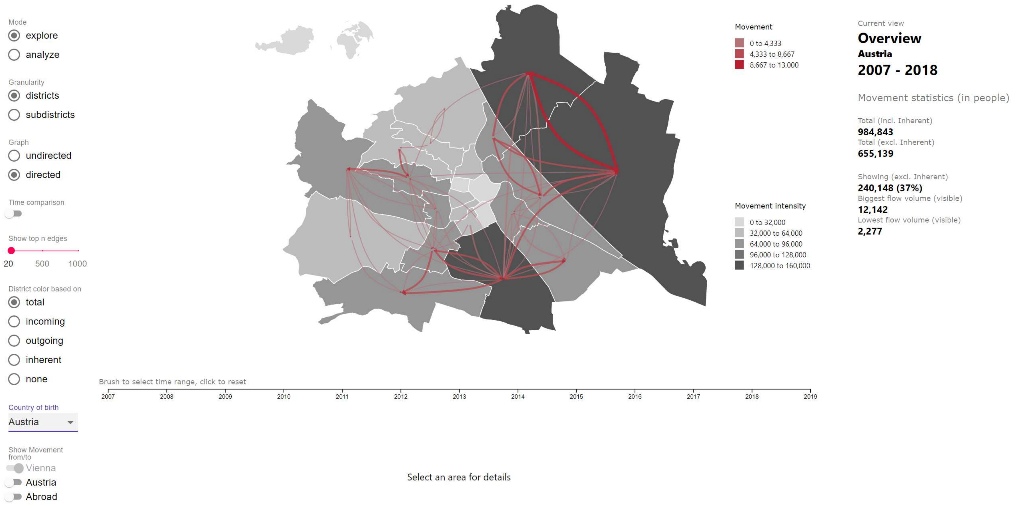
Total (incl. Inherent)
90,825
Total (excl. Inherent)
61,059

Showing (excl. Inherent)
35,813 (59%)
Biggest flow volume (visible)
1,140
Lowest flow volume (visible)
184

Select an area for details

Figure 6.9: *Migration flows on district level for people born in Turkey*: The map shows intense flows between several districts. The district *10., Favoriten* has the most intense migration.

In comparison to the migration flows of people born in Turkey, figure 6.10 shows the same configuration of the prototype, but filtered for people born in *Austria*. Although *Favoriten* has again a very high movement intensity, the district with the highest immigration volume is *Donaustadt*. Generally, the most intense migration flows are between the outer districts on the other bank of the *Danube*: *Floridsdorf* and *Donaustadt*. The evaluator mentioned a more evenly distribution over Vienna.



Mode
● explore
○ analyze

Granularity
● districts
○ subdistricts

Graph
○ undirected
● directed

Time comparison

Show top n edges
20 500 1000

District color based on
● total
○ incoming
○ outgoing
○ inherent
○ none

Country of birth
Austria

Show Movement from/to
Vienna
Austria
Abroad

Current view
Overview
Austria
2007 - 2018

Movement statistics (in people)

Total (incl. Inherent)
984,843
Total (excl. Inherent)
655,139

Showing (excl. Inherent)
240,148 (37%)
Biggest flow volume (visible)
12,142
Lowest flow volume (visible)
2,277

Figure 6.10: *Migration flows on district level for people born in Austria:* People born in Austria move extensively between *Floridsdorf* and *Donaustad* as well as *Favoriten*.

The evaluator extended the analysis and switched to the sub-district level. Figure 6.11 shows the migration flows of people born in Turkey. In this view, the evaluator observed, that the distribution followed a pattern along the *Gürtel*, with intense clusters around *Favoriten*, *Simmering*, *Ottakring*, and *Brigittenau*. By reducing the number of visible flows, the clusters became even more visible.

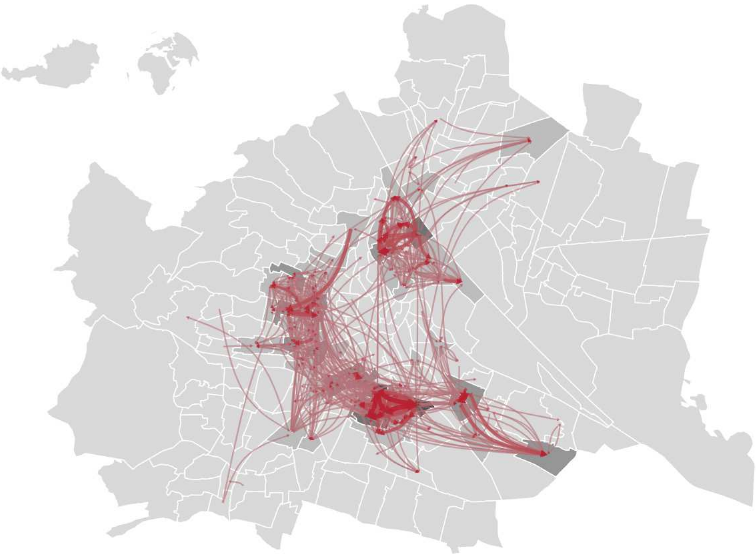


Figure 6.11: *Sub-district drill-down, Turkey:* Migration flows on sub-district level reveal an intense migration flow along the *Gürtel*. Reducing the number of visible edges reveals distinct clusters.

Figure 6.12 shows the migration flows on sub-district level of people born in *Austria*. Compared to the movement pattern of people born in Turkey, this view shows a more evenly distributed movement. Furthermore, the evaluator mentioned the many short distance flows, which indicates the preferred movement within the neighbourhood. Therefore, the movement pattern in this view reveals many smaller clusters.

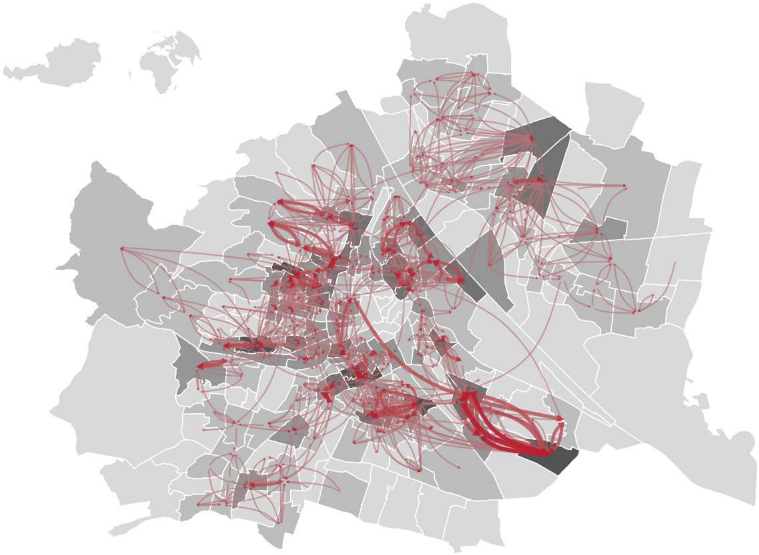


Figure 6.12: *Sub-district drill-down, Austria*: The flow map shows short, high volume edges. Long distance migration flows are less intense, which reveals many small clusters.

Based on the findings in the analysis of the migration flows, the evaluator wanted to know, how the sub-districts are correlated to each other disregarding the geographic location. Therefore, the evaluator switched to the *force directed graph* to see, if the clusters with regards to the correlation of areas are visible there as well.

Figure 6.13 shows the correlation graph for people born in *Turkey*. In this view, the evaluator's observations were confirmed, as it also shows a clear distinction of 2 to 3 correlation clusters. The evaluator mentioned, that the distinction of clusters is easier in the correlation graph, but due to the loss of the geographic location, the distance of movements is also lost. Only the equal coloring of sub-districts belonging to the same districts indicates short distance movement.

The same configuration was used again to analyze the correlation of sub-districts for people born in Austria. The correlation graph in figure 6.14 shows the many small clusters, which are highly related to the political borders of districts as well as their geographic location, confirming the observation on the map. This view indicates some strong correlations between the districts *d21 and d22*, *d2 and d20*, as well as a big cluster containing *d14, d15, d16, d17, d18, and d19.*

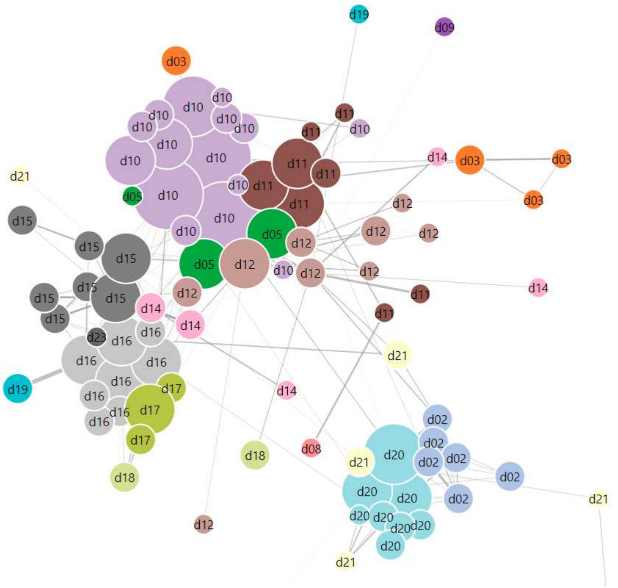


Figure 6.13: *Correlation graph, Turkey*: The force directed layout shows a distinction of 2 or 3 clusters.

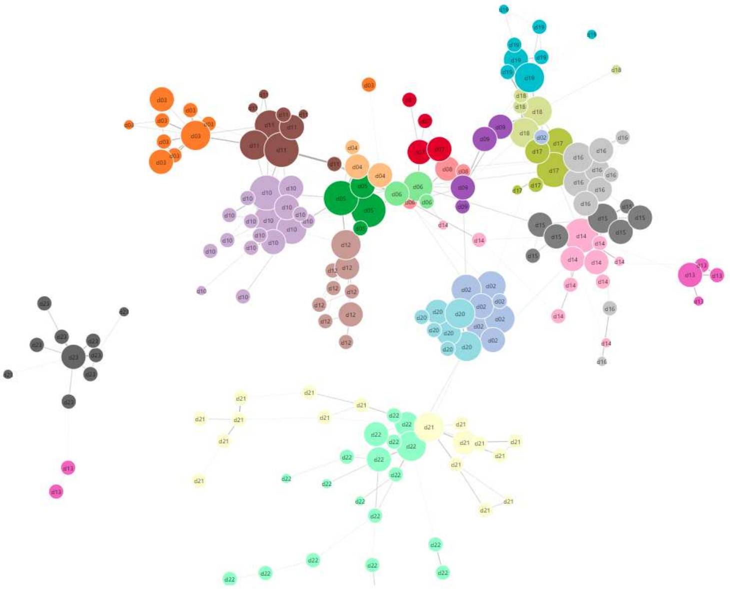
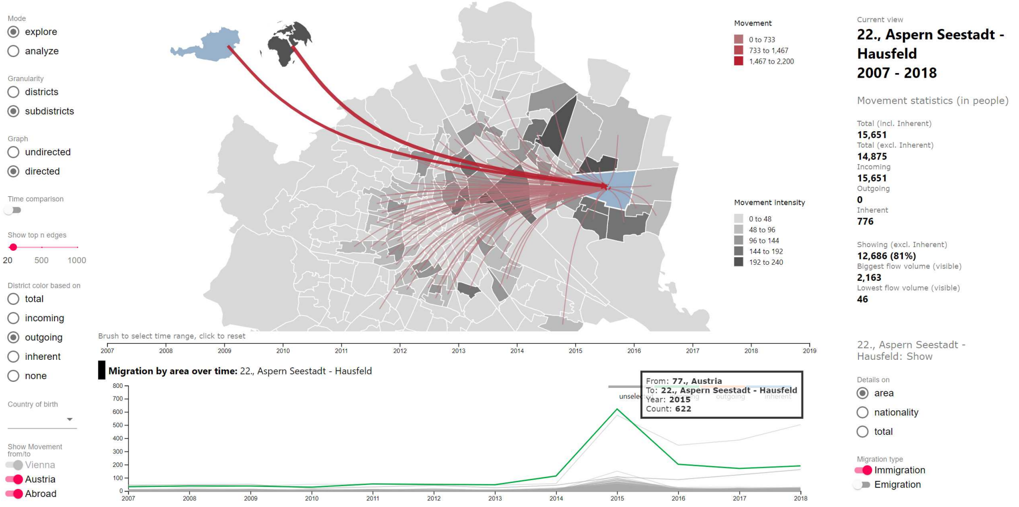


Figure 6.14: *Correlation graph, Austria*: The very homogeneous clusters regarding the political borders are an interesting detail in the migration pattern of people born in Austria.

6.2.5 T10 - Seestadt Aspern

In this task, the evaluator had to analyze the immigration to the newly built - *city within a city* - *Seestadt Aspern*. This project is one of the biggest development projects in Europe [See] with a total investment volume of 5 billion euros. The area is designed to foster around 20.000 citizens and to offer the same amount of jobs in the future and is therefore, an interesting region to analyze.

Figure 6.15 shows the migration flows of the selected sub-district *Seestadt Aspern*. The evaluator analyzed the immigration flows and quickly identified a peak in 2015, when most of the housing projects were finished. The majority of the people migrated from the rest of Austria and from abroad, as the biggest flows indicate. While the immigration from Austria stagnated in the following years, the immigration count from abroad increased.



Mode
● explore
○ analyze

Granularity
○ districts
● subdistricts

Graph
○ undirected
● directed

Time comparison

Show top n edges
20 500 1000

District color based on
○ total
○ incoming
● outgoing
○ inherent
○ none

Country of birth

Show Movement from/to
Vienna
Austria
Abroad

Movement
🟥 0 to 733
🟥 733 to 1,467
🟥 1,467 to 2,200

Movement intensity
▨ 0 to 48
▨ 48 to 96
▨ 96 to 144
▨ 144 to 192
▨ 192 to 240

Current view
22., Aspern Seestadt - Hausfeld
2007 - 2018

Movement statistics (in people)

Total (incl. Inherent)
15,651
Total (excl. Inherent)
14,875
Incoming
15,651
Outgoing
0
Inherent
776

Showing (excl. Inherent)
12,686 (81%)
Biggest flow volume (visible)
2,163
Lowest flow volume (visible)
46

Brush to select time range, click to reset

Migration by area over time: 22., Aspern Seestadt - Hausfeld

From: **77., Austria**
To: **22., Aspern Seestadt - Hausfeld**
Year: **2015**
Count: **622**

22., Aspern Seestadt - Hausfeld: Show

Details on
● area
○ nationality
○ total

Migration type
Immigration
Emigration

Figure 6.15: *Immigration to Seestadt Aspern*: The visualization shows the migration flows to *Seestadt Aspern* as well as the change over time of migration volume by origin areas.

Figure 6.16 shows the evolution of the total migration of this region, showing again the peak of immigration in 2015. Although steadily increasing after 2015, the total migration count was far less. This highlights the *initial* settlement of the finished accommodation units in 2015, as this was an important milestone in the project.

The next part of the task was to analyze the diversity regarding the *country of birth* of the people immigrating to *Seestadt Aspern*. Figure 6.17 shows the distribution over time of the geo-political entities filtered only for *immigration*. The graph shows, that the majority of people settled in the *Seestadt Aspern* in 2015 were born in Austria, followed by people born in the EU.

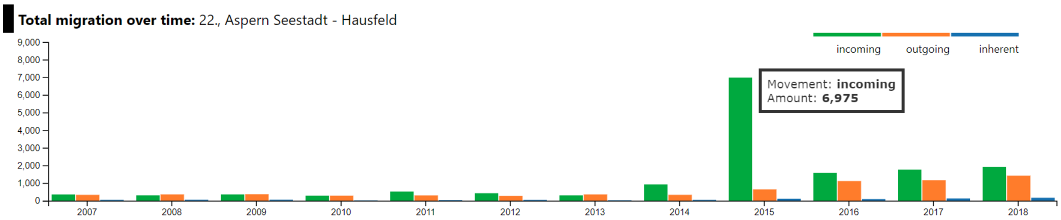


Figure 6.16: *Total migration over time*: The bar chart shows a peak in the immigration to *Seestadt Aspern* in 2015, when a lot of housing projects were finished.

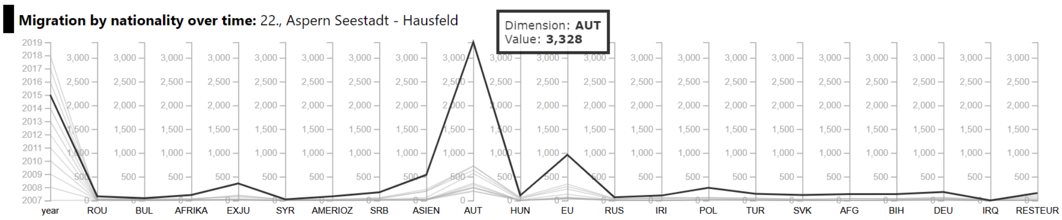


Figure 6.17: *Diversity of immigration*: The parallel coordinates graph shows the distribution of countries the immigrating people were born in. Most of the people moving to *Seestadt Aspern* were born in Austria, followed by immigrants from the EU.

The evaluator wanted to compare the diversity of immigration to *Seestadt Aspern* with the overall immigration from abroad to Vienna. Therefore, the evaluator chose to select abroad to analyze the *parallel coordinates graph*. Figure 6.18 shows a similar distribution of the countries of birth compared to the distribution in figure 6.17. The evaluator mentioned correctly, that the peak of people born in Austria is to be neglected because most of these people are immigrating *from* Austria and are not included in this selection of immigrants from abroad.

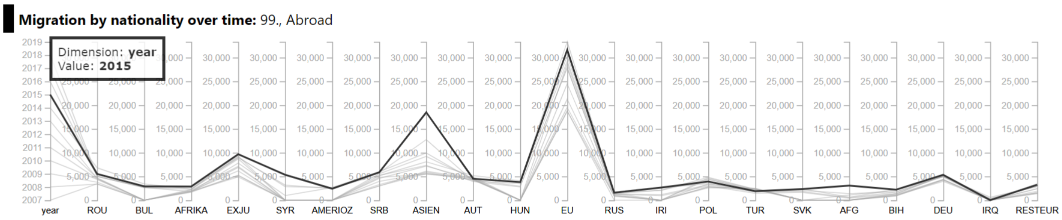
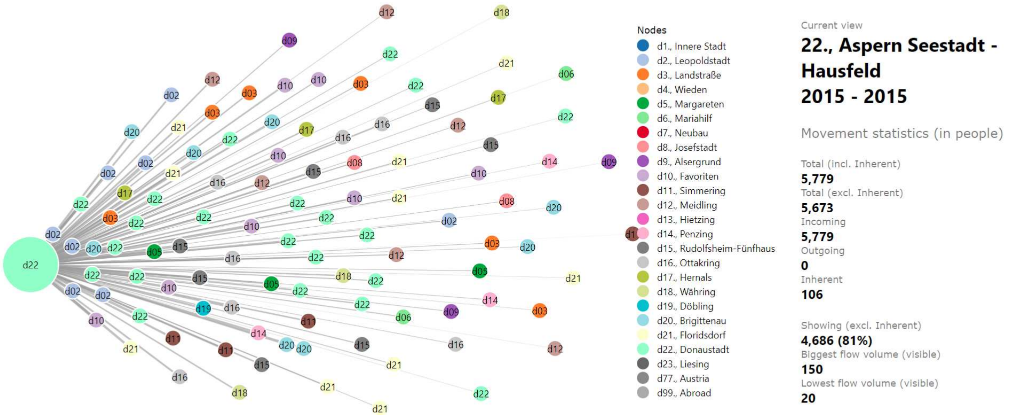


Figure 6.18: *Diversity of immigration from abroad*: The parallel coordinates graph is used to show the distribution of the country of birth of people moving to Vienna from abroad. This graph shows a distribution similar to the one shown in figure 6.17. People born in Austria should be ignored as they mostly immigrate from the rest of Austria instead of from abroad.

To analyze the immigration behaviour within Vienna, the evaluator filtered for *internal migration*. The correlation graph shown in figure 6.19 was used to determine the relation between the sub-districts Viennese citizens immigrated from and the *Seestadt Aspern*. The evaluator identified *d2, d20*, and *d22* as the districts with the highest correlation relation to *Seestadt Aspern*.



Current view

22., Aspern Seestadt - Hausfeld
2015 - 2015

Movement statistics (in people)

Total (incl. Inherent)
5,779
Total (excl. Inherent)
5,673
Incoming
5,779
Outgoing
0
Inherent
106

Showing (excl. Inherent)
4,686 (81%)
Biggest flow volume (visible)
150
Lowest flow volume (visible)
20

Figure 6.19: *Correlation graph, Austria*: The very homogeneous clusters regarding the political borders is an interesting detail in the migration pattern of people born in Austria.

6.3 Identified strengths

Throughout the evaluation sessions, the evaluators highlighted strengths of the chosen design.

Structure

Five out of the six evaluators highlighted the structure and composition of the UI components. They found the arrangement to be well structured, consistent and intuitive. The layout with the distinction of general filter to the left, data visualizations in the middle and statistics and special filter to the right were explicitly mentioned. The main attention during the analysis was drawn to the main visualization component and the time series visualization components.

Interactivity

Each evaluator commended the interaction techniques used to explore and analyze the complex data. The combination of filtering, selecting, hovering, and cross-highlighting supports the user in the process of gaining insight and understanding the migration in Vienna. The option to change the number of edges was explicitly mentioned by three evaluators as an important tool to control the visual clutter and readability of the map.

Map design

The map visualization was evaluated as an excellent form of displaying the migration data. The majority of the evaluators focused on this visualization and solved the tasks by almost only utilizing the map and the time series visualizations. The evaluators highlighted the efficiency of analyzing network flows with this visualization design paired with the interaction techniques. Furthermore, all evaluators mentioned the familiarity which lies within the utilization of geographic maps.

Force layout

In contrast to the *Map layout*, this visualization was utilized heavily by only two evaluators. The other four evaluators took the *Force directed layout* as a valuable addition to the map but mentioned, that the map layout is adequate enough to do intense analysis of migration patterns or even cluster identification. One evaluator stated, that graph layouts in general are harder to understand while people are very much accustomed to a geographical layout.

The two evaluators who used this layout more intensively, highlighted the additional possibilities to analyze the migration flows and the correlation patterns between districts and sub-districts, especially with the option to add more and more edges to the network without introducing visual clutter.

Time series visualizations

The evaluators utilized all three time series visualizations. Depending on the task, they explored and analyzed the relationships between districts in the line chart, the change over time of geopolitical entity movement in the parallel coordinates graph, and the change over time of the total movement intensity in the bar chart. Furthermore, five evaluators commended the structure to separate additional filter specific to the selected district from the general filters.

Data transparency

The visual representation of the migration flows were supported by computed values in the statistics component. Four of the six evaluators highlighted especially the numbers of the currently shown migration network to support assumptions made during the explorations and to compare them to the total movement numbers. The summary of the filter settings was also mentioned as a helpful addition to the big picture. One of the evaluators stated, that decreasing the distance between the filters and the summary by placing the text onto the main visualization components, would highlight the currently applied filters even more.

6.4 Identified weaknesses

Although the evaluators could solve all the task and rated them towards *easy to solve*, they identified minor weaknesses. Especially the tasks T3, T5, T6, T8, and T11 were rated to be more difficult to solve with the prototype.

Finding areas

The task *T5*, which demands the evaluators to analyze the impact of the extension of the underground line *U1* on the sub-district *Oberlaa*, was easy to solve for every evaluator. Nevertheless, finding this particular sub-district within the dense map was harder to achieve. Even with a high confidence in the geography of Vienna, the evaluators searched for *Oberlaa* by hovering over the sub-districts around the south of Vienna until they found it. This suggests, that a search for specific areas, especially in the sub-districts, would support the user in the analysis.

Special node selection

In task *T6* the evaluators had to find the geopolitical entity with the highest immigration from *abroad*. Two of the evaluators had minor difficulties to solve this task at first, because they forgot about the possibility to select the special nodes *Austria* and *abroad* for detailed analysis. The reason for that may lie in the abstraction of the external migration. The special nodes do not have a geographic location, they are rather depictions of the abstract concept of the *rest of Austria* or *rest of the world*. This may cause the user to believe that they are not part of the *selectable* geographic entities. A guidance system which would guide the user through the application would help to eliminate this shortcoming.

Three of the evaluators mentioned, that the point where edges terminate at the special nodes could be a bit misleading, since they actually do not represent the exact geographic location on the icons. They suggested to draw a border around the icons and to terminate the edges on this border rather to point them into the center of the icons.

Comparison with two different filter settings

The comparison tasks *T3*, *T8*, and *T9* were more difficult to solve and were rated with lower scores. Two of five evaluators found it sub-optimal to compare two scenarios with different filters by switching between them. The evaluators mentioned that it is hard to preserve the mental model during the change of the filters. The cognitive load is very high during the comparison of the different situations.

An efficient comparison of situations, where different filters have to be applied, could be achieved with alternative approaches: One possibility would be to use small multiples to compare the migration patterns side by side. This would lead to smaller versions of the map which could be compared directly instead of changing the filters to observe the other pattern. Another approach would be to freeze the current map and color the edges differently. In this scenario the user would be able to change the filters and compare directly the new pattern on top of the frozen one as an overlay. Nevertheless, there is the risk of introducing much more visual clutter and losing some readability. Additionally, the different colors to differentiate the two scenarios should be chosen very carefully.

Time Comparison feature

The task *T11* to compare the movement intensity of 2010 and 2018 was basically easy to solve. Nevertheless, the evaluators identified a possible visualization problem in the color coding of the districts with gradients. We used the same colors to encode the movement intensity as in the overview map. Small changes in the movement intensity, where the two numbers stay within the threshold of the color, are hard to identify because colors stay the same. Even if the relative change in percentage is very high, these changes, which are listed in the table to the right, are not reflected on the color coding of the districts. To overcome this problem, an additional filter could be added with which the user controls if the nodes are colored according to the relative (percentage) or absolute change of movement. This would add more data transparency as well as an additional analysis option to the prototype. One evaluator suggested that it would be possible to integrate this feature in the timeline by enabling multiple selections in the brushing.

In general, the *Time comparison* feature was the most misleading part, since this visualization was utilizing a different visual encoding and data aggregation than the other ones and left one evaluator even with the feeling of using a different piece of software. Nevertheless, all evaluators mentioned, that this feature has its justification but could be improved by choosing another encoding or even other interaction methods.

Geopolitical entities

The privacy restriction and therefore, the aggregation of the migration data for the countries which did not meet the threshold of 2000 people moving, is a complex and difficult situation. Given that one target group is the public, it is almost impossible to understand without an explanation, how the data is aggregated and what movement data is included in the geopolitical entities. The evaluation showed, that users, who have no further information about these entities, could be mislead and make wrong assumptions or conclusions on how the country of birth is structured in the data. To overcome this problem, a guidance system could be applied to showcase this aggregation. The evaluators stated, that with the introductory explanation, the relation between the countries of birth and geopolitical entities was clear.

6.5 Additional features

We discussed with the evaluators which features would be a useful addition to the prototype in order to leverage the analysis process and the acquisition of insight even more.

Multiple selection of areas

One evaluator mentioned in the process, that the selection of multiple areas would be a helpful feature to explore and analyze the migration data even further. One example of this exploration would be to select both of the special nodes to analyze the total external migration from and to Vienna.

Filtering out specific countries of birth

In the analysis of the diversity of migration flows, one evaluator wanted to filter out the movement data of people born in Austria to focus on a detailed comparison of the other countries. This could be applied with a picklist of Countries respectively geopolitical entities instead of a drop-down list.

Zooming into a district

Two evaluators tried to zoom into a specific district to analyze the movement data only for the sub-districts associated with this district. While the selection of multiple areas has the same effect, drilling down into the data of an area of interest is a common technique to support top-down analysis.

Saving filter settings, defining presets

The option to save a specific filter setting and to refer to it later on in a comparison task, can speed up the analysis and exploration process. For the public, there could be also predefined filter settings to show interesting migration patterns or the migration flows for specific events, e.g., the refugee crisis.

Map overlays

One evaluator stated that showing additional layers of the map would be an interesting extension to the geographic view. These layers could be e.g., the transportation network, recreation areas, or schools, to observe how migration patterns evolve in areas which are closer to the respective places.

CHAPTER 7

Future work

In this thesis we laid the groundwork for exploring and analyzing a multivariate, temporal, and spatial network. This prototype is a tool to analyze the migration pattern of residents of Vienna and can be used by the public to satisfy general interests as well as by city planners of the city of Vienna. Due to the multivariate aspects of the network, it is possible to extend the data by adding more attributes to either nodes or edges.

Extending node attributes

One interesting example for extending attributes of the nodes could be the cross referencing of the migration data with housing prices of the districts. This addition would allow to analyze not only *how* but also *why* people are moving the way they are. Other attributes could be the densities of public transport, educational institutions, like schools or kindergartens, or health care institutions.

Extending edge attributes

Additional attributes on the nodes, like housing prices could be encoded similar to the movement intensity as node coloring or node size and also be included in the filters.

Extending the attributes of the edges would include attributes of the groups of moving people. To analyze which migration flows follow certain patterns, considering not only the country of birth, but also the gender, age, profession, or the residency type (principal residence or second home) would give even more insight into the migration.

The additional attributes could be included in the filter to focus on specific attributes and in the parallel coordinates view with brushing through each dimension. Also, new visualizations could be added to the tool to explore other aspects of the migration flows.

Analysis through Dimensionality Reduction

As we already discussed, adding a lot of attributes to the network tremendously surges the complexity. Keeping track of every attribute and finding patterns in such complex data structures is a hard task. One approach to diminish this complexity is to apply *Dimensionality Reduction techniques*. With TimeCurves [BSH$^+$16] we showed an example of the application of such a technique. Bach et al. employed *multi-dimensional scaling (MDS)* along with the time dimension to reduce multiple datapoints to the 2D space. Another possibilty would be to apply *Principal component analysis* to reduce the data into the principal component space for further analysis, as described in the work of Aigner et al. [AMST11]. The results of *Dimesionality Reduction* could be used for the *Force directed graph layout* or feed data to new visualizations.

Another approach is to analyze the data by utilizing *Self Organizing Maps*, a type of an *Artificial Neural Network*, where a grid in a low dimensional space adapts to the characteristics of the data, forming a map. This map could be included as an additional visualization to extend the analysis possibilities even further.

We suggest that these extensions would be designed by carefully considering the *Data-User-Tasks-Triangle* by Miksch et al. [MA14], to provide optimal Visual Analytics techniques to the target audiences.

CHAPTER 8

Conclusion

This work describes the user centered design process of building a visualization for exploring and analyzing a spatial, multivariate network over time. The target audiences are - besides the public - city planners, who plan and develop the city. We lay the groundwork of a Visual Analytics tool to gain insight into the internal and external migration of residents in the city of Vienna. Due to the multiple attributes attached to either nodes or edges in this network, the visualization must justify the various aspects of the data and present it in a way that users can understand it and gain insight by exploring and analyzing the migration patterns.

This justification relies on three hypotheses of this thesis:

H1: User-centered design leads to an efficient visualization which enables the users to fulfill their tasks.

The three cornerstones of the design triangle define the structure of the data, which users are going to utilize the visualization, and which tasks are needed to be achieved with the support of the visualization. Applying this structured methodology defined the foundation for the visualization design. Based on these cornerstones, we were able to efficiently identify the fundamental requirements for the prototype. These requirements demanded a careful selection of visualization techniques to build a clear, consistent and intuitive tool which the users are able to fulfill their tasks with.

H2: Combining the right visualization techniques leverages *insight* into the different aspects of networks in time and space.

Throughout the literature search and the analysis of the migration network considering the data-user-tasks triangle, we came to the conclusion, that no visualization technique covers every aspect of the requirements on their own. Various visualizations have strengths in displaying the change over time, others efficiently display multivariate data, and others visualize a network structure in a comprehensive way.

Therefore, we took several visualization approaches, each carefully selected by their strengths and the applicability and effectiveness to tailor them to our needs. This combination proved to be very effective and amplified the expressiveness of the data.

We chose a *Geographical Map Layout* to reflect the spatial aspect of the network. The edge design proved to be crucial to the readability and usability of this representation. The nodes, as well as the edges, were suitable entities to encode additional data onto the map, like movement intensities. We chose a *Force Directed Graph Layout* to visualize the relationships and correlation between districts disregarding the geographic location. Attraction and repulsion forces moved the nodes based on a physical model of particles to form a correlation graph. The temporal dimension was encoded in three different time series visualizations of which each focused on one of the multivariate aspects of the data. The expert evaluation showed that the combination of different layouts enabled many and various effective analysis options for the users.

H3: Interactive methods in the visualization help to overcome the problems with the complexity of time-oriented multivariate networks.

A multivariate network with temporal and spatial aspects forms a very complex structure. The prototype needs to address the possibility to observe and analyze every aspect of this complexity, giving insight into details as well as exploring the network as a whole. We combined the chosen visualizations by applying common interaction techniques, allowing to cross-reference data points across different visualizations, highlighting points of interest based on user-made selections, filtering for every aspect of the data, and therefore, providing a top-down analysis from overview to details. This interactivity added the flexibility to the exploration and analysis process, so that the user can gain the desired insight. The expert evaluation showed that the interactivity was essential to solve the predefined tasks. The possibility to observe a point of interest from different perspectives enables manifold ways to obtain the understanding of the data.

The combination of applying an efficient methodology, interconnecting different visualization and interaction techniques facilitates the exploration and analysis of the complex nature of multivariate network over time and space. This flexibility offers many possibilities to extend the network by additional attributes to either nodes or edges and therefore, enrich the analysis process even further.

There is no single truth in the exciting field of *Visual Analytics* and every visualization technique has its advantages and disadvantages. A careful design process is essential to visualize the data in an effective, expressive and appropriate way. We conclude our work with the following very accurate quote:

> The whole is greater than the sum of its parts.
>
> *Aristotle*

List of Figures

2.1 Multivariate networks. A taxonomy of different layouts to visualize multivariate networks (MVNs) [NSML19a]. 6
2.2 Multivariate networks. A taxonomy of operations applied to different levels in a network [NSML19a]. 7
2.3 Straight vs. curved edges. Flow map shows better readability with *curved*(left) than with *straight*(right) lines [JSM+18]. 11
2.4 Encoding direction. The Map shows better readability with *arrowheads* (left) than with *tapered lines* (right)[JSM+18]. 12
2.5 Edge termination. Flows terminating at an *anchor point* (left) and somewhere in the *areas* (right)[JSM+18]. 13
2.6 Encoding intensity. The flow volume on the left is encoded only in the *line width*. On the right, the volume is encoded with *line width and color* [JSM+18]. 13
2.7 Hierarchical edge bundling. Radial network layout without bundling (left) and utilizing hierarchical edge bundling (right) [Hol06]. 14
2.8 Tree layout. Common tree layouts for the use of hierarchical edge bundling. From left to right: rooted tree, radial tree, baloon tree, treemap layout [Hol06]. 14
2.9 Edge bundling example. Edge bundling applied to flight connections in the United states on a geographical map [Fli]. 15
2.10 *OD-Maps*. Travel-to-work OD-map of the US-State Ohio [WDS10]. 15
2.11 *Flowstrates*: Two maps combined with a heat map to display the change over time [BBBL11]. 16
2.12 Visualizing univariate data. Examples of a LinePlot (left) and a BarGraph (right) [AMST11]. 19
2.13 *Sparklines*. Enrichment of stock marked data by integrating small visualizations into text [AMST11]. 19
2.14 *ThemeRiver*. Multivariate data attributes visualized in form of a width-changing river. The graph shows the topic relevance in the media from 1959 to 1961 [HHN00]. 20
2.15 *Data Tube*. 3D visualization technique where the viewer walks along the timeline and observes the change of data attributes along the walls. This example shows stock market development [Ank01]. 21
2.16 *Trendalyzer*. Dynamic visualization through animation of the life expectancy vs. income per person over time by Gapminder [Gap]. 22

2.17 *Small multiples.* Visualizing change over time with the arrangement of miniature versions of the visualization of migration data on a map [AMST11]. . . 22
2.18 *Data Vases.* Vertical plots for each attribute show the time series data of crime rates in counties of North Carolina U.S. [TR09]. 23
2.19 *Time curves.* Folding a timeline of data cases according to similarity. Similar cases are drawn to each other [BSH+16]. 24
2.20 *Time Curves example.* Evolution of the Wikipedia article on Palestine with an edit war at point C [BSH+16]. 24

3.1 *Design triangle*: Data, Users, and Tasks as the dominant factors in the design of interactive visualizations [MA14]. 27

4.1 *Visualization Layout*: Arrangement of the various visualization components. 34
4.2 *Main visualization component*: Geographical representation of Vienna. . . 35
4.3 geoJSON feature: Geographical shape of the 17th district, *Hernals*. 37
4.4 Special nodes: Geographical shape of the additional nodes representing the rest of Austria (left) and the rest of the world (right). 38
4.5 Node color coding: Districts are filled based on their total movement metric. The darker a district, the more people are moving from, to, and within the district. 40
4.6 *Encoding direction*: 11.142 people moving from the 22nd to the 21st district. The edge is highlighted when hovering over with the mouse. The arrowhead displays the direction. 41
4.7 *Curvature:* Straight edges. 42
4.8 *Curvature:* Applying curved edges. 42
4.9 *Edge design*: Circular offset at the source and target points improve readability. 42
4.10 *Time comparison feature*: The incoming movement intensities for each subdistrict is shown as a gradient fill from left for 2015 to right for 2016. . . 43
4.11 *Visual clutter problem*: Displaying many edges on the map reduces the information gain of edges and nodes. 44
4.12 *Analyze mode*: The Force-Layout replaces the map layout. 45
4.13 *Node Labelling and Coloring*: Color scheme and labelling for the 23 districts plus *Austria* and *abroad*. 47
4.14 *Color coding of sub-districts*: Sub-districts of the same districts have the same color and label. 47
4.15 *General filter component*: Filtering the data is positioned in the left grid column. 49
4.16 *Selecting the time range*: Brushing to select a time range. 53
4.17 Filtering the time dimension. A brush on a timeline offers the selection of the desired time range, in this particular example 2010 - 2014. 53
4.18 *Time series visualizations*: Three different visualizations let the user analyze the change over time by different aspects. 54

4.19 *Time series: Migration by area over time.* The line graph shows the change over time of the movement volume between *Favoriten* and other districts. Each line represents a migration relation with another district. The highlighted line shows the immigration from abroad. 55

4.20 *Line coloring:* Three different colors depict the migration types: immigration (incoming), emigration (outgoing), and inherent (migration within the selected area). 55

4.21 *Time series: Immigration to Seestadt Aspern.* The immigration peaked in 2015, when a milestone of the project was reached, resulting in a high immigration from Austria, stagnating from 2016 on. 56

4.22 *Composition of countries of birth.* A parallel coordinates graph shows the migration volume of each country or geopolitical entity migrating to *6., Mariahilf*. 57

4.23 *Composition of countries of birth.* A parallel coordinates graph shows the migration volume of each country or geopolitical entity migrating to *20., Brigittenau*. 57

4.24 *Migration volume over time.* The grouped bar chart shows the time-oriented data of the migration volume of *17., Hernals*. 58

4.25 *Special filtering*: If a node is selected, the user has additional possibilities to filter the time series data and to choose from different time-series-visualizations. 58

4.26 *Cross-highlighting.* The screen shows the migration in the selected 3rd district *Landstraße*. Hovering over the region *abroad* highlights the migration flows corresponding to both regions in the line chart. 60

4.27 *Node tooltip.* Showing the attributes of a node. 60

4.28 *Edge tooltip.* Showing the attributes of an egde. 60

4.29 *Time series tooltip.* Showing the attributes of a data point. 60

4.30 *Show me the numbers*: Visual anchor for the user to analyze numbers. . . . 61

4.31 *Selected district.* The component shows the currently selected district *19., Döbling* along the selected time range, as well as detailed information about how many people were migrating and what parts of this migration are visible. 62

4.32 *Overview.* Currently, the overview is displayed in the visualization. Detailed information about how many people were migrating and what parts of this migration are visible is shown below. 62

4.33 *Time comparison*: The statistics component shows the top 15 regions with the biggest relative change between the first and last year of the selected time range. 63

5.1 *Migration records*: Excerpt from the table which stores the migration data. 66

6.1 *Elementary comparison*: The specific district coloring, based on the immigration count of each district, reveals *Favoriten* as the district with the highest immigration. 82

6.2 *Elementary comparison*: The bar chart for the total migration over time of *Favoriten* shows a steady increase of the green bars until 2018, when the number of immigrating people slightly decreased. 82
6.3 *Inherent movement over time:* Hovering over the selected district highlights the movement within *Simmering* over time. 83
6.4 *External migration*: Hovering over Austria highlights the movement between *Simmering* and the *rest of Austria*. The biggest flow indicates the highest movement volume of people leaving *Simmering*. 84
6.5 *Internal immigration:* Movement behaviour of people moving to *11., Simmering*. The biggest flow shows the high immigration from *Favoriten*. 84
6.6 *Internal emigration:* Movement behaviour of people leaving *11., Simmering*. The biggest flow shows the high emigration towards *Favoriten*. 84
6.7 *Analyzing Oberlaa*: The bar chart shows an immigration peak one year after the completion of the U1 station and the most intense migration flows. . . 85
6.8 *In-depth analysis*: The most correlated areas in 2018 and the diversity regarding the *country of birth*. 85
6.9 *Migration flows on district level for people born in Turkey*: The map shows intense flows between several districts. The district *10., Favoriten* has the most intense migration. 86
6.10 *Migration flows on district level for people born in Austria:* People born in Austria move extensively between *Floridsdorf* and *Donaustad* as well as *Favoriten*. 87
6.11 *Sub-district drill-down, Turkey:* Migration flows on sub-district level reveal an intense migration flow along the *Gürtel*. Reducing the number of visible edges reveals distinct clusters. 87
6.12 *Sub-district drill-down, Austria*: The flow map shows short, high volume edges. Long distance migration flows are less intense, which reveals many small clusters. 88
6.13 *Correlation graph, Turkey*: The force directed layout shows a distinction of 2 or 3 clusters. 89
6.14 *Correlation graph, Austria*: The very homogeneous clusters regarding the political borders are an interesting detail in the migration pattern of people born in Austria. 89
6.15 *Immigration to Seestadt Aspern*: The visualization shows the migration flows to *Seestadt Aspern* as well as the change over time of migration volume by origin areas. 90
6.16 *Total migration over time*: The bar chart shows a peak in the immigration to *Seestadt Aspern* in 2015, when a lot of housing projects were finished. . . . 91
6.17 *Diversity of immigration*: The parallel coordinates graph shows the distribution of countries the immigrating people were born in. Most of the people moving to *Seestadt Aspern* were born in Austria, followed by immigrants from the EU. 91

6.18 *Diversity of immigration from abroad*: The parallel coordinates graph is used to show the distribution of the country of birth of people moving to Vienna from abroad. This graph shows a distribution similar to the one shown in figure 6.17. People born in Austria should be ignored as they mostly immigrate from the rest of Austria instead of from abroad. 91

6.19 *Correlation graph, Austria*: The very homogeneous clusters regarding the political borders is an interesting detail in the migration pattern of people born in Austria. 92

List of Tables

3.1 Data attributes 28
3.2 Time attributes 29

4.1 Migration count for Syrian people. The movement data in the colored cells are aggregated to *Asia* due to privacy restrictions. 51

6.1 Task-Requirement mapping 79
6.2 The evaluators confidence scores. 80
6.3 The evaluators impressions of the prototype, taken after step 1 and revised at the end of the session. The prototype was rated to be useful, quite easy to use, rich on features and fast. 80
6.4 Solving rate of the tasks. 81
6.5 Difficulty rating of the tasks. It shows, that the task T11 was rated the hardest with regards on how much the prototype supported the user. 81

Bibliography

[AA06] Natalia Andrienko and Gennady Andrienko. *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer Science & Business Media, 2006.

[AMST11] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of Time-Oriented Data*. Springer London, 2011.

[Ank01] Mihael Ankerst. Visual data mining with pixel-oriented visualization techniques. In *Proceedings of the ACM SIGKDD Workshop on Visual Data Mining*, 2001.

[BBBL11] Ilya Boyandin, Enrico Bertini, Peter Bak, and Denis Lalanne. Flowstrates: An approach for visual exploration of temporal origin-destination data. *Computer Graphics Forum*, 30(3):971–980, 2011.

[BSH$^+$16] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):559–568, 2016.

[CMS99] Stuart Card, Jock Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision To Think*. 01 1999.

[Col] Colorbrewer 2.0, color advice for cartography. `http://colorbrewer2.org/#` accessed March 3, 2020.

[CT05] Kristin A Cook and James J Thomas. Illuminating the path: The research and development agenda for visual analytics. 2005.

[D3A] D3 api documentation. `https://github.com/d3/d3/blob/master/API.md` accessed March 12, 2020.

[D3F] Force-directed graph layout in d3. `https://github.com/d3/d3-force` accessed March 12, 2020.

[D3J] d3.js. `https://d3js.org/` accessed November 21, 2019.

[FJ10] Camilla Forsell and Jimmy Johansson. An heuristic set for evaluation in information visualization. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 199–206, 2010.

[Fli] Flight path connection visualization. https://bl.ocks.org/sjengle/2e58e83685f6d854aa40c7bc546aeb24 accessed November 29, 2019.

[Gap] Trendalyzer by gapminder, gapminder foundation. https://www.gapminder.org/tools/?from=world accessed November 29, 2019.

[Geo] Geojson format specification. https://tools.ietf.org/html/rfc7946 accessed February 28, 2020.

[HHN00] S. Havre, B. Hetzler, and L. Nowell. Themeriver: visualizing theme changes over time. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, pages 115–123, Oct 2000.

[HIvF11] D. Holten, P. Isenberg, J. J. van Wijk, and J. Fekete. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *2011 IEEE Pacific Visualization Symposium*, pages 195–202, March 2011.

[Hol06] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE transactions on visualization and computer graphics*, 12:741–8, 09 2006.

[JSM$^+$18] Bernhard Jenny, Daniel M. Stephen, Ian Muehlenhaus, Brooke E. Marston, Ritesh Sharma, Eugene Zhang, and Helen Jenny. Design principles for origin-destination flow maps. *Cartography and Geographic Information Science*, 45(1):62–75, 2018.

[JSO] The javascript object notation (json) data interchange format specification. https://tools.ietf.org/html/rfc7159 accessed February 28, 2020.

[KAF$^+$08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual analytics: Definition, process, and challenges. 03 2008.

[KPW14] Andreas Kerren, Helen C. Purchase, and Matthew O. Ward. *Multivariate Network Visualization: Dagstuhl Seminar #13201, Dagstuhl Castle, Germany, May 12-17, 2013, Revised Discussions.* Springer International Publishing, 2014.

[MA2] Ma 23 - wirtschaft, arbeit und statistik. https://www.wien.gv.at/kontakte/ma23/ accessed November 21, 2019.

[MA14] Silvia Miksch and Wolfgang Aigner. A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data. *Computers & Graphics, Special Section on Visual Analytics*, 38:286–290, 2014.

[Nie94] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann, 1994.

[NSML19a] Carolina Nobre, Marc Streit, Miriah Meyer, and Alexander Lex. The state of the art in visualizing multivariate networks. *Computer Graphics Forum (EuroVis '19)*, 38:807–832, 2019.

[NSML19b] Carolina Nobre, Marc Streit, Miriah Meyer, and Alexander Lex. The state of the art in visualizing multivariate networks. *Computer Graphics Forum (EuroVis '19)*, 38:807–832, 2019.

[OGDa] Katalog bezirksgrenzen wien. `https://www.data.gv.at/katalog/dataset/2ee6b8bf-6292-413c-bb8b-bd22dbb2ad4b` accessed February 28, 2020.

[OGDb] Katalog zählbezirksgrenzen wien. `https://www.data.gv.at/katalog/dataset/e4079286-310c-435a-af2d-64604ba9ade5` accessed February 28, 2020.

[OGDc] Open government data. `https://www.data.gv.at/` accessed November 21, 2019.

[Pos] Postgresql: The world's most advanced open source relational database. `https://www.postgresql.org/` accessed March 12, 2020.

[Rea] React, a javascript library for building user interfaces. `https://reactjs.org/` accessed March 12, 2020.

[See] aspern, die seestadt. `https://www.aspern-seestadt.at/` accessed May 24, 2020.

[SFD15] Beatriz Sousa Santos, Beatriz Quintino Ferreira, and Paulo Dias. Heuristic evaluation in information visualization using three sets of heuristics: An exploratory study. In Masaaki Kurosu, editor, *Human-Computer Interaction: Design and Evaluation*, pages 259–270, Cham, 2015. Springer International Publishing.

[Shn96] Ben Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[Sta] Statistik austria. `https://www.statistik.at/` accessed November 21, 2019.

[TR09] Sidharth Thakur and Theresa-Marie Rhyne. Data vases: 2d and 3d plots for visualizing multiple time series. 11 2009.

[Tuf83] Edward R Tufte. The visual display of quantitative information.(p. 197). *Cheshire, Conn.(Box 430, Cheshire 06410)*, 1983.

[WDS10] Jo Wood, Jason Dykes, and Aidan Slingsby. Visualisation of origins, destinations and flows with od maps. *The Cartographic Journal*, 47(2):117–129, 2010.

[ZC06] Torre Zuk and Sheelagh Carpendale. Theoretical analysis of uncertainty visualizations. In *Visualization and data analysis 2006*, volume 6060, page 606007. International Society for Optics and Photonics, 2006.