

Interactive visualization of dynamic multivariate networks

van den Elzen, S.J.

Published: 18/11/2015

Document Version

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

Citation for published version (APA):

Elzen, van den, S. J. (2015). Interactive visualization of dynamic multivariate networks Eindhoven: Technische Universiteit Eindhoven

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

INTERACTIVE VISUALIZATION *of* DYNAMIC MULTIVARIATE NETWORKS

Stef van den Elzen

INTERACTIVE VISUALIZATION *of* DYNAMIC MULTIVARIATE NETWORKS

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van
de rector magnificus, prof.dr.ir. F. P. T. Baaijens,
voor een commissie aangewezen door het College
voor Promoties in het openbaar te verdedigen op
woensdag 18 november 2015 om 16:00 uur

door

Stefano Johannes van den Elzen

geboren te Nijmegen

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:


voorzitter: prof.dr. J. de Vlieg
promotor: prof.dr.ir. J.J. van Wijk
co-promotor: dr.ir. D.H.R. Holten (SynerScope BV)
leden: prof.dr. H. Hauser (University of Bergen)
 prof.dr.ir. R. van Liere
 prof.dr. S. Miksch (Vienna University of Technology)
 prof.dr. J.B.T.M. Roerdink (RUG)
 prof.dr. B. Speckmann

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Colophon



The work in this dissertation was financially supported by SynerScope B.V. and has been carried out under the auspices of the research school ASCI (Advanced School for Computing and Imaging). ASCI dissertation series number 333.

Typeset with Xe_{La}TeX (TeX live 2013/Debian)
Sans-serif font (headings): Cabin
Serif font (main): Linux Libertine 

Cover Design: Stef van den Elzen
Printing: Gildeprint Drukkerijen, Enschede



Copyright ©2015 Stef van den Elzen. All rights are reserved.
Reproduction in whole or in part is prohibited without the written consent of the copyright owner.



A catalogue record is available from the Eindhoven University of Technology Library ISBN: 978-90-386-3942-0

Contents

	Page
Colophon	v
Contents	vii
Preface	xiv
1 Introduction	1
1.1 Motivation	2
1.2 Objective	4
1.3 Outline & Contributions	5
1.4 Publications	7
2 Background	9
2.1 Introduction	10
2.2 Visualization	10
2.3 Information Visualization	11
2.4 Visual Analytics	13
2.5 Network Visualization	14
2.6 Dynamic Multivariate Network Model	20
2.7 Taxonomy	21
3 Small Multiples, Large Singles	27
3.1 A New Approach for Visual Data Exploration	28
3.2 Introduction	28
3.3 Related Work	30
3.4 Small Multiples, Large Singles	31
3.5 Evaluation	37
3.6 Scalability	42
3.7 Conclusions	43

4	Multivariate Network Exploration and Presentation	45
4.1	From Detail to Overview via Selections and Aggregations	46
4.2	Introduction	46
4.3	Related work	48
4.4	From detail to overview	49
4.5	Detail view	50
4.6	Selections	51
4.7	High-level infographic-style overview	56
4.8	Examples and Use cases	57
4.9	Discussion and Limitations	64
4.10	Conclusions	66
5	Massive Mobile Phone Data	69
5.1	Exploration and Analysis of Massive Mobile Phone Data	70
5.2	Introduction	70
5.3	Design Principles	71
5.4	Related Work	73
5.5	Visual Analytics Approach	73
5.6	Use cases	79
5.7	Conclusions	88
6	Massive Sequence Views	91
6.1	Dynamic Network Visualization with Extended MSVs	92
6.2	Introduction	92
6.3	Related work	94
6.4	Definitions and features	95
6.5	Reordering techniques	99
6.6	Circular Massive Sequence Views	106
6.7	Extending the model	106
6.8	Use case	108
6.9	Limitations and Workarounds	111
6.10	Conclusions	114
7	Reducing Snapshots to Points	117
7.1	A Visual Analytics Approach to Dynamic Network Exploration	118
7.2	Introduction	118
7.3	Related Work	119
7.4	Reducing Snapshots to Points	121
7.5	Use Cases	129
7.6	Discussion	136
7.7	Conclusions	137

8	Conclusions	139
8.1	Conclusions	140
8.2	Reflections	145
8.3	Future Work	149
	Bibliography	174
	List of Figures & Tables	179
	Summary	180
	Curriculum Vitæ	183
	Index	186

Preface

FOUR years ago, when I was finishing my master's thesis, I had mixed feelings. I was happy that soon I would be given an engineering title but simultaneously I realized that doing research would stop. It was not until my master's project that I became aware of how much I actually loved doing research. This was enabled and strengthened due to the pleasant cooperation with my supervisor prof.dr.ir. Jarke J. van Wijk who was supportive, bright, enthusiastic, and had a great sense of humor. So the next logical step to me was to extend my stay at the visualization group and pursue a doctoral degree in this area. During that same time a spin-off company, SynerScope, was started by Danny Holten and Jan-Kees Buenen focusing on developing tools and techniques for the analysis of dynamic networks. I am grateful they offered me a position in their start-up and would sponsor a PhD position under the guidance of Jack van Wijk (TU/e), Danny Holten (SynerScope) and Jorik Blaas (SynerScope) which, as would later turn out, are amongst the most intelligent people I have ever met. This position meant I would get to experience industry, academia, and start-up practice. I could not refuse such a great opportunity and have enjoyed every bit of it!

First and foremost I want to thank my promoter, prof.dr.ir. Jarke J. van Wijk, Jack, thank you for your never-ending enthusiasm, creativity, and support. We both share the desire (or curse) for high standards and perfectionism. I soon learned these are no superfluous characteristics in a competitive research community, quite the opposite. Jack, you were a true mentor and our enjoyable collaboration resulted in some high quality work. It has been a great experience.

On a similar note, I would like to thank my SynerScope co-promoter and supervisors dr.ir. Danny Holten and dr.ir. Jorik Blaas for challenging and inspiring me, providing feedback, and the many insightful and mentally draining brainstorm sessions on network visualization. I am happy I could learn from the best!

I thank prof.dr. Helwig Hauser (University of Bergen, Norway), prof.dr.ir. Robert van Liere (CWI & TU/e), prof.dr. Silvia Miksch (Vienna University of Technology, Austria), prof.dr. Jos Roerdink (Rijksuniversiteit Groningen), and prof.dr. Bettina Speckmann (TU/e) for accepting the invitation to join the thesis committee and form the opposition.

The visualization group at Eindhoven University of Technology has always been a nice working environment with plenty serious and not so serious discussions and I want to thank the current and former members that were part of the group during my stay:

Michel Westenberg, Huub van de Wetering, Andrei Jalba, Robert van Liere, Meivan Cheng, Jing Li, Niels Willems, Miekeal Verschoor, Kasper Dinkla, Roeland Scheepens, Paul van der Corput, Bram Cappers, Martijn van Dortmont, and Alberto Corvò.

Likewise, SynerScope also provided a nice working environment with great people. I thank both former and current employees for brainstorm sessions, advice, expertise, and fun inspiring discussions: Jan-Kees Buenen, Danny Holten, Niels Willems, Jorik Blaas, Bart van Arnhem, Wiljan van Ravensteijn, Martijn van Dortmont, Pieter Stolk, Thomas Ploeger, Willem van Hage, Tessel Boogaard, Jesper Hoeksema, Zahra Parvaneh, Helen Gissing, Marieke Beijssens, Sylvia Wijshijer, Phil Loewen, Monique Hesseling, Freddy Nurski, Eric Elsackers, Dave Dekkers, Paul Buyink, Amanda Heithuis, Greg Cooke, Omer Einhorn, Jennemieke Poodt, Peter Schaafsma, Richard Guha, Andrew Marane, Rolf Smit, and Erik Stabij. I especially want to mention and thank Jan-Kees Buenen for being flexible and patient with me during the final stage of writing my PhD thesis. Furthermore, I want to thank SynerScope for all the experiences that would not have crossed my path had I chosen a purely academic PhD position. I enjoyed the trainings, customer interaction, exhibitions, talking to patent lawyers, on-site data analysis *et cetera*.

Furthermore, I thank the members of the MIT Prince of Wales fellows and Sensemaking Fellowship for nice and successful collaborations: Steve Chan, Simone Sala, Robert Spousta, Anna Miao, Charles Atencio, Juhee Bae, Adam Hollick, and Alison Kuzmickas. One person I would like to thank in particular is Steve Chan for nice discussions and making my visit to Boston a great experience.

Non-work related thanks go out to the magic the gathering players at the GameForce for the Friday nights and the occasional Sundays which were a welcome distraction.

Finally, I want to thank my friends and family for their continued support. In particular Hans & Petra, Rik & Pleun, Gerrie, and Martijn & Anique. Pap, bedankt voor de steun door de jaren heen en dat je me geleerd hebt om nooit op te geven en overal het beste uit te halen. Mam, bedankt voor alle steun, gezelligheid, liefde en goede zorgen!

This final spot is reserved for my beloved girlfriend Ester, she, without a doubt, helped me the most by being herself: supportive, bright, understanding, caring and loving. Ester, thank you for sharing your life with me!

And for all those I failed to mention: Thank you!

Stef van den Elzen
Eindhoven, June 2015



Introduction

1

1.1 Motivation

NETWORKS describe the *relations* between *objects*. With networks we can model complex physical and non-physical phenomena that occur in the world around us (see Figures 1.1, 1.2). Some examples of non-physical networks are financial networks where money is transferred (relation) between bank accounts (objects); e-mail networks where e-mails are sent (relation) between e-mail accounts (objects); and online social-networks with friendships between persons. Some examples of physical networks are transportation networks where goods are transported between companies; (tele-)communication between persons; migration of people between cities; airplanes flying between airports, *et cetera*.

In a business or engineering setting, the task of an analyst is often to improve or optimize networks or the underlying processes. In order to improve the underlying structure or procedures supported by the network we first need to understand the network. Understanding can be achieved by building knowledge via the gathering of insights. If we do not know what we are looking for — and often we do not know this in advance — insights can be obtained from (visual) exploration of the network. However, in general these networks are large, in the order of hundreds to thousands of relations between hundreds of thousands of objects. This makes exploration a real challenge, and finding a visual representation for large networks with suitable interaction techniques that enable exploration is a non-trivial task.

1.1.1 Multivariate Networks

Networks often contain more information than just the objects and the relations between them, which further increases complexity. We call such a network a *multivariate* network: besides the topological structure of the network, multivariate data attributes on the objects and relations are available. For example, in case of a company e-mail network we know data attributes of the persons (objects) involved, like age, gender, and current job title. We also have more information about the e-mails (relations) such as time-sent, subject, header-information, and body-text. The exploration and analysis of large multivariate networks is still a challenge. Current methods are focused on either the structural aspects of the multivariate network, or the

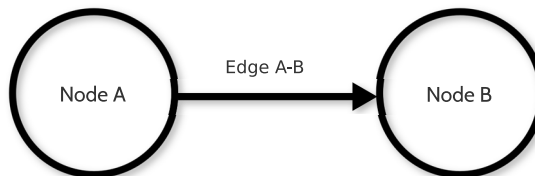


Figure 1.1: Network model with different elements: objects (nodes) and relations (edges) visually represented by circles and arrows between them.

multidimensional data associated with the objects and relations. However, we believe the greatest insights are gained from simultaneous exploration, as the two might be correlated or influence each other. For example, we could study who is e-mailing to whom (structure) or whether females or males are communicating more (multivariate data), but we might be more interested in whether females are communicating more with females or more with males, between which departments most communication takes place and what the distribution is over time (both structure and multivariate data). For this we need to be able to inspect the attributes in context of the underlying network topology and *vice versa*.

1.1.2 Dynamic Networks

In general, networks are rarely static; they are dynamic, meaning that the structure and/or associated multivariate data change over time. All networks mentioned in the previous paragraphs are examples of dynamic networks. Understanding the evolution of dynamic networks is a challenge. Next to the structural properties of the network, such as *communities* and *motifs*, we are interested in temporal properties and patterns such as *trends*, *periodicity*, *temporal shifts*, and *anomalies*. Time could simply be an additional attribute in the multivariate network. However, time is perceived differently and visualizations could benefit from methods that exploit this.

Also, typical insights to be gained are the discovery of states in the dynamic network that characterize the evolution of the network. The identification of *stable states*, *recurring states*, *outlier states*, and the *transitions* between these states helps in understanding the network. For example, the network can change gradually from one state to another, it could alternate between multiple states, or it might not be stable at all. An approach for the identification of states, temporal patterns, and obtaining insights in the evolution of the network in general is needed.

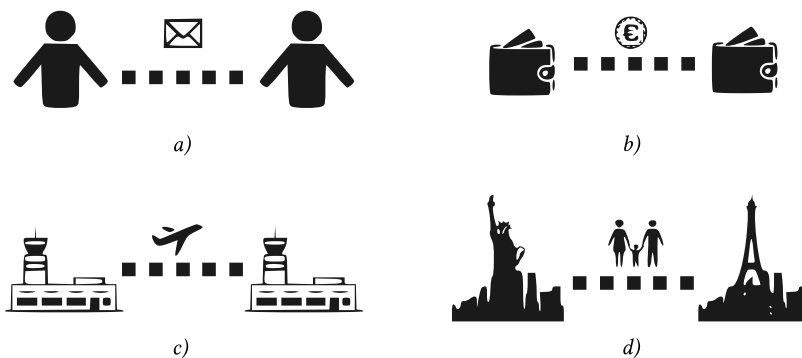


Figure 1.2: Real-world dynamic multivariate networks: a) e-mail messages that are sent between persons; b) financial transactions between bank accounts; c) airplanes flying between airports; and d) the migration of people between cities. All these networks evolve over time and have associated multivariate attributes.

1.1.3 Interactive Visual Analytics

One method that enables the exploration of large *dynamic multivariate networks* as described above is by computing metrics using techniques from statistics and data mining. Some example metrics that describe the network are the number of nodes and edges, network density, and degree distribution. Such measures provide high-level summaries of the network. However, we believe that purely automatic methods, such as these, fall short due to aggregation of results, lack of user steering with domain knowledge, and loss of context. Furthermore, automatic methods are often highly focused and designed for one specific task that is to detect the expected, not allowing for exploration to discover unexpected patterns [260]. Similarly, purely visual methods fall short due to scalability issues; in general networks are large and screen space is limited. This can partially be overcome by interaction techniques such as zoom, pan and filtering of the data. However, this leaves less apparent, more complex patterns in the data hidden. Therefore this dissertation focuses on developing interactive visualizations following a visual analytics [259] approach: a tight integration of visualization, interaction and algorithmic support that leverages the benefits of the individual parts, *i.e.*, “the whole is greater than the sum of its parts”.

1.2 Objective

The main research question addressed in this dissertation is as follows:

“ How to enable people to obtain insight in dynamic multivariate networks using a combination of automated and interactive visual methods? ”

Networks do not necessarily have to be both *dynamic* and *multivariate*, and we aim to provide tools and techniques to deal with both, hence we present individual and composite solutions. In combination, these solutions provide guidelines, recommendations, and techniques that enable network exploration and analysis while also being applicable in isolation.

To answer the research question an experimental approach is applied in this dissertation. For each topic of interest the solution is implemented in an application prototype, which is iteratively enhanced and improved.

Interaction is an underestimated element of information visualization that empowers the interplay between the visualization and the user, hence, this plays a key role in all our methods and prototypes. In addition, we do not restrict ourselves to just one application area, but we aim at generic applicability of the developed techniques to support a broad range of areas; many real-world phenomena can be approached as a dynamic multivariate network problem. Furthermore, during the design of all methods, techniques and tools we bear in mind *scalability*, *intuitiveness*, and *usability*: “*Think as a user, act as a user, be a user*” – Van Wijk [279].

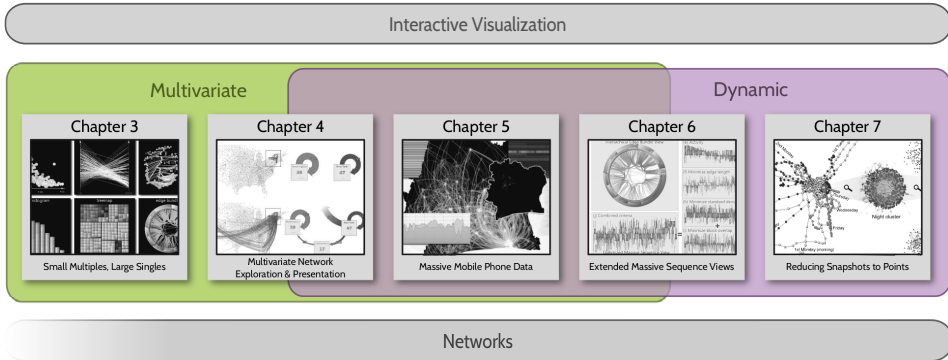


Figure 1.3: Overview of the chapter contributions to the research question. Each chapter addresses a combination of Multivariate, Dynamic, and Network aspects, while all contributions have Interactive Visualization as the key element. Chapter 3 focuses less on networks because the developed interaction technique can be applied to general multivariate data, including networks. Chapter 5 presents a practical implementation of visual analytics for massive mobile phone data and serves as an introduction and example of dynamic multivariate networks.

1.3 Outline & Contributions

The title of this dissertation contains the following keywords: *interaction*, *visualization*, *dynamic*, *multivariate*, and *network*. Figure 1.3 provides an overview how each keyword is addressed by the chapters. The remainder of this dissertation is organized as follows.

Chapter 2 provides an overview of interactive visualization techniques for network exploration and analysis. Next to static networks, techniques for multivariate, dynamic and the combinations thereof are discussed. Limitations of current methods and open problems are identified.

Chapters 3 to 7 contain the main contributions of this dissertation. Chapters 3 and 4 present new interaction techniques for multivariate data and multivariate network exploration and presentation. Chapter 5 presents a practical *big data* visual analytics solution for a real-world dynamic network dataset and serves as an illustration of the challenges involved with dynamic multivariate network visualization and exploration. Chapters 6 and 7 present novel visualization and interaction methods for the exploration of dynamic networks.

The key contributions of this dissertation are:

1. In Chapter 3 we present a novel visual exploration method based on small multiples and large singles for effective and efficient data analysis. Users are enabled to explore the state space by offering multiple alternatives from the current state and can then select the alternative of choice and continue the analysis. Furthermore, the intermediate steps in the exploration process are preserved and can be revisited and adapted using an intuitive navigation mechanism based on the well-known undo-redo stack and filmstrip metaphor. In

this chapter the effectiveness of the exploration method is tested using a formal user study comparing four different interaction methods.

2. Chapter 4 focuses on the non-expert user and proposes a novel solution for multivariate network exploration and analysis that tightly couples structural and multivariate analysis. In short, we go from Detail to Overview via Selections and Aggregations (DOSAs): users are enabled to gain insights through the creation of selections of interest, and producing high-level, infographic-style overviews simultaneously.
3. We present a system for the exploration and analysis of massive mobile phone data in Chapter 5. First we identify user tasks and develop a system following a visual analytics approach by tightly integrating visualization, interaction and algorithmic support. The system is evaluated by exploring a massive mobile phone dataset containing 2.5 billion calls and sms exchanges between around 5 million users located in Ivory Coast over a period of 5 months. This chapter serves as an introduction to the challenges involved in working with large scale dynamic (multivariate) networks.
4. In Chapter 6 we present a technique that extends the *Massive Sequence View* (MSV) for the analysis of temporal and structural aspects of dynamic networks. Using features in the data as well as Gestalt principles in the visualization such as closure, proximity, and similarity, we developed node reordering strategies for the MSV to make these features stand out that can optionally take the hierarchical node structure into account. This enables users to find temporal properties such as trends, counter trends, periodicity, temporal shifts, and anomalies in the network as well as structural properties such as communities and stars. We introduce the *circular* MSV that further reduces visual clutter. In addition, the (circular) MSV is extended to also convey time-series data associated with the nodes. This enables users to analyze complex correlations between edge occurrence and node attribute changes.
5. As a final contribution of this dissertation we propose a visual analytics approach for the exploration and analysis of dynamic networks in Chapter 7. We consider snapshots of the network as points in high-dimensional space and project these to two dimensions for visualization and interaction using two juxtaposed views: one for showing a snapshot and one for showing the evolution of the network. With this approach users are enabled to detect stable states, recurring states, outlier topologies, and gain knowledge about the transitions between states and the network evolution in general.

Finally, Chapter 8 concludes the dissertation by providing an overview and discussion of the results, presents a reflection by extracting general lessons learned, and closes with directions for future work.

1.4 Publications

All chapters in this dissertation are mostly self-contained and are based on the following research publications and patent application (ordered by chapter):

- “Small Multiples, Large Singles: A New Approach for Visual Data Exploration.” S. van den Elzen and J. J. van Wijk. *Comput. Graph. Forum*, 32(3pt2):191–200, 2013. (CHAPTER 3)
- “Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations.” S. van den Elzen and J. J. van Wijk. *IEEE Trans. Vis. Comput. Graphics*, 20(12):2310–2319, Dec 2014. (Best Paper Award IEEE InfoVis 2014). (CHAPTER 4)
- “Exploration and Analysis of Massive Mobile Phone Data: A Layered Visual Analytics approach.” S. van den Elzen, J. Blaas, D. Holten, J.-K. Buenen, J. J. van Wijk, R. Spousta, A. Miao, S. Sala, and S. Chan. *In Proc. 3rd Int. Conf. Analysis of Mobile Phone Datasets*, Cambridge, MA, May 2013. (Best Visualization Award D4D 2013). (CHAPTER 5)
- “Method and System for Data Visualization.” S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. *WO Patent App. PCT/EP2013/067,518*, 2012. (CHAPTER 6)
- “Reordering Massive Sequence Views: Enabling Temporal and Structural Analysis of Dynamic Networks.” S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. *In Proc. IEEE PacificVis*, pages 33–40, Feb 2013. (Best Paper Award IEEE PacificVis 2013). (CHAPTER 6)
- “Dynamic Network Visualization with Extended Massive Sequence Views.” S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. *IEEE Trans. Vis. Comput. Graphics*, 20(8):1087–1099, Aug 2014. (CHAPTER 6)
- “Reducing Snapshots to Points: A Visual Analytics Approach to Dynamic Network Exploration.” S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. *IEEE Trans. Vis. Comput. Graphics*, xx(xx):xx–xx, Dec 2015. to appear (Best Paper Award IEEE VAST 2015). (CHAPTER 7)

Another publication to which I contributed during my PhD but that is not included in this dissertation:

- “Data for Development Reloaded: Visual Matrix Techniques for the Exploration and Analysis of Massive Mobile Phone Data.” S. van den Elzen, M. van Dortmont, J. Blaas, D. Holten, W. van Hage, J.-K. Buenen, J. J. van Wijk, R. Spousta, S. Sala, S. Chan, and A. Kuzmickas. *In Proc. 4th Int. Conf. Analysis of Mobile Phone Datasets*, Cambridge, MA, April 2015. (Best Visualization Award D4D 2015).



Background

2

2.1 Introduction

IN the previous chapter the motivation and research question of this dissertation are presented. In this chapter we provide a background to place our work into context. First, we introduce the field of **visualization** and more specifically, the fields of *information visualization* and *visual analytics*. Furthermore, we discuss networks and the components involved. We provide an overview of network visualization. Moreover, we extend the current state-of-the-art taxonomy and refine it further by also taking multivariate data into account. Related work specific to the methods and techniques presented in the following chapters is discussed in the chapters themselves.

2.2 Visualization

Visualization concerns the transformation of data into images using graphical elements, which enables users to observe, explore, and interact with their data for *visual knowledge discovery*. Card *et al.* [59] define *visualization* as

“ the use of computer-supported, interactive, visual representations of data to amplify cognition. ”

Visualization is based on human visual perception. With visualization we exploit the pattern recognition capabilities of the human visual system. Our eyes act as a very high-bandwidth channel to the brain where a large portion is dedicated to visual processing. The visual information processing occurs for a large portion in parallel at the preconscious level [195, 292]. With rapid parallel processing of the environment by extraction of features, orientation, color, texture, and movement patterns, we are able to effortlessly detect and recognize objects within milliseconds. This mechanism, the detection of objects and relationships without cognitive inference, is known as *pre-attentiveness* [292].

If we know exactly what we are looking for in our data, and we know all the questions to be asked in advance, then we do not need visualization. We can just use some automatic method, *e.g.*, a complex algorithm or a simple computation to provide the answer. However, most of the time we do not know what we are looking for in our data and we do not know all the questions to be answered. We need a means to *explore* the data and form hypotheses to be tested. For this purpose, visualization is a powerful technique; we enable exploration by showing the data and providing the user with controls for navigation.

The visualization process, as described by Card, MacKinlay, and Shneiderman [59] (see Figure 2.1), consists of a number of components that support visual sense making. First, the collected raw data is *transformed* into derived (*reduced, aggregated, filtered*) data that is easier to manipulate and comprehend. Next, a *visual mapping* is defined that creates *visual structures* for the data. After mappings are defined, a *view transformation*

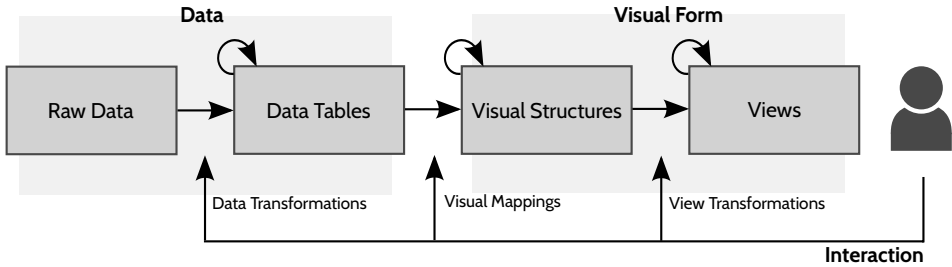


Figure 2.1: Reference model for visualization by Card et al. [59]. Visualization can be described as the mapping of raw data to visual form using graphical elements. User interaction enables navigation and visual knowledge discovery.

provides a perspective on the data that is presented to the user with an image. Then, the perceptual and cognitive capabilities of the user enable them to interpret the image and build knowledge, gather insights, and form hypotheses about the data. *Interaction* plays a key role in the exploration of data. Users are enabled to iteratively interact with each step of the visualization process to *navigate* through the data to understand the complex patterns involved and to focus on what is most relevant.

2.3 Information Visualization

Multiple definitions for information visualization have been proposed [58, 59, 62, 80, 168, 209, 249, 263]. In summary, *information visualization*, as opposed to *scientific visualization*, is the art of creating interactive visualizations for non-physical, abstract data. Abstract data has no inherent spatial mapping, typical examples are multivariate data (tables), networks (graphs), hierarchies (trees), and time-series data.

At the core, information visualization consists of two components: *representation* and *interaction*. Many design models and interaction techniques have been proposed for the creation of effective interactive visualizations. Below we briefly discuss a selection of the most important methods and techniques:

- **Information Seeking Mantra** [243] – *overview first, zoom and filter, details-on-demand*. First, users should be provided with an overview of the data. This overview gives an impression of global patterns and outliers. Next, users should be enabled to zoom in and filter the data to see more detailed patterns. Finally, exact values of (individual) items should be shown on demand. All components need to be supported with simple intuitive interaction methods and transitions between them should be smooth. A variation on the information seeking mantra, applied to the domain of networks, is discussed by Van Ham and Perer [278]: *search, show context, expand on demand*. They advocate that a network overview is not always the best start for exploration, hence first users *search* for a node of interest. They should next be enabled to analyse this node in *context* and continue the exploration by *expanding on demand*. Both variations are valid and depending on context different techniques are preferred.

- **Direct Manipulation** [242] – direct manipulation states that users should be enabled to directly manipulate (select, highlight, move) visual items in a scene. Direct manipulation is both intuitive and encourages data exploration with interaction. Direct manipulation is supported by good design of visual *affordances* [200]. Affordances are the perceived properties of a (visual) item that determines how it can be used. This can be supported by the use of familiar metaphors in the visualization design.
- **Dynamic Querying** [242] – instead of traditional querying from a database by using a special purpose query language, users should be enabled to query visually, using direct manipulation on graphical interface elements such as sliders, scented widgets [298], and selection boxes, upon which results are shown instantaneously.
- **Multiple Coordinated Views** [289] – users can be provided with multiple views that each have a different viewpoint on the data, to enable users to observe complex relations that would otherwise be hidden. Combined with *linking and brushing* the finding of relationships is further improved and simplified.
- **Linking and Brushing** [47, 165] – corresponding items that are highlighted or selected in one view, are also highlighted or selected in all other views. This overcomes the shortcomings of a single visualization and provides more information than exploring the visualizations in isolation.
- **Focus + Context** [59] – selected items of interest are presented in detail with an according higher-level overview that provides insight and places the items in context for better relational understanding.
- **Overview + Detail** [66] – related to *multiple coordinated views* and *focus and context*, at least two views are simultaneously presented to users; one with a detailed view of items of interest and the other visualizing the entire visualization space showing less detail.
- **Zoom and Pan** – users should be enabled to use zoom and pan techniques. With zooming and panning operations, the visible viewport of the visualized data is geometrically transformed (zoom – scaled or pan – repositioned) while the data and visualization remain unchanged. Zooming and panning overcomes the limitation of visualization resolution and color depth.
- **Semantic Zoom** [205] – semantic zoom, in addition to geometric zoom, states that upon zooming-in, gradually more detailed information of the items involved is shown in the visualization.

All these visualization design and interaction techniques are utilized in the developed prototypes presented in the following chapters. Chapter 8 concludes by introducing several other visualization design and interaction techniques that are derived from our own experience during the development of the tools and techniques presented in this dissertation.

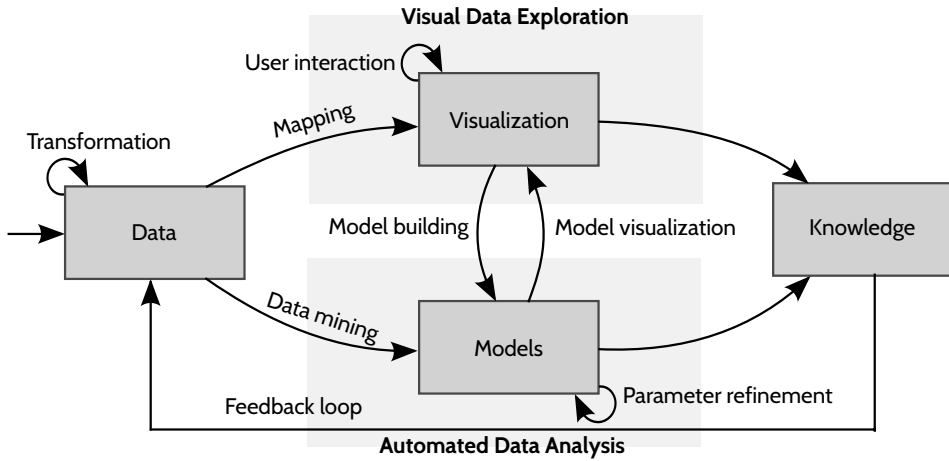


Figure 2.2: The visual analytics process by Keim et al. [166]. Visual data exploration and automated data analysis are combined through interaction with the data, visualization, and models, to obtain knowledge.

2.4 Visual Analytics

The field of visual analytics is an extension of information visualization. Where information visualization is mainly focusing on *representation* and *interaction*, visual analytics is a multi-disciplinary field of research that supports users in the analytical sense making process. Thomas and Cook [259] defined the field as, “*visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces*”. A more elaborate definition is given by Keim et al. [166]:

“ *visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex datasets.* ”

Visual analytics combines methods from the fields of information visualization, statistics, data mining, machine learning, cognitive psychology, perception, knowledge and data management, and human factors. Visual analytics focuses on the integration of human decision making and automated data analysis methods to support a collaborative decision-making process. The main idea is to combine the strengths of human sense-making and automatic data analysis. Using visualization and interaction the semi-automatic analytical process is steered with a *human in the loop* approach.

The visual analytics process, shown in Figure 2.2, is more involved compared to the information visualization process (see Figure 2.1). Next to data filtering and mapping, also automated data analysis methods from data mining are utilized to build, refine, and visualize models. Combined, the visual data exploration and automated data analysis

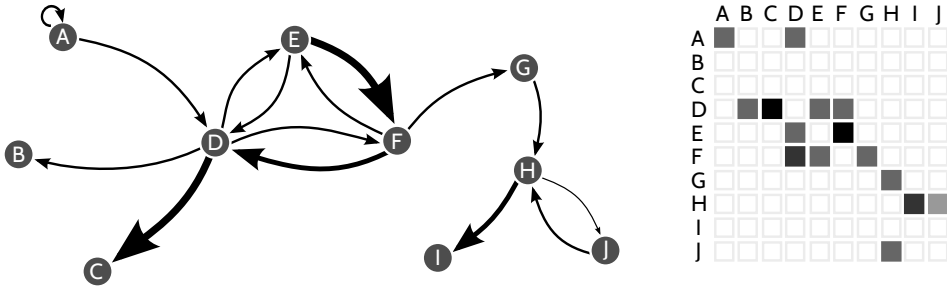


Figure 2.3: Node-link diagram (left) and corresponding visual adjacency matrix (right). Link weights can be encoded with different visual variables such as width in the node-link diagram and grayscale color in the visual adjacency matrix.

enable iterative knowledge building. The information seeking mantra is adapted accordingly by Keim *et al.* [167]: “analyze first; show the important; zoom, filter and analyze further; and details-on-demand”. This extension indicates that, in contrast to the information seeking mantra, it is not sufficient to just retrieve and visualize data; rather, it is necessary to first analyse the data according to items of interest, showing only the most relevant aspects. Next, the data is interactively analyzed further showing details on demand.

2.5 Network Visualization

Static networks are typically visualized using a node-link diagram (see Figure 2.3). In a node-link diagram a node is visualized using a dot, point, or other representative glyph such as a circle or rectangle. Links of the network are visualized by drawing straight or curved lines between nodes that have a relation. Often, arrow heads are used to show the directionality of the relation. The greatest advantage of a node-link diagram is its intuitive representation, which is easy to understand by non-expert users.

The geometrical position of the nodes and links in the network is defined as the *layout* or *embedding*. Computing a two dimensional embedding of the network is an important area of research in the field of *graph drawing* [24]. Many algorithms have been developed, an important class are the *force-directed layout* algorithms [173]. Some examples of force-directed layout algorithms are Fruchterman-Reingold [109], Kamada-Kawai [164], and LinLog [199]. These methods are typically based on a simulation of a physics model consisting of attracting and repelling forces; nodes repel each other and are simultaneously attracted if a link exists between them, acting as a spring. The position of the nodes is iteratively updated based on the forces applied to them and eventually converges to a (near) stable node configuration.

The performance of standard force-directed methods does not scale well to large networks due to the significant amount of conflicting forces. As a consequence, the algorithm needs a large number of iterations to converge to a stable state. A solution to this are *multi-scale methods* [67, 111, 123], that start by laying out a small coarse

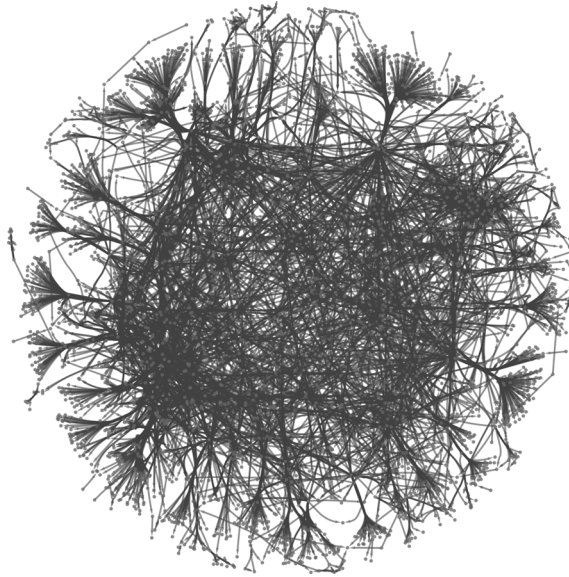


Figure 2.4: Force-directed layout applied to a moderate size network that consists of 2924 nodes and 6827 edges. Nodes are rendered using dots and links are visualized using arrows. In addition, the links are routed using a force-directed edge bundling algorithm [140] to improve readability. The resulting visualization resembles a hairball. Image produced using Cytoscape [234].

representation of the network and gradually layout finer, more precise, representations of the network until the entire graph is processed. These multi-scale approaches scale better compared to the force-directed algorithms while providing similar results.

In general, network layouts are computed such that *readability* and *aesthetic criteria* [24], are maximized. Some examples of criteria are: 1) all edges should be of similar length, 2) the number of crossing edges should be minimized, 3) nodes should not overlap with each other, 4) high-degree nodes should have a central position, 5) symmetry should be maximized, and 6) communities should be clearly visible. These criteria are often conflicting and this results in NP-hard optimization problems. Therefore, many algorithms are based on heuristics. We further elaborate on this in Chapter 6, where heuristic methods are presented that improve the visual recognition of temporal and structural patterns in a dynamic network.

The most prominent problem with node-link diagrams other than *computational scalability* is *visual scalability*. When the number of nodes and edges is large, in the order of thousands or bigger, finding a suitable configuration is difficult due to the *small-world* property [294] of most real-world networks. In a small-world network the average path length is low in comparison with an equivalent size random network. Also, in a small-world network the connectivity among nodes is high. Because of these properties, the resulting network visualization typically resembles a *hairball* from which no insights can be extracted due to visual clutter and heavy overdraw (see Figure 2.4).

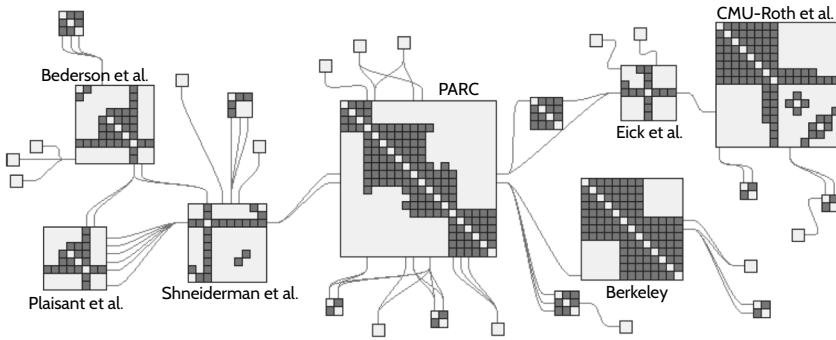


Figure 2.5: NodeTrix [133]: a hybrid node-link and matrix network representation.

A more visually scalable solution is the *visual adjacency matrix*, e.g., the Matrix Zoom approach by Abello and Van Ham [10], or the interactive large-scale graph visualization by Elmqvist *et al.* [86]. With this technique the network is visualized using a direct visual representation of the adjacency matrix. Nodes in the network are mapped to rows and columns in the matrix M (see Figure 2.3). A cell $M_{i,j}$ in the (visual) matrix depicts the existence of edge $i \rightarrow j$ in the network using a visual variable such as color. Next to visual scalability, the visual adjacency matrix does not suffer from crossing edges and maximizes edge visibility. The matrix representation outperforms node-link diagrams on most user tasks [115]. However, a drawback of the visual adjacency matrix is the difficult identification and challenging interpretation of structural properties of the network (e.g., communities, paths, and motifs [238]). The ability to identify and recognize network topology depends on a non-trivial ordering of the nodes [246]. A combination of node-link diagrams and visual adjacency matrices, trying to leverage the advantages of both with a dual representation system, is proposed by Henry and Fekete in the MatrixExplorer [132]. A hybrid node-link matrix representation is explored in the NodeTrix technique [133], see Figure 2.5. For *sparse networks*, where the number of nodes is relatively large compared to the number of edges, the matrix representation leaves a lot of space unused, this effect is already visible in the example of Figure 2.3. The fraction of unused space generally increases as the number of nodes grows. Solutions to this are folding [88] and compression [79] of the matrix.

A variation on the visual adjacency matrix is the visual adjacency list [135]. With this technique nodes are layed out in a vertical column. Next, for each node a horizontal row is used to depict incoming and outgoing edges. Edges that are incoming are positioned before this column and outgoing edges are shown on the right side of the node. By using the horizontal space for time, also dynamic networks can be visualized.

Another visual scalable solution is *hierarchical edge bundling* by Holten [137]. Nodes are positioned on a circle and the network node hierarchy is exploited to bundle edges. This prevents clutter and shows both global and local communication patterns. This technique needs additional static hierarchical node data. A variation that does not need the hierarchical information, based on a force-directed model, is presented by Holten and Van Wijk [140]. Multivariate networks with an additional static node hierarchy are *compound networks*.

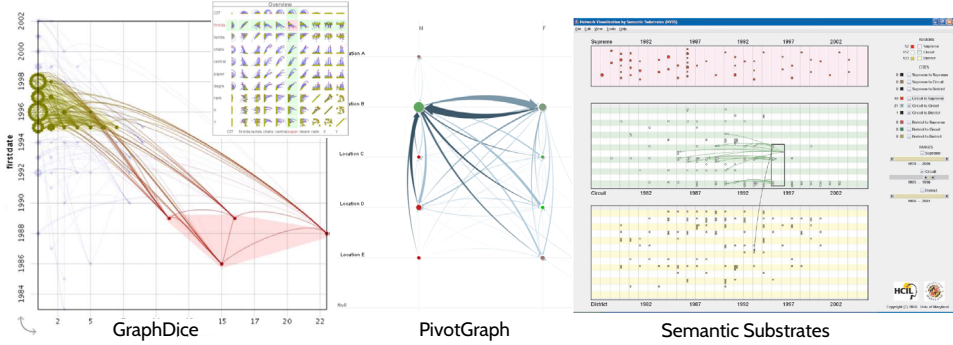


Figure 2.6: Multivariate network visualization and exploration enabled with GraphDice [31], PivotGraphs [293], and Semantic Substrates [244].

2.5.1 Multivariate Network Visualization

Multivariate data associated with the nodes and links of the network is commonly depicted using *visual variables* in a standard network visualization. Examples of visual variables are color, size, shape, thickness, and texture of both the nodes and links, e.g., [38, 101, 204, 233, 237, 261]. Next to visual variables, glyph representations of both nodes and links are used to convey multivariate data [291].

In case the network is visualized using a node-link diagram, the multivariate data can be used to compute an *attribute-based embedding*, e.g., [18, 97, 163, 302]. With an attribute-based layout the position of the nodes provides insight in the associated multivariate data. A disadvantage with this method is the reduced readability of structural properties of the network. Taking this concept further, multivariate data can also be used to directly position nodes in a scatterplot, and drawing the links of the network on top of the nodes in the scatterplot. An example of this is the GraphDice system by Bezerianos *et al.* [31]. Aggregated scatterplots enable a higher-level exploration of the multivariate data as in PivotGraphs [293]. With this technique, network topology is not preserved due to aggregation. Instead of scatterplots, groups can be defined for the multivariate data values, as in the Semantic Substrates techniques by Shneiderman and Aris [244]. The groups are non-overlapping and contain nodes according to the data values. The layout of the nodes within the groups is flexible and can be set directly to attribute values or computed with a force-directed algorithm. Again, edges are superimposed within and between the regions. This reveals both structural and multivariate data patterns. Representative visualizations by the GraphDice, PivotGraph, and Semantic Substrates techniques are shown in Figure 2.6. Stolper *et al.* [253] define graph-level operations to simplify the challenge of building such multi-technique network visualization applications.

Finally, a general technique is to use familiar charts such as a scatterplot, bar-chart, histogram, or a treemap to depict the multivariate data. This can be used in a multiple coordinate view setting with (at least) two juxtaposed views; one view providing insight in the network structure and a linked view showing the associated multivariate data.

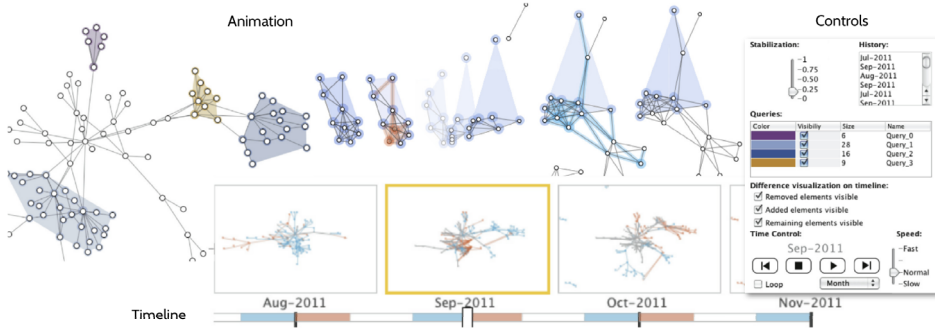


Figure 2.7: GraphDiaries by Bach et al. [20]: a system implementing animated staged transitions of node-link diagrams with timeline and controls, for the exploration of dynamic networks.

2.5.2 Dynamic Network Visualization

For dynamic network visualization typically the evolving structure of the network is of interest: understanding how nodes and edges are added and removed to get insight in the underlying dynamics. In addition, the multivariate data associated with the nodes and links can also evolve over time, so called *time-series* data. In the literature different terms exist to describe the same concept: *time-varying network*, *time-stamped network*, *longitudinal network*, *evolving network*, and *temporal network*. Furthermore, the term *network* is often interchanged with the more technical term *graph*. In this dissertation we will refer to the concept described above as *dynamic network*. There are three main approaches for the exploration and visualization of dynamic networks: *animation*, e.g., [106, 177, 214], *small multiples* [30, 265], e.g., [93, 118, 216] and *integrated* approaches, e.g., [54, 135, 218].

Animation For each timestep of the dynamic network a layout is computed that is used for animation. The series of layouts are next sequentially shown and played like a movie. In this animation, nodes and edges appear, disappear, and change position. In addition, visual variables of the nodes and edges may vary. To enhance exploration, users can generally pause, replay, and skip to parts of the animation using a timeline control [20, 220], see Figure 2.7 for an example by Bach *et al.*

If a node-link diagram is used for the animation, e.g., [37, 76, 90, 108, 122, 201], it is generally deemed important to keep the variation of the layouts over time as small as possible. As an example, Federico *et al.* [95] show how changes can be minimized in a visual analytics approach for dynamic social networks. A stable network layout during the animation keeps the cognitive load of users at a minimum and is known in the literature as *preservation of the mental map*. Preservation of the mental map can be achieved by computing constrained node layouts for new timesteps. A constraint is for example to anchor nodes to the position in the previous timestep [42, 107, 108, 185]. Despite the research in this area, the claimed positive effect of preserving the mental map is not proven. In user studies, no positive effect is confirmed [19, 210, 221], rather a good individual layout seems to provide better results.

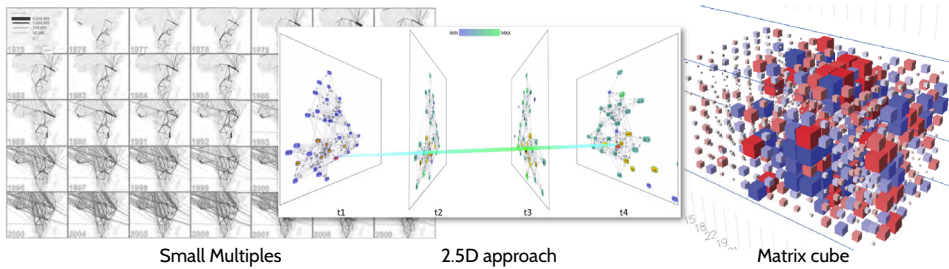


Figure 2.8: (left) *Small multiples* of a dynamic network with a grid layout as used in a study by Boyandin et al. [39]. (middle) *2.5D approach* by Federico et al. [96]: stacking timesteps with node-link diagrams. (right) *Stacking visual adjacency matrices with the matrix cube technique* by Bach et al. [22].

Animation is not only combined with node-link diagrams but also with visual adjacency matrices in AniMatrix by Rufiange and Melançon [220].

Animation has drawbacks: users need to focus on many moving or changing items simultaneously and need to keep track of (multiple) changes over longer time periods. Also, *change blindness* plays a role in dynamic network animation [202]. Change blindness is a phenomenon in human visual perception that occurs when people do not notice changes in a scene when they are focusing on one item or area [248]. This might also occur when the change is abrupt, hence hard to recognize. In addition, it occurs if the change is smooth but the context of the visualization is lost or is (shortly) blanked out. The small multiples technique, introduced in the next section, overcomes some of these perceptual and cognitive issues, but also introduces problems of its own.

Small Multiples Similar to the animation technique, the small multiples technique is independent of the visualization method that is used. In the small multiple technique the different timesteps are shown as juxtaposed visualizations using a filmstrip or grid layout. For each timestep a node-link layout or other visual representation of the network is computed. An example small multiple setting is shown in Figure 2.8(left).

For small multiples we need to decide on the number of multiples to display in the visualization. If many multiples are used, the visualization space for each individual timestep is limited. This reduces the readability and the multiples might be far apart from each other, which hampers the ability to relate and compare them for the discovery of patterns. A solution to this when small multiples are used interactively, is the use of large singles for detailed inspection (see Chapter 3).

Instead of positioning the small multiples using a grid layout or arranging them linearly, several techniques stack the multiples on top of each other using two-and-a-half or three dimensional drawing techniques. They can be stacked vertically, as a stack of sheets, or horizontally, resembling standing books. For better association between the layers, corresponding nodes of sequential timesteps can be connected with lines, e.g., [22, 83, 94, 95, 96, 120]. The metaphor used here is that of a flipbook and the visualization is intuitive for non-expert users. The major concern however, is scalability in terms of timesteps.

While these techniques work for about five timesteps, visualizing more timesteps needs sophisticated interaction techniques. Also, the visual adjacency matrix is extended to a stacked variant in the matrix cube approach by Bach *et al.* [22]. Figure 2.8 shows a 2.5D approach by Federico *et al.* [96] and the matrix cube technique by Bach *et al.* [22].

Integrated approaches The animation and small multiple techniques both make use of the network at different timesteps. Another approach is to provide a static overview of the entire time span of the network in one visualization. In such an overview individual nodes and edges can be shown or they can be aggregated using time-intervals or by constructing a *super-graph*. A super-graph is an overview representation of the dynamic network constructed by aggregating all timesteps. Each edge is assigned a weight according to the number of appearances in each of the individual timesteps. This super-graph can be used to guide transitions in the animation of node-link diagrams, *e.g.*, [182].

A visual matrix can be used to convey the dynamics of the network with *intra-cell* [43, 53, 124, 305] or *layered* [22, 283] techniques. With the intra-cell technique each cell of the visual adjacency matrix contains a glyph that conveys the evolution of that link. For the layered technique one dimension of the matrix is used to represent time.

Some visualization examples that provide a static overview of the entire timespan of the network are the *Massive Sequence View* (see Chapter 6), *Timeline Trees* [48], *TimeSpiderTrees* [51], *TimeRadarTrees* [50], *Layered TimeRadarTrees* [52], *Parallel Edge Splatting* [54], *TimeEdgeTrees* [55], *TimeArcTrees* [119], *Alluvial diagrams* [218], *Radial Layered Matrix Visualization* [283], and *Visual Adjacency Lists* [135]. The advantage is that a complete overview of the network is presented and global patterns can directly be identified. Disadvantages are that these specialized visualizations are often difficult to interpret, especially for casual users; they are difficult to reproduce; and generally pose restrictions on the number of timesteps or the network type, *e.g.*, acyclic, directional, compound *et cetera*.

2.6 Dynamic Multivariate Network Model

Dynamic multivariate networks have many different aspects, and their definition is different depending on the context. Below we describe some additional properties involved in dynamic network data that are not discussed in previous sections:

- *online* versus *offline*: in an online setting the nodes and edges of the dynamic network are not known beforehand. A consequence is that computing a node-link layout that preserves the mental map based on the super-graph is not possible. Furthermore, online is often linked to a *streaming data* setting in which nodes and links are added and removed in real-time. For offline networks, the entire network evolution is known at the moment of analysis. Offline approaches therefore allow for better optimization of the layout, visualization, and enable better preservation of the mental map.
- *continuous* versus *discrete*: for continuous dynamic networks the nodes and edges have a real-valued time-stamp, *i.e.*, time is modeled continuously. The edge

occurrences can therefore be visualized on an infinitely zoomable continuous timeline. In contrast, time can also be modeled as discrete time steps. Depending on the model used, or inferred by the data, according visualization designs should be considered. Transformation from continuous to discrete is possible by aggregation at the cost of losing information. Transformation from discrete to continuous does not lose information but may not be appropriate.

- *instant versus duration*: edge occurrence can be modeled as an instant event, *i.e.*, the occurrence has no duration, or as having a start- and end-time. Obviously, this has implications for the visual design. Modeling edge occurrence as instant events can also be used as a simplification method.

In addition to these properties, the multivariate data associated with the nodes and edges can be static or time-varying. Static information on the nodes and edges concerns properties of the objects such as year of birth and gender in case of persons. This information can also relate to a hierarchy, such as current job-title within the company. Time-varying data generally denotes measurements in the form of time-series data.

In this dissertation we touch upon most of the variants discussed above and provide interaction and visualization techniques for these. For example in Chapter 6 we discuss dynamic networks with hierarchical static node information, node time-series data, and dynamic edge data. Chapter 4 discusses static multivariate data on both the nodes and edges. Chapter 5 also involves hierarchical static node information and dynamic edge information. Due to the variation of networks discussed we do not provide an overall dynamic multivariate network model, but define the models that we use separately in the according chapters. As a constant factor, we assume in all chapters that we are working in an *offline* setting; the set of nodes of the network does not change, edges change over time, and we have complete information for the entire timespan.

2.7 Taxonomy

To position the work discussed in Sections 2.5.1 and 2.5.2 and to provide a context for the work presented in the next chapters, we provide an overview of all work in this area, starting from the hierarchical taxonomy of dynamic multivariate networks as presented by Beck *et al.* [27] in a recent state-of-the-art survey paper. New work, not covered by the survey, is added to the taxonomy. We extend the taxonomy by introducing a category branch *projection* on the highest level besides *animation* and *timeline*. In Chapter 7 we present a method that reduces snapshots of the dynamic network to points and *project* these to two dimensions. In addition to this extended taxonomy we provide an overview to show the involvement of multivariate data by adding another orthogonal taxonomy.

At the highest level the dynamic network visualization taxonomy is split into **animation** and **timeline**. We extend upon this and add to this the branch *projection*.

At the next level **animation** is split into **general purpose layout** and **special purpose layout**. The **general purpose layout** is again further divided into *online*, *offline*, and *transition*. The **special purpose layout** is divided into the categories *compound* and *other*.

The **timeline** category is split into **node-link** and **matrix**. Also here we extend the taxonomy with a branch *adjacency list* to position new work by Hlawatsch *et al.* [135]. At the leaf level, by dividing the category **node-link**, we find the categories *juxtaposed*, *superimposed*, and *integrated*. The **matrix** branch is divided into *intra-cell* and *layered*.

2.7.1 Explicit Multivariate Data

Since the original taxonomy of dynamic network visualization includes multivariate data implicitly, no real distinction is made between purely dynamic networks and dynamic multivariate networks. To better show this distinction and enable the identification of gaps in the literature we make this explicit by adding a second taxonomy and reconsider all literature. The second taxonomy provides a more precise classification of current state-of-the-art in dynamic multivariate network visualization.

At the highest level we distinguish between **multivariate data** and **non-multivariate data**. Next, for **multivariate data** we further divide into **static** and **dynamic**. At the lowest level, both categories are divided into *node* and *edge* data association. Typical static node data is hierarchical information that does not change over time. Static edge data generally states the type of relation between two nodes, e.g., “*is boss of*”, “*is parent of*”, “*works at*”, *et cetera*. Obviously, in the real world such data can be dynamic, and considering these as static is just used to limit the scope and complexity for given cases. Dynamic node data is time-series data associated with the nodes. Typically this involves measurements, for example the balance of a bank account (node) that changes with every transaction (edge). Dynamic edge data is commonly known as a *weighted* network. Each edge occurrence in time has an associated value such as the amount of a financial transaction.

Figure 2.9 shows our extended taxonomy based on the original by Beck *et al.* [27] and the further subdivision in explicit multivariate data associated with the nodes and edges of the dynamic network visualization. Also, relevant publications are indicated. Outlined cells contain work presented in this dissertation with the according chapters denoted inside the cell. The work introduced in Chapter 3 on Small Multiples and Large Singles is not directly targeted towards networks, but the techniques are generic and can be applied to dynamic multivariate networks, hence we included this in the taxonomy.

From the diagram we see that most dynamic multivariate network visualization methods are based on a timeline approach with node-link visualization combined with a juxtaposed or integrated technique. Also, animation techniques with a special purpose layout to visualize compound networks, *i.e.*, dynamic networks with a static hierarchy on the nodes, are prominent. Predominantly, when matrices are used for the visualization of dynamic networks only static multivariate node and edge data is taken into account which is frequently conveyed with intra-cell visual variables.

We do not believe in animation as a suitable technique for the visualization, exploration, and analysis of large dynamic multivariate networks for reasons given in Section 2.5.2. For our own work (see Figure 2.9, outlined blocks) we mainly focused on timeline based approaches with node-link diagrams as visualization technique. For the node-link approach we developed methods for all three visualization techniques, juxtaposed (Chapter 3 and 5), superimposed (Chapter 4), and integrated (Chapter 6). Furthermore,

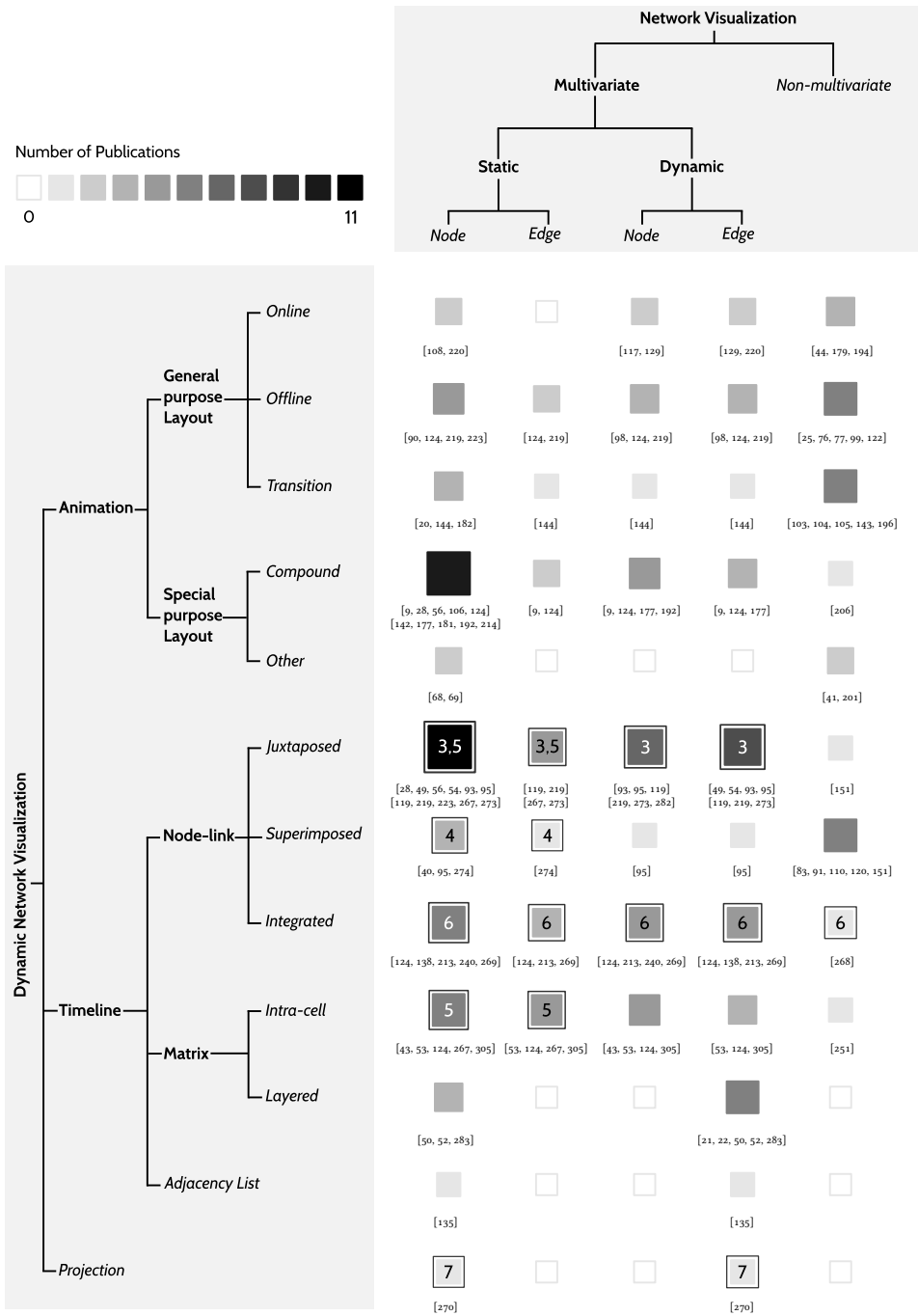


Figure 2.9: Dynamic multivariate network visualization taxonomy. A cell shows the intersection between dynamic and multivariate network visualization, labels refer to chapter numbers of this dissertation. The taxonomy of Beck et al. [27] (left) is extended and multivariate data association is made explicit (top).

we implemented a temporal matrix approach with timeline (Chapter 5). We mainly focused on node-link based visualization techniques because we believe they are most accessible to a broad audience due to the intuitive representation. The disadvantages of large node-link diagrams, *i.e.*, overdraw, clutter, poor embedding, are overcome with carefully designed interaction techniques.

Next to the timeline node-link and matrix approach we extend the boundaries of known techniques by presenting a projection based method in Chapter 7 that creates possibilities for further exploration by enlarging the design space.



Small Multiples, Large Singles

3

This chapter is based on [273]:

“Small Multiples, Large Singles: A New Approach for Visual Data Exploration.” S. van den Elzen and J. J. van Wijk. *Comput. Graph. Forum*, 32(3pt2):191–200, 2013.

3.1 A New Approach for Visual Data Exploration



Figure 3.1: *Novel visual data exploration method using small multiples. Users alternate between Small Multiples and Large Singles for comparison and guidance during exploration using a filmstrip metaphor.*

WE present a novel visual exploration method based on small multiples and large singles for effective and efficient data analysis. Users are enabled to explore the state space by offering multiple alternatives from the current state. Users can then select the alternative of choice and continue the analysis. Furthermore, the intermediate steps in the exploration process are preserved and can be revisited and adapted using an intuitive navigation mechanism based on the well-known undo-redo stack and filmstrip metaphor. As proof of concept the exploration method is implemented in a prototype. The effectiveness of the exploration method is tested using a formal user study comparing four different interaction methods. By using small multiples as data exploration method users need fewer steps in answering questions and also explore a significantly larger part of the state space in the same amount of time, providing them with a broader perspective on the data, hence lowering the chance of missing important features. Also, users prefer visual exploration with small multiples over non-small multiple variants.

3.2 Introduction

Visualization plays an important role in the analysis of multivariate data. Many use visualization to obtain insight, to select details, and to produce presentations of their data. Here we focus on non-expert users, who want to explore data incidentally, who are not used to complex multiview displays, and require as simple as possible means to explore their data. The creation of different views on the data is a crucial step in this. One way to do this is to iteratively select a parameter and to choose a new value for this parameter. In short, visual data exploration is a sequence of steps where in each step:

1. a parameter is selected;
2. a value is chosen.

There are, however, several problems with this iterative approach. First, it can be time-consuming and error-prone. Users often do not know what they are looking for in their datasets. In traditional data exploration, often parameter selection and choosing a

value are offered in a single operation, which leads to an iterative trial-and-error process. Often it is unclear as to what value a parameter needs to be changed to, to gain insight in the data and enabling the discovery of features. Typically, all (parameter, value) pairs are tried one by one, or different parameter value pairs are changed more or less in a random fashion. This is not only inefficient, it can also lead to missing interesting features because a parameter value is not inspected. Furthermore, comparison of results for different parameter values is not supported, and the visualization for the old parameter value is typically lost on change. Also, as no history is kept except for undo-redo operations, it is difficult to link findings discovered early in the exploration process to features found later on. These shortcomings can be overcome by introducing a visual exploration method in which users are offered:

- easy comparison of the effect of different parameters;
- guidance on what value to choose; and
- a history trail of the exploration path.

Our approach is based on the use of small multiples as the central element for exploration. Showing many small visualizations simultaneously facilitates comparison, presentation, and storytelling. Viewers can compare the separate images, and look for patterns, trends, and outliers. Small multiples are currently mainly used as a static visualization technique. They tend to get very small, which is not an issue for presentation on posters, but is cumbersome for typical user device displays, and applications that do use small multiples suffer from this. Small multiples are typically shown here as end results, and play no explicit role in the exploration process. One exception is their use as a preview for alternatives, e.g., changing chart style in Microsoft Excel. This shows the viability of the use of small multiples for interactive exploration, but in these cases their application is often disruptive. They are shown as a pop-up, hiding the original visualization, and not integrated in the base visualization itself.

The challenge addressed in this chapter is how to effectively integrate small multiples in interactive visual data exploration, such that they are not only helpful for visualization, but also provide guidance and support for the exploration process itself. We implemented different exploration interaction methods in a prototype and tested different designs using a user study. This led to a new visual data exploration method using small multiples with key aspects:

- alternating Small Multiples and Large Singles;
- simultaneous display of current and new state(s); and
- use of a filmstrip metaphor.

The chapter is organized as follows; first, related work is discussed in Section 3.3. Next, the interaction design and navigational techniques offered are presented in Section 3.4. In Section 3.5 we discuss the effectiveness and usability of the small multiples and large singles method based on a user study. Finally, limitations, conclusions and directions for future work are provided in Sections 3.6 and 3.7 respectively.

3.3 Related Work

The term *small multiples* is introduced by Tufte [265] who described them starting from resemblance of movie frames: a series of graphics, showing the same combination of variables, showing changes in another variable. However, they were earlier proposed under the different term *Trellis* displays [29] due to their resemblance of a garden trellis fence. Even earlier they were called *collections* by Bertin [30]. Up to now, small multiples is mainly used as a static visualization technique, but is rarely used for interaction and seamless integration in the visual data exploration process.

Chi *et al.* [64] introduce a spreadsheet approach to information visualization where the cells contain visualizations resembling small multiples. The spreadsheet technique is formalized [65] and applied to web analytics [63]. Users are enabled to set different variables to the horizontal and vertical axis of the spreadsheet table. This is extended by Jankun-Kelly and Ma to include encapsulation of the history process, which can be replayed via animation [155, 156, 157]. Marks *et al.* [190] propose *design galleries* for visual input parameter exploration using small multiples in a computer graphics setting. Small multiples presented in table form with emphasis on sorting within each multiple is explored by Rao and Card [212]. In contrast to our method, these exploration methods are solely aimed at the visual parameter space.

Small multiples are applied to data analysis in different application domains, *e.g.*, the biomedical domain implemented by Sarni *et al.* [224], the geographic domain implemented by Guo *et al.* [121] and Willems *et al.* [297]. A small multiple interface is used to explore large cancer simulation parameter spaces by Lunzer *et al.* [184]. In these interactive systems, however, small multiples are used as embedded visualization method and are not used as exploration method, as we aim for.

In Polaris [255] users have the ability to rapidly change the table configuration, type of graphic [187], and visual encodings used to visualize a dataset. This is extended to create multiscale visualizations based on data cube projections [254, 256]. MacEachren [186] investigates high dimensional data space exploration using small multiples. However, they mainly focus on finding interesting dimensions in the dataset to create small multiples for. Bavoil *et al.* focus on the construction and optimization of visualization pipelines to generate small multiples for analysis using the Vistrails system [26, 57, 225, 247]. Heer and Shneiderman discuss the advantages of small multiples as coordinated multiple views [131]. Boyandin *et al.* compare small multiples against animation in a qualitative study with focus on exploration of temporal changes in flow maps [39].

Small multiples are a classic visualization method, used in many systems. We see however that they tend to be used as end-results. We argue that they can be used even more effectively if they are used as a visualization and as an exploration method simultaneously, and we did not find work where this simple, yet powerful, idea has been proposed before.

3.4 Small Multiples, Large Singles

A typical visual data exploration system aiming at occasional users employs a Large Single visualization. Alongside this visualization there is typically a menu present to manipulate the displayed visualization by changing visualization parameters, such as the type of visualization, and the according parameters, for example what to display on the x -axis. Furthermore, there are standard options to select and filter the data to be shown in the visualization. By combining visual parameters with data filtering, a large number of possible visualizations on the data are available to users. A specific combination of visual parameter settings and data filter options can be thought of as a *state* in the exploration process.

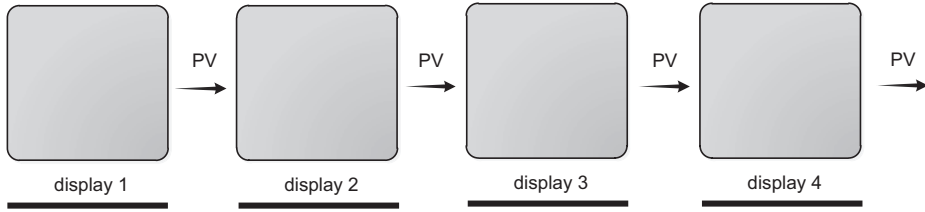
In terms of the standard visualization pipeline, data is transformed into visualizations in a sequence of steps (filtering, mapping, viewing), where each step is controlled via a number of parameters, which can be changed interactively by users. If we extend this to visual analytics, also one or more steps where data is analyzed (clustered, classified, *et cetera*) can be included, which introduces yet more parameters. Some parameters depend on other parameters, such as visualization type specific parameters, others are independent such as cluster-method used.

Users typically navigate in the exploration process by changing parameters, schematically shown in Figure 3.2(a); in each step a parameter is chosen and a new parameter value is assigned. If users are experienced and know the aim, they can do this in a straightforward way, in many cases however a trial-and-error process is needed to obtain insights, as no guidance is provided what parameter to change next or what parameter value needs to be chosen to discover features present in the data, such as anomalies, clusters, correlations and trends.

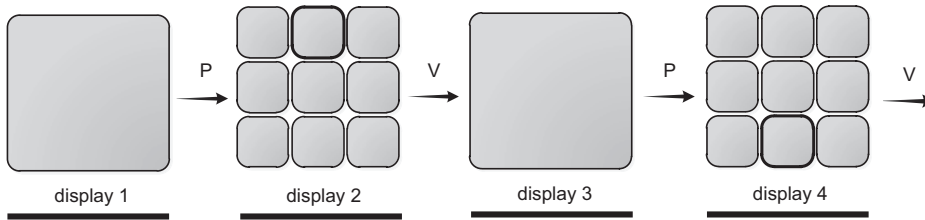
3.4.1 Approach

The use of small multiples enables users to compare all different alternatives and guides them to choose the most interesting value for the parameter to continue the visual data exploration process with. Figure 3.2(b) shows this schematically. Users select a parameter, next a small multiples display is shown, from which the preferred value is chosen, which is shown enlarged next as a large single image. This enables a more structured navigation in the exploration process, lowering the chance important data features are missed. Also, showing all alternatives for parameter values helps users to understand the parameter, by visually showing its effect. This navigation method, however, has still some limitations. It does not prevent users from revisiting states that already have been visited before; details of small multiples are visible only after selection and cannot be compared quickly; and it is difficult to link findings early on in the exploration process to features identified later on in the exploration process.

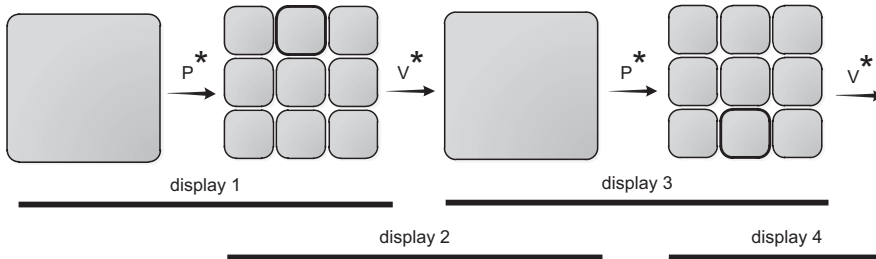
Figure 3.2(c) shows the simple solution we developed to remedy many of these issues. Instead of just one step, two steps in the exploration process are shown simultaneously. When a large single is shown on the left, users can select different parameters and see



(a) Standard approach.



(b) Alternating Large Singles and Small Multiples.



(c) Showing a Large Single and Small Multiples alternating in one display allowing for multiple operations.

Figure 3.2: Visual data exploration methods. A P represents the selection of a parameter, a V represents the selection of a value for this parameter. Display denotes what is shown on screen to the user. Note that for (c) always two sets are in display and also multiple P and V operations can be performed, hence the star.

the effects via the small multiples on the right. If satisfied, the small multiples are moved to the left, and the user can select instances of these to be shown as a large single on the right. This enables close-up inspection of the effect of different parameter values, as well as quick comparison of different settings. When a satisfactory new state is found, the large single can be moved to the left, and the parameter-value selection cycle restarts. Here, the visual exploration process is visualized directly to the user as a sequence of alternating small multiples and large singles. We expand on this, and facilitate this as follows.

Users can navigate along the exploration path with left and right buttons at both sides of the screen. Also a compact navigation trail with all important operations is shown at the bottom of the screen, allowing for fast navigation. Finally, images of the entire visualization trail can be shown or part of it using zoom-and-panning techniques. Here we use a hybrid approach of actions and states for the visual history as described by

Heer *et al.* [130]. The visual history provides users with more general advantages that enable users to:

- **suspension** – pause and resume the exploration;
- **explanation** – quickly explain how insights are gained;
- **presentation** – share their results because of the visual history and slide-like presentation technique; and combined this enables users to
- **collaboration** – share their exploration with colleagues providing them with performed actions (visual history trail), offering explanation of the current findings and enable them to continue further investigation.

The visual history trail provides users with a single *linear* exploration path. If somewhere in the trail a different operation is applied, over the already applied operation, the operation is executed and everything in the trail after this operation is lost (after user agreement via a pop-up dialog). One may prefer *branching* behavior to keep both the old and new exploration path, similar to Shrinivasan and Van Wijk [245]. This is a tradeoff between flexibility (multiple paths versus one path) and complexity (simple navigation, easy to understand). We choose for one exploration path to keep things simple and yet powerful. Finally, branching behavior can be achieved by starting a new exploration with the chosen multiple as starting point. In short, our approach of showing combinations of small multiples and large singles enables users to view (a) the effect of selection of parameters, (b) the effect of different values for parameters, and also (c) provides a natural visual history mechanism. Taken together, we expect that these enable more efficient and effective data exploration, as well as increased user satisfaction. In the following we expand on this, for a quick and lively overview we recommend to watch the accompanying video¹.

3.4.2 Generation of Small Multiples

Small multiples are created based on one large single visualization by inheriting all parameters from the large single, except for the parameter that was selected to be varied over the small multiples. We call the process of generating small multiples from a large single a *splitting* operation. In the following we describe how we have designed these for four different types of parameters: for filtering, mapping, binding, and analytics.

Filter By applying a filter-split on a large single visualization, small multiples are created based on a large single and selected attribute. The value range of the chosen attribute to split on is divided into different smaller ranges, or bins. For each of the bins a small multiple is created. The determination of the different ranges can be done either manually or automatic. For a manual division users have to provide the number of bins and/or the range for each of the bins. On automatic division the bins and ranges are algorithmically determined (see Section 3.4.3 for more detail). If the

¹<http://www.stef.vdelzen.net/dissertation>

split attribute is categorical (ordinal or nominal) then for each categorical value a small multiple is created. The data for each multiple is filtered to adhere to the according bin value(s), for both categorical and numerical attributes. Example filter-splits are shown in Figures 3.3(a) and 3.3(b).

Mapping A mapping-split creates a small multiple for each visualization type available to users, enabling them to explore what visualization type is best for their problem, as this is often an open question. This split operation gives users a powerful technique to effortlessly try different visualization types, or even try them all at once. In addition, this enables users to explore visualization types unknown to them as the brushing and linking mechanisms provide clues what the different visualization elements encode. An unfamiliar visualization type can be understood by highlighting elements in a more familiar visualization due to visual linking. Figure 3.3(c) shows an example mapping-split. Visual mappings have a variety of parameters, such as what attributes to use for the axes, color, and size; and for instance what color scales to use. All such parameters lend themselves well to generation of alternatives shown as small multiples, see Figures 3.3(d) and 3.3(e). Some parameters are mapping independent (e.g., color use), others are dependent on the mapping, e.g., the number of axes used varies over different mappings. In the menu shown on top of the large single this is taken care off.

Visual analytics Visual analytics methods can be used to enhance the exploration experience. Often it is valuable to cluster the data, however, clustering methods contain a number of parameters to be set first. Often, users do not know what the influence of the parameters on the clustering is, or worse, do not know the different parameters at all. The clustering process is therefore a highly sensitive trial-and-error process. Fortunately, we can make this process less painful by integrating the setting of these parameters also in our approach. As a proof of concept we implemented split operations for clustering the data contained in a visualization. We introduced three parameters, *clustering type*, *number of clusters*, and *cluster distance*. It proved highly useful to split on for example number of clusters to show small multiples with clustering results for 1–10 clusters. From this we can directly observe what number of clusters still makes sense for the data and which does not. Also, being able to observe the difference in clustering type helps users to understand these. With our small multiple approach we provide users with easy accessible operations to help them understand parameters and help them in choosing appropriate values through comparison and guidance. Figure 3.3(g) shows an example exploration path using clustering parameters.

Advanced split operations Split operations are not only possible on one large single, but also on a group of small multiples: *multiple split*. This enables users to, for example, create a scatter plot matrix by first applying a mapping split on the *x*-axis and next apply a *y*-axis mapping-split simultaneously on the created small multiples. Users are able to select multiple small multiples and apply a split operation on these. This breaks the simple mechanism of alternating large singles and small multiples, but was still included for flexibility at the cost of simplicity, hence, it will naturally only be used by more experienced users.

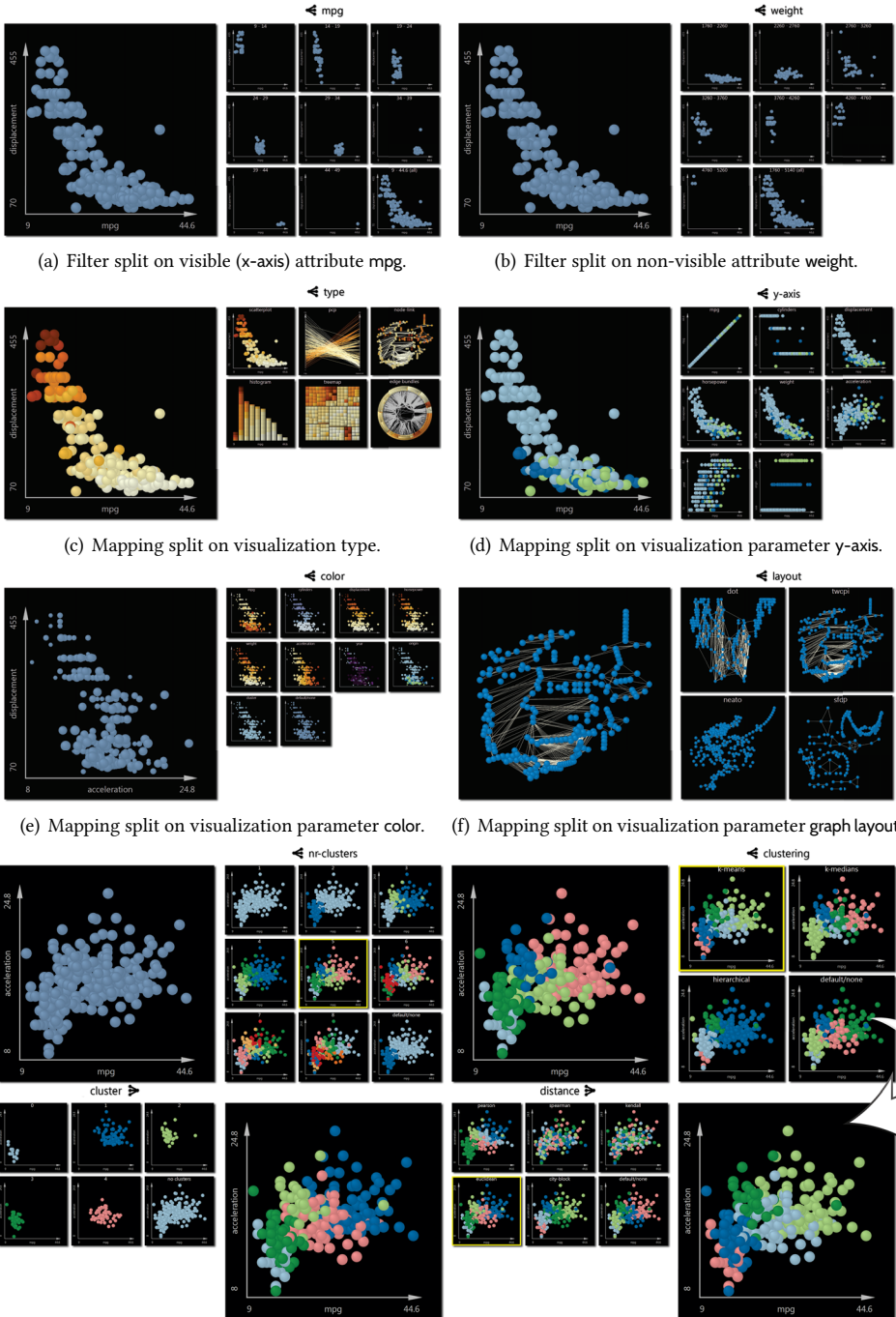


Figure 3.3: Generation of small multiples by splitting a large single.



Figure 3.4: Graphical user interface: The menu (B) on top of the large single (A) shows different parameters that can be selected to produce small multiples. Hovering over the menu items provides information on the current value such as what attribute is mapped to color, or what is shown on the x-axis for example. (C) Items are highlighted across all visualizations, identical to standard brushing and linking mechanisms in multiple coordinated view systems. A user-adjustable radius around the mouse cursor determines what items are highlighted. The number of highlighted items in the current visualization is shown in the upper right corner, along with total number of items in the visualization and total number of items in the dataset. (D) Information on parameter that is split on. Also, multiples can be sorted based on parameter value or number of items contained in each small multiple. (E–G) Navigation mechanism to explore the visual history trail. (H) Undo, redo and reset exploration options. (I) Legend with color and size information of the small multiple visualization that is currently hovered.

Also, creating small multiples for different datasets can be valuable. A *dataset split* operation creates small multiples for a set of available datasets with similar structure as the current dataset. The visualization and filter operations remain similar for the small multiples allowing for comparison of features in different datasets. Often users are interested in typical differences between datasets, for example to investigate the differences between a training and test set in a typical machine learning setting.

3.4.3 Implementation

We implemented the small multiple visual exploration method in a prototype developed using Qt/C++ that runs on Windows, Linux and Mac operating systems (see Figure 3.4 for a screenshot of the graphical user interface). There are a number of implementation details worth mentioning.

Bin range For the filter split operation we choose to keep things simple and yet powerful, therefore, we automatically determine the number of bins (number of small multiples) and the according data range values for each bin. We use the Freedman-Diaconis formula [102] for this, because it is very robust against outliers and works well in practice [153]. Next, the range values are slightly adjusted to have a nice upper and lower bound. With nice we mean we always use (1, 2, 5, 10, 20, 50, 100, *et cetera*) as start or end value of the ranges for each bin. Also, if the data varies highly on one attribute and the Freedman-Diaconis method returns a high number of bins, we cap this to display 25 small multiples at most to prevent the small multiples from being rendered too small. A different solution would be to introduce a new small multiple that displays the text *more*, and when selected provides the next set of small multiples using a pagination mechanism.

Computation To speed up the computation for the creation of small multiples for a split operation, all small multiples are created in parallel using a multi-threaded approach. First this is to increase scalability and second to not block user interaction if one small multiple takes significant time to compute, *e.g.*, a force directed layout for a graph on showing all different graph layouts for a large single (see Figure 3.3(f)).

Parameter reset In addition to implementing explicit undo and redo actions, we also create an extra small multiple, displayed last in the grid, for undo convenience. This last small multiple always resets the data filter range to *all* for a filter split and to *default/none* for a parameter split.

3.5 Evaluation

We evaluate the small multiples visual exploration method using a formal user study, testing *efficiency* and *user satisfaction*. First we define four different interaction methods. As a start we want to test our method against a traditional visual exploration system. There are two aspects in our method that are different from a standard system. Our system uses:

- Small Multiples (SM), and,
- a Dual view visual history mechanism (D).

We could evaluate our system (SM-D) solely against the traditional system (NSM-S), however, for completeness and curiosity we also test the two other variants: Small Multiples without dual view visual history (SM-S) and No Small Multiples (large singles) with dual view visual history (NSM-D). This leaves us with four different interaction methods involving large singles and small multiples (see Figure 3.5). The goal of the user study is twofold. First, we believe due to comparison and guidance the visual exploration methods using small multiples will be more efficient than existing methods and want to test this hypothesis. Second, we hope to deduce from the user study results what the best method is to integrate small multiples in the exploration process.

3.5.1 Setup

Twelve participants, eleven male, one female, with ages between 23 and 35 were recruited for the user study. Six participants are working or studying at the Eindhoven University Computer Science Department, the other six participants are working as software engineers / scientists on information visualization software in industry. The participants each got to work with the four different interaction methods. For each of the interaction methods users are provided with a new (synthetic) multivariate dataset, generated using the PCDC-tool [45]. The datasets each consist of 500 rows, 8 attributes and each data point represents a winter holiday accommodation.

Users are given five different questions to answer using each of the four interaction methods. The questions are constructed to reflect typical visual exploration tasks [13, 296, 306] and included *identify*, *correlate*, *compare*, and *cluster*. As fifth task users were asked to think of their own requirements for a perfect winter holiday accommodation and identify it. Here users were left with more freedom in the visual exploration process. The five different questions to answer were:

- **identify** – *Identify the Guesthouse that is closest to the centre of all accommodations that offers a 7 days stay.*
- **correlate** – *Is there a correlation (positive, negative, no) between price and distance to ski-area among accommodations that have a rating ≥ 70 ?*
- **compare** – *What accommodation type (hotel, chalet, pension, villa, guesthouse) offers the most possibilities among plane transportation and difficulty rating ≤ 20 ?*
- **cluster** – *Which accommodation type is similar to hotels concerning rating and distance to centre?*
- **explore** – *Think of your own requirements for an ideal accommodation and identify it.*

All questions were provided to users in a more elaborate story form. For example, the story for the first question was: *You want to go on holiday to a Guesthouse accommodation for a seven days stay. The guesthouse should be as close as possible to the city centre. Which do you pick?* For each interaction method a different dataset is presented, hence answers to the questions differed for each method. Furthermore, for each of the twelve participants the order of interaction methods was different. This is done to prevent carry-over effects such as learning and fatigue. We used a partly counterbalanced scheme between methods NSM-S and SM-D, because we expect the greatest difference there.

Finally, after completing all five tasks, users were asked to express the system's usability by filling out a questionnaire consisting of Likert-scale, ranking and open-ended questions. Note that we did not ask for opinions on effectiveness, because with each of the four interaction methods users are in principle capable to answer all questions correctly.

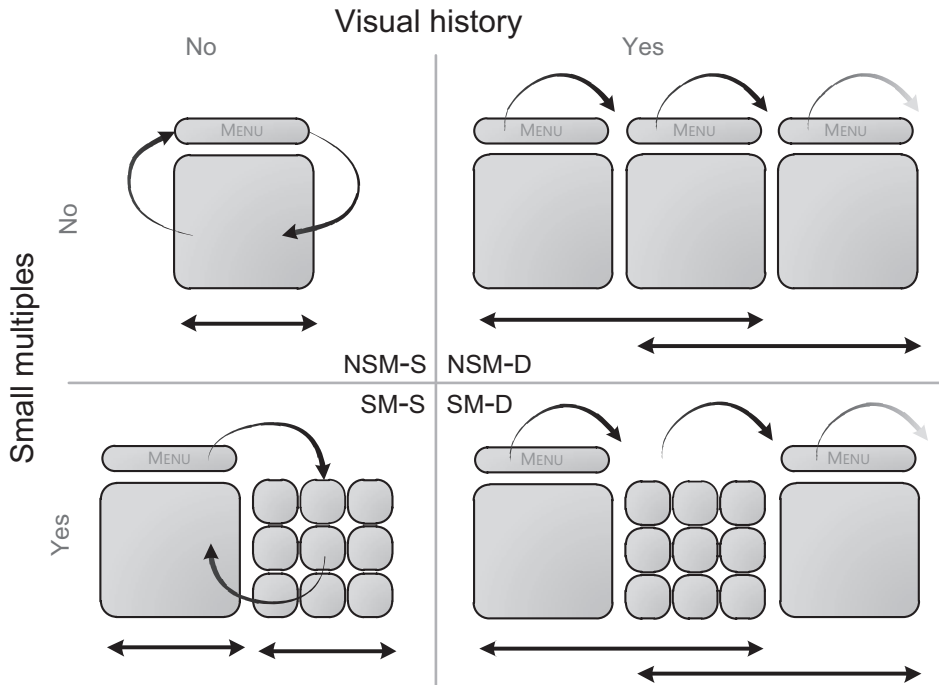


Figure 3.5: Four different tested interaction methods involving combinations of Small Multiples and Visual History.

At the start of the user study the dataset is explained to the participants. Next, the common elements in the graphical user interface are explained in detail, such as undo, redo, question submitting, visualization and menu. Visual history navigation is restricted to the use of right and left pan operations only and the selection of actions in the history trail, hence we left out zooming navigation to keep the navigation mechanism simple. Furthermore, users were provided with the standard split operations, *filter*, *mapping*, and, *analytical*. We did not provide the more advanced split operations. Users were asked to try out all explained elements to make sure they felt comfortable with these.

Next, all four interaction methods were briefly explained. Before each interaction method users were again explained the interaction method to be tested. After each method introduction they were given two questions to practice and familiarize themselves with the given interaction method. During this period users were allowed to ask questions if things were unclear. Also, they were provided with feedback on their given answers and were allowed to try to answer the question again in case of an incorrect answer.

The test took approximately one hour for each participant consisting of 10 minutes introduction, 40 minutes of performing tasks using each of the four different interaction methods and 10 minutes of questionnaire answering.

3.5.2 Results

We measured for each task: duration *time*, *steps* needed to answer question, *errors* of revisiting states, number of unique *states* seen during answering. Due to large differences in experience and exploration behavior of the participants the measures variances were high. Therefore, we normalized the results per person, per task, using fractional ranking, *i.e.*, for each person we ranked the methods for each task. Measures that compared equal received the same ranking number being the mean of what they otherwise would have for ordinal ranking. For each measure we compared the means using one-way ANOVA analysis. This test was followed by robust tests of equality of means using Welch and Brown-Forsythe statistics when Levene statistics reported significance, *i.e.*, the ANOVA assumption of homogeneity of variances was violated. Next, post-hoc Tukey HSD tests were performed to determine between which interaction methods the difference occurs. For each measure this process is repeated three times; first, we compare between the four individual methods, next differences in small multiples versus no small multiples, and finally comparison between visual history versus no visual history.

Efficiency Our primary hypothesis (H_1) is that visual exploration using small multiples is more efficient compared to no small multiples due to comparison and guidance. This means we expect less time spent per task, fewer steps, fewer errors in the navigation (revisits), and a larger part of state space explored. Our secondary hypothesis (H_2) is that we expect that exploration using the visual history is more efficient compared to single view due to focus + context in one view. Here we expect fewer navigation errors. Finally, we expect that exploration with combined small multiples and visual history is even more efficient compared to their single counterparts (H_3). Statistically significant results are shown in Table 3.1.

Although users were not faster in executing the tasks using the small multiples approach, and also no difference in errors were found, users needed significantly fewer steps and explored a significantly larger part of the state space in the same amount of time as standard approaches. Due to these results we cannot fully accept hypothesis H_1 but definitely not reject it. We did not find any statistically significant difference in the exploration methods using the visual history, therefore, we cannot confirm hypothesis H_2 . Also, hypothesis H_3 cannot be fully confirmed. We did find a statistically significant difference in task execution time and part of state space explored in favor of small multiples. For execution time of tasks in the small multiple exploration methods we also found that users were faster using no visual history. Therefore, if time is most important, small multiple exploration method should be used without a visual history. However, this might not balance out all advantages the visual history brings as mentioned in Section 3.4.1.

Satisfaction The usability is tested on three different aspects, *easy to use*, *easy to understand*, and *usefulness*. These aspects were all rated on a five point Likert scale for each of the exploration methods. In Table 3.2 we see that a majority of participants found the two small multiple approaches easiest to use. The single views were easier to

Table 3.1: Results for the ANOVA analysis on efficiency.

Measure	F-test	Post-hoc analysis
Time	$F(3, 236) = 6.205$, $p < 0.001$	Statistically significant difference occurs between the NSM-D and SM-D methods (p-value 0.009) in favor of SM-D, and between the SM-S and SM-D methods ($p < 0.001$) in favor of SM-S.
Steps	$F(3, 236) = 3.077$, $p = 0.028$	Differences occur between the NSM-S and SM-S method, in favor of the SM-S method (p-value 0.035).
States	$F(3, 235) = 352.667$, $p < 0.001$	Difference is reported between all methods, except between NSM-S and NSM-D and between SM-S and SM-D. All other differences are in favor of the small multiple variants.
<i>Small multiples vs. No small multiples</i>		
Steps	$F(1, 238) = 8.347$, $p = 0.004$	After testing Welch ($p = 0.004$) and Brown-Forsythe ($p = 0.004$) because the Levene statistics reported borderline significance $p = 0.049$ meaning that homogeneity of variances was violated. However, both are significant after robust tests of equality of means ($p < 0.05$) in favor of small multiples.
States	$F(1, 238) = 1066.966$, $p < 0.001$	Difference was found in favor of small multiples. Welch ($p < 0.001$), Brown-Forsythe ($p < 0.001$).
<i>Visual history vs. No visual history</i>		
States	$F(1, 238) = 1066.966$, $p < 0.001$	Difference was found in favor of visual history.

Table 3.2: Usability questionnaire results.

	Easy to use			Easy to understand			Useful		
	agree	neutral	disagree	agree	neutral	disagree	agree	neutral	disagree
NSM-S	6	3	3	11	1	0	8	3	1
NSM-D	6	3	3	8	4	0	7	3	2
SM-S	10	1	1	10	2	0	11	1	0
SM-D	9	2	1	9	3	0	10	1	1

understand compared to the dual view systems and finally the small multiple systems were rated most useful.

Users were asked to rank each of the four interaction methods to express their preference with respect to visual data exploration. We ran a Friedman test followed by a

Fishers exact test reporting a statistically significant result of $\chi^2(9) = 18, p = 0.03517$ and $p = 0.02387$ respectively. Both are significant ($p < 0.05$). Next, a post-hoc false discovery rate analysis is run to determine between which interaction methods the difference occurs. Small multiples with single view are preferred over no small multiples with single view ($p = 0.0262$). Also, small multiples with dual view is preferred over no small multiples with single view ($p = 0.0016$). Next, statistical significance is found for preference of small multiples over no small multiples ($\chi^2(3) = 12.667, p = 0.005416$). This translates to the interaction method preferences graph shown in Figure 3.6.

Some of the strong points of the visual exploration using small multiples and visual history pointed out by participants included: *"With this method I am having a sense of progression due to the shifting and visual history trail"*, *"Categorical split is very powerful to gain insight in group statistics"* and *"With small multiples you spot differences that you would otherwise perhaps not see / miss"*.

Finally, 11 out of 12 participants think small multiples are a good idea for visual data exploration. Also, 10 out of 12 persons express they want to use the small multiple dual view exploration method on their own data.

3.6 Scalability

If lots of attributes are present in the dataset, the menu becomes less practical because navigation to the attribute of interest takes a lot of effort. Therefore, we suggest to first filter on the number of attributes if the dataset has a high number of attributes, to select only those of highest interest. The same applies to a high number of parameters. A different solution is to develop a scalable easy-to-use menu that allows for a high number of parameters.

The small multiples exploration approach scales well with respect to the number of items in the dataset. Partly because this depends on the visualization used and second, because the proposed exploration method enables easy partitioning of the dataset by (recursively) applying filter split operations. The individual parts can then be analysed in isolation while keeping the context due to the visual history mechanism.

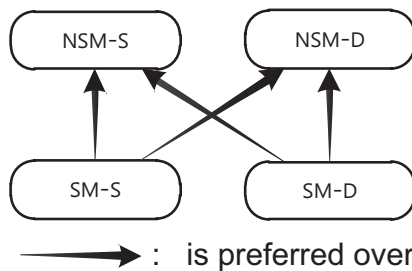


Figure 3.6: Interaction method user preference.

For complex visualizations the small multiples may become too small, however, here we aim at casual users, who want to explore data incidentally and who are not used to complex visualizations and require as simple as possible means to explore their data.

3.7 Conclusions

In this chapter we introduce a new visual exploration method for multivariate data analysis using small multiples. We introduce a model based on the alternation between large singles and small multiples. The small multiples are produced by applying split operations on large singles. We propose different split operations each having their own use. Furthermore, we introduce a navigation mechanism based on explicitly showing the visual history of the exploration path. As proof of concept the exploration method is implemented in a prototype. The effectiveness of the exploration method is tested using a formal user study comparing four different interaction methods. For efficiency in terms of execution time of tasks, no advantage of small multiples was found. Also, no fewer errors were made using the small multiples approach. However, we did find users needed fewer steps in answering the questions and also explored a significantly larger part of the state space in the same amount of time, which gives them a broader perspective on the data, lowering the chance important data features are missed.

On top of this the small multiple exploration method offers comparison and guidance simplifying and increasing the satisfaction of the exploration process. Furthermore, we found significant differences in which method users preferred to use for their data exploration. Small multiple interaction methods were preferred over their no small multiple counterparts. In conclusion, users were more satisfied and preferred exploration methods using small multiples, but if a visual history should be integrated is still an open question and needs further investigation.

Future Work There are several directions for future work. One such direction is the exploration of integrating branching behavior in the exploration method. Key challenge here is to keep the interaction down to a level that is still easy to understand, easy to use and useful. Also, the visual exploration method using small multiples offers comparison and guidance, which is most effective if users do not know what they are looking for. Due to the within subjects design this may have had an influence on the user study results. This requires further investigation and perhaps different user studies for evaluation such as think-aloud methods.



Multivariate Network Exploration and Presentation

4

This chapter is based on [274]:

“Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations.” S. van den Elzen and J. J. van Wijk. *IEEE Trans. Vis. Comput. Graphics*, 20(12):2310–2319, Dec 2014. (**Best Paper Award IEEE InfoVis 2014**).

4.1 From Detail to Overview via Selections and Aggregations

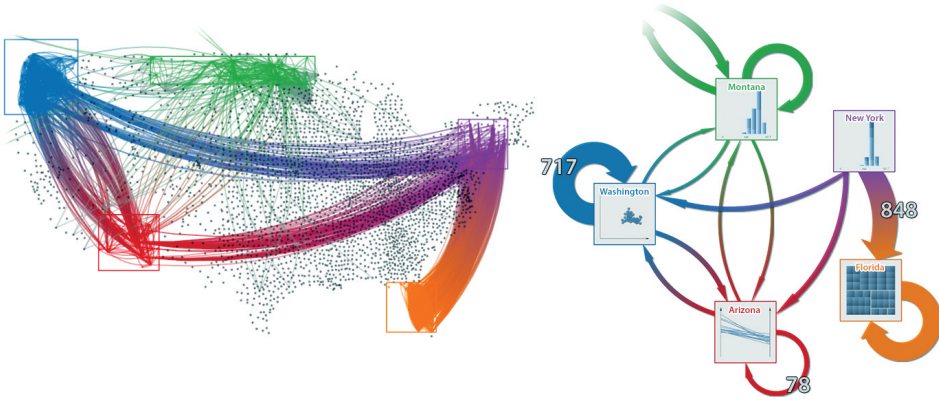


Figure 4.1: Multivariate network exploration using selections of interest, detail view (left) and high-level infographic-style overview (right).

4

MULTIVARIATE NETWORK EXPLORATION AND PRESENTATION

NETWORK data is ubiquitous; e-mail traffic between persons, telecommunication, transport and financial networks are some examples. Often these networks are large and multivariate, besides the topological structure of the network, multivariate data on the nodes and links is available. Currently, exploration and analysis methods are focused on a single aspect; the network topology or the multivariate data. In addition, tools and techniques are highly domain specific and require expert knowledge. We focus on the non-expert user and propose a novel solution for multivariate network exploration and analysis that tightly couples structural and multivariate analysis. In short, we go from Detail to Overview via Selections and Aggregations (DOSA): users are enabled to gain insights through the creation of selections of interest (manually or automatically), and producing high-level, infographic-style overviews simultaneously. Finally, we present example explorations on real-world datasets that demonstrate the effectiveness of our method for the exploration and understanding of multivariate networks where presentation of findings comes for free.

4.2 Introduction

Many real-world phenomena can be modeled as multivariate networks: e-mail traffic between persons within a company, a telecommunication network, money flowing between bank accounts, or physical objects such as airplanes flying from airport to airport or migration of people between cities. The common theme here is the connection (relation, link, edge) between objects (nodes, vertices). The number of nodes and links of real-world data is generally large, in the order of thousands. For these networks often more information on the nodes and links is available. For example, in case of a company e-mail network we know more attributes of the persons (nodes) involved, like age, gender, and job title. We also have more information about the e-mails (links) such as time-sent, header-information, and body text.

The exploration and analysis of large multivariate networks is still a challenge. Current methods are focused on either the structural aspect of the multivariate network, e.g., [299] or the multidimensional data attached to the nodes and links, e.g., [244]. However, the greatest insights are gained from simultaneous exploration, as the two might be correlated or influence each other. For example, we are not only interested in who is e-mailing to whom (structure) or whether females or males are communicating more (multivariate data), but we are more interested in whether females are communicating more with females or more with males and also between which departments and what the distribution over time is (both structure and multivariate data). For this we need to be able to inspect the attributes in context of the underlying network topology. We provide a method that enables users to explore both aspects in a uniform method using selections of interest as central element. In summary, we go from Detail to Overview via Selections and Aggregations, which explains the acronym we selected for our approach: DOSA. And also, a dosa is a spicy Indian wrap, which resonates with our aim to combine existing ingredients into a tasteful result.

Multivariate networks are commonly visualized using node-link diagrams for structural analysis [261]. However, node-link diagrams do not scale to large numbers of nodes and links and users regularly end up with hairball-like visualizations. The multivariate data associated with the nodes and links are encoded using visual variables like color, size, shape or small visualization glyphs [233]. From the hairball-like visualizations no network exploration or analysis is possible and no insights are gained or even worse, false conclusions are drawn due to clutter and overdraw. For the non-expert user, the large network visualizations are overwhelming, confusing and contain too much detail. The casual user just wants a simple (minimalistic) visualization that conveys a clear message about the relation between network structure and multivariate data. We support both expert and casual users by presenting two juxtaposed coupled views; a detail view with all low-level network elements and a high-level infographic-style overview with aggregated components. During exploration in the detail view, the high-level overview is updated automatically. Exploration and analysis is supported by defining selections of interest. Domain experts can still use advanced measures like network distance and centrality in an uncomplicated and uniform manner, while a simplified overview that can, for example, be used for communication to the non-expert user, is generated for free and can be further refined with minimal effort. The casual user is supported with intuitive controls and playful interaction that encourages to explore the network.

In this chapter we propose a novel method for multivariate network exploration and analysis. More specifically, our main contributions are:

- a tightly coupled exploration method, enabling users to explore and analyse both network structure and multivariate data associated with the nodes and links simultaneously, using
- intuitive creation and modification of selections of interest, and
- a juxtaposed detail and high-level overview, for
- effortless production of high-level, infographic-style overviews, focusing on the non-expert user.

The chapter is organized as follows. First, related work is discussed in Section 4.3. Next, our approach to multivariate network exploration and analysis is described in Section 4.4. We describe the two juxtaposed views in Sections 4.5 and 4.7, and explain how exploration is facilitated using selections of interest in Section 4.6. Next, example explorations on real-world data are given in Section 4.8 and limitations are discussed in Section 4.9. Finally, conclusions and directions for future work are provided in Section 4.10.

4.3 Related work

The most well-known and widely used method to visualize networks is a node-link diagram. Each object is represented by a dot and if there is a connection between two objects a line is drawn in between. Much work is focusing on computing two-dimensional layouts (embeddings) for node-link diagrams that best convey network topology while taking aesthetic criteria into account to improve readability [24]. Multivariate data associated with the nodes and links is commonly depicted using visual variables, such as color, size, and shape of both the nodes and links [38, 101, 204, 233, 237, 261]. Also, glyphs are used to represent the nodes [291] and motif glyphs enable structural insight [82].

As opposed to emphasizing topological properties of the network, multivariate data can be used to compute attribute-based layouts [18, 97], such as the spherical Self-Organizing Maps [302] and JauntyNets [163], to provide more insight in the multivariate data involved. Furthermore, multivariate data can be used to directly define a layout by using a scatterplot for the nodes and superimposing edges onto this, as in the GraphDice system [31]. Readability of node-link diagrams for large networks is challenging due to overlap, overdraw and clutter in general, this is aggravated further by the use of visual variables to convey associated multivariate data. A broadly used metaphor to prevent clutter in node-link diagrams are lenses practicing focus+context techniques [32, 262]. Lenses are used to enable inspection of dense areas of the network [299] and show more information for nodes of interest by displaying in-situ visualizations [162] or extract subparts of the network for further exploration [145]. Our solution also involves selections of interest, represented by boxes partially based on ideas of lenses.

A method specifically designed for multivariate network exploration and closest to our technique is Semantic Substrates [244]. In Semantic Substrates non-overlapping regions are introduced representing different categorical node attributes. In each region, nodes can be positioned directly according to the node attribute values or the positions can be computed via a force-directed layout algorithm. Edge visibility is controlled via graphical user interface controls to prevent clutter; for each region, visibility of an edge to another (or the same) region can be set. Our selections of interest are similar to the non-overlapping regions of Semantic Substrates, albeit more flexible. Semantic Substrates regions are restricted to a single categorical node attribute and link attributes are not taken into account. We support both n -dimensional regions as well as link attributes. Furthermore, link visibility is controlled globally, while we implement a more fine-grained local region control. PivotGraphs provide an aggregated view on

the network by showing two axes with categorical node attributes and positioning the nodes on the grid according to their associated attribute values [293]. This provides abstraction and a means to explore categorical node attributes supported by pivot and roll-up operations, inspired by database OLAP (*online analytical processing* [70]) actions. Unfortunately, due to aggregation, network topology is not preserved, turning structural exploration into a challenge as multiple operations need to be performed for a comprehensive image. Aggregation is also used in the GraphTrail system [81] for multivariate network exploration. Here the focus is mainly on capturing the user interaction and integrating this into a history trail. Familiar charts are shown for the exploration of the multivariate data. Pretorius and Van Wijk [208] enable multivariate network exploration by treating links as first class citizens. Link labels are placed in sequence top-to-bottom in a rectangular region centered between source and target nodes on both sides. Each node is contained in a hierarchy defined by associated multivariate data rendered as an icicle plot that is positioned on both sides of the edge labels. Next, each node is connected with a line to the according edge label. This is extended to multiple hierarchies in the Parallel Node-Link Bands approach [114]. Users can interactively inspect and query the graph, however, due to the bipartite node layout it is difficult to explore network topology.

The field of multivariate network visualization and interaction is large and we only discussed the most relevant related work. For a more complete overview, we refer to survey papers on the visual analysis of large graphs [134, 287] and a recent book on multivariate network visualization [171].

In summary, current methods are focused either towards structural exploration or multivariate data exploration. No method facilitates both the structural and multivariate analysis in a tightly coupled exploration technique. Also, no system provides users with an easy to understand simplified overview showing both structure and associated multivariate data, except for PivotGraph, but there the low-level details are missing.

4.4 From detail to overview

Large multivariate network exploration is a challenge due to size and inability to explore node and link attributes in context of the underlying network topology. Furthermore, to non-expert users a low-level visualization showing all individual elements is overwhelming, confusing and provides too much detail. They rather need an aggregated overview showing the most important components. Also, the expert user needs this as a means of communication to stakeholders. In summary, to support this, we need:

- a scalable interactive method to simultaneously explore network structure and associated multivariate data for the nodes and links using direct manipulation, and
- the ability to see both the low-level details and aggregated high-level elements, all using
- familiar metaphors.

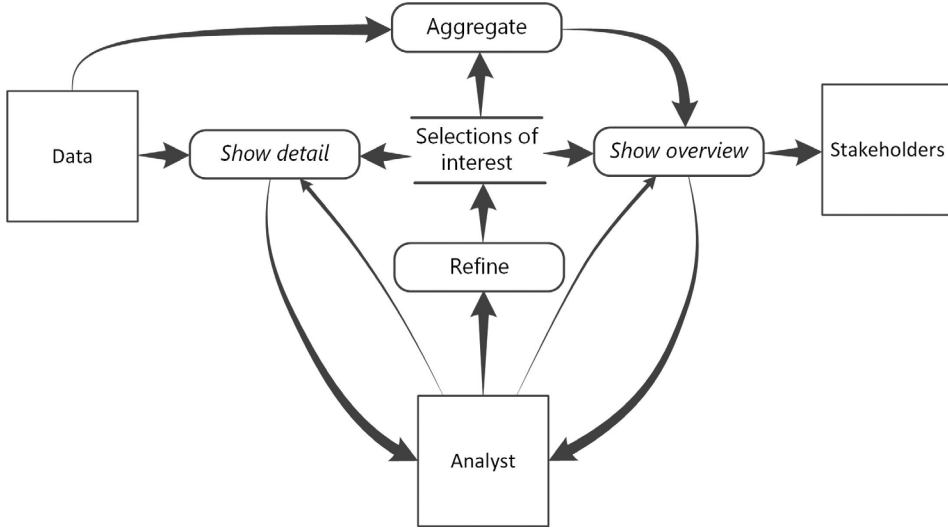


Figure 4.2: The DOSA exploration process with selections of interest as central element. The analyst (bottom) refines selections of interest, which influences both the detail and overview visualizations from which insight is gained on a low detail and aggregated high level. Finally, insights can be communicated directly to stakeholders in a simplified, infographic-style visualization conveying a clear message.

To tackle the scalability problem there are two main approaches, top-down [243] and bottom-up [278] exploration. In a top-down approach, exploration starts with an overview of the entire network. From this overview interesting features are identified and the exploration continues with a more narrow focus on sub-structures of the network. This is difficult with a large node-link diagram; due to clutter and overdraw interesting features are hard to discern. Inversely, a bottom-up approach starts with a (predetermined) single node of interest and then continues the exploration to neighboring nodes. Here we pursue a hybrid approach; we do not limit the exploration to just one node but to a number of selections of interest (see Section 4.6) that each contain one or more nodes and we simultaneously always show both the low-level detail (see Section 4.5) and high-level overview (see Section 4.7). Our novel DOSA exploration process using the described elements is schematically shown in Figure 4.2.

4.5 Detail view

To provide an overview of the network, each individual node is shown in the detail view. The position of the nodes is determined by a customizable two-dimensional projection of the network based on node attributes. Note that this is not limited to a scatterplot-like visualization, but can also be a (precomputed) force-directed network layout or more familiar geographic plot as both can be encoded via attributes of the nodes. The axes involved in the projection can be shown or hidden on demand. Animation is used to show the relation between projections upon change and maintain the notion of a unified information space. Users are enabled to freely zoom-and-pan the projection space to

navigate and explore. To prevent clutter, edges are initially not shown. We only show the edges involved in the selections of interest, see the next section for more detail. The two main approaches to depict edge direction are arrows and color, *e.g.*, [137]. Here, we choose to render edges using quadratic curves in a clock-wise fashion to convey directionality (see Figure 4.3(a)). This prevents overdraw of bidirectional edges, avoids clutter because arrow-heads do not have to be drawn and finally, the visual variable color can be used to convey a different attribute, here visual association between different selections of interest.

Alongside the available multivariate data at the nodes, we also compute structural network properties for each node such as degree and centrality measures closeness and betweenness. By changing the projection, exploration can start from an interesting multivariate data property, *e.g.*, cities with a low population and high crime rate, or from an interesting structural node property such as high betweenness, a geographical region, or a combination of these. The creation and interaction with these selections of interest is described in the next section.

4.6 Selections

In the multivariate network exploration process, users need to be enabled to focus on subparts of the network and then aggregate these to perform high-level comparison and inspection. For the selection of interesting subparts, the following candidate solutions can be employed:

- *brushing selection*: one set consisting of the current brushed items is highlighted, the rest of the items are treated as background;
- *partitioning*: multiple sets of items, supported by, *e.g.*, brushing with different colors in Xmdv [191, 290] or automatically coloring of items by conditional formatting as in Microsoft Excel.

There are two underlying principles that enable the creation of selections: *painting* and *querying*. For painting the elements are pointed at with the mouse cursor and colored accordingly. For querying, a list of predicates on attributes is specified. We want a method that is both *expressive* and *simple*. However, in general these requirements are conflicting, for example the DataMeadow approach [89] is expressive but complex. We selected to use an approach based on partitioning and (visual) querying, because this provides better support for scanning and exploring the data. We provide a visual querying mechanism to explore multivariate networks based on a node partitioning. Our solution for node selection is based on the following familiar metaphors:

1. draw boxes,
2. select ranges,
3. order selections (similar to arranging layers in Adobe Photoshop, or arranging objects in Microsoft PowerPoint), all using
4. direct manipulation.

Users can create selections of interest by adding boxes to the current projection in the detail view. The nodes belonging to this selection are the nodes contained in the box, *i.e.*, within the ranges of the two projected attributes. Users can freely reposition the box by dragging in the projection, which dynamically changes the ranges of the selection. Also, the size of the box can be adjusted via standard selection controls, such as drag handles, to directly influence the associated ranges. We support users to quickly set a similar attribute range to all selections. If this attribute option is active then the according range is synchronized over all boxes.

We choose for boxes here over other alternatives such as freeform selection to support intuitive simple interaction: the selections of interest are easy to visualize, and the boundaries of the box directly translate to ranges for the two projected attributes in the detail view, which enables intuitive and simple manipulation, in both the detail view and the Scented Widget [298] controls.

Boxes and contained nodes have a (adjustable) color for visual association. Upon changing the projection, the previously defined ranges for a selection of interest are maintained. The position, width and height of the box are adjusted to reflect the current ranges for the projected attributes. Users are enabled to shift their focus to specific boxes while maintaining a context using smooth zoom and pan methods [280].

Each selection is shown in a selection component (see Figure 4.3(d)), that has additional operations such as hide and lock, similar to the layer approach in Adobe Photoshop. Furthermore, the color and name of a selection can be changed to something semantically meaningful. The selection component provides a box selection mechanism and simultaneously serves to resolve conflicts in the selections; as a result of our partition approach, a node can only belong to a single selection. If there is overlap of the boxes, then the order of the selections is decisive. Selections higher in the list bind stronger. The sequence of the selections can be changed using drag and drop or button controls to enable fast switching of possible box configurations in the selection component. The selection order can also be influenced using a context menu with arrangement controls on the boxes in the detail view, *e.g.*, bring to front, sent to back.

Initially users are provided with a single *background* selection that contains all nodes. This serves two purposes: it provides an overview of the entire network showing dense and sparse areas to start the exploration, and it provides a context to the selections made.

The underlying formal model with technical details on the realization is described in the next section.

4.6.1 Model

We have a network $G = (N, E)$ with nodes $n \in N$ and edges $e \in E$. Furthermore, nodes can have attributes $a_i \in A_{nodes}$, also, edges can have a number of attributes $a_j \in A_{edges}$. Each node $n \in N$ has an associated attribute value v_i for each node attribute $a_i \in A_{nodes}$. Similarly, each edge $e \in E$ has associated attribute values v_j for each edge attribute $a_j \in A_{edges}$. Both the node and edge attributes can be ordinal (continuous or discrete) or categorical.

A predicate $P_k(v) : \text{dom}(k) \rightarrow \mathbb{B}$ over an attribute k is defined as:

$$P_k(v) = \begin{cases} v \in [v_{k_1}, v_{k_2}] & \text{if } a_k \text{ is ordinal;} \\ v \subseteq V_k & \text{if } a_k \text{ is categorical.} \end{cases} \quad (4.1)$$

where $v_{k_1}, v_{k_2} \in \mathbb{R}$ and $V_k \subseteq \text{dom}(k)$. $\text{dom}(k)$ represents the domain of attribute k ; for non-categorical attributes $\text{dom}(k) = \mathbb{R}$ and for categorical attributes this is the set of all possible values of attribute k .

A selection of interest $S_i \subseteq N$ is determined by a set of predicated $\{P_{k_i}\}$. To determine whether a node is in a selection, we use the order of the selections to prevent conflicts, *i.e.*, a node $n \in N$ always belongs to only one selection S_i :

$$n \in S_i \text{ if } n \notin S_j, j = 1, \dots, i-1 \text{ and } \bigwedge_{a_k \in A_{nodes}} P_{k_i}(n.v_k). \quad (4.2)$$

A node is contained in a selection if it is not already contained in a selection that is higher in the ordering, and its attribute values adhere to each of the selection predicates.

4.6.2 Interaction and Direct Manipulation

Following from the previously described selections of interest, we need to support three basic direct manipulation operations to enable visual querying:

- select current set,
- adapt range, and,
- change order.

For this we designed three different components in the graphical user interface; box visualizations representing the selections of interest in the detail view (see Figure 4.3(a)), an attribute component to adapt selection ranges (see Figure 4.3(c)), and a selection component to control the ordering (see Figure 4.3(d)).

In the attribute component, all node attributes are enlisted using Scented Widgets [298]. For each continuous attribute a scented span slider is shown and for each nominal or discrete ordinal attribute a scented selection widget is shown. In a different tab-page all edge attributes are shown, similar to the node attributes, using Scented Widgets. These attribute controls are directly linked to the current selection of interest. If a box is selected, either by pointing and clicking in the detail view or selection in the selection component, the Scented Widgets are updated to reflect the current attribute ranges or values. At any projection all attribute ranges can be adapted, also the ones currently not shown, to refine the current selection. Upon repositioning of the box in the detail view, the currently projected attribute ranges are updated.

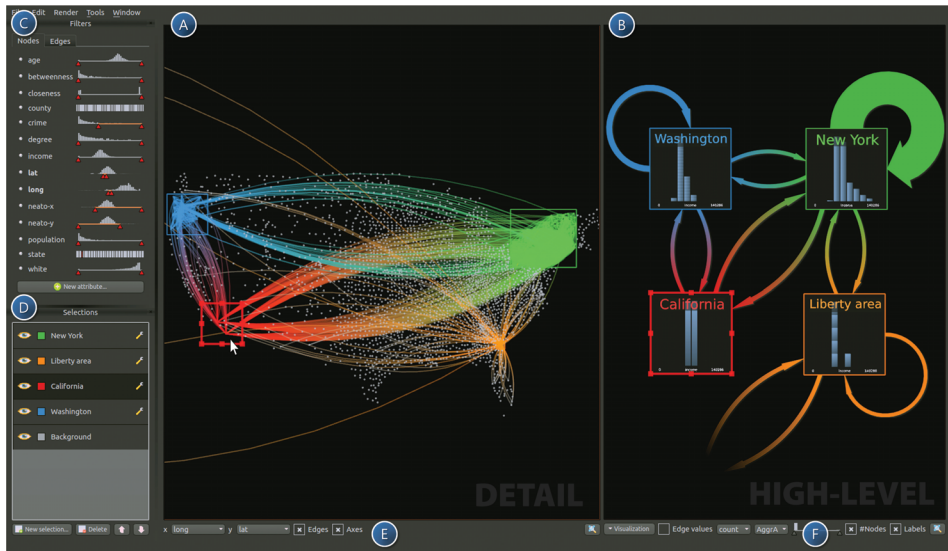


Figure 4.3: Graphical user interface of the implemented prototype showing all components: (A) Low-level detail view showing a two-dimensional projection of the nodes based on available attributes. The projection and other visual attributes can be set using controls at the bottom (E). Four selections of interest are shown in the detail view, visualized using boxes for direct manipulation. All selections of interest show between edges, the green, blue, and, orange selections also show within edges. The orange selection additionally shows edges with the background selection. (B) High-level overview showing aggregations of the selections of interest including associated aggregated edges. For each of the selections an interactive histogram visualization is shown. Visual representation and attribute mapping are configurable to users' needs with controls at the bottom (F). (C) Attribute component showing all available attributes with according Scented Widgets for the nodes and links in different tab-pages. The Scented Widgets provide information on the distribution of attributes and can be used to directly control the ranges of the multidimensional selections of interest. (D) Selection component containing a list of all selections. Selection priority (order) is controlled via drag and drop operations. Additionally, selections can be hidden or locked here.

Being able to slice-and-dice each attribute to refine the selection and directly gain feedback both on a structural- as well as the attribute-level provides users with a powerful exploration mechanism. For example, first two (or more) geographical regions of interest can be created in a latitude-longitude projection. Next, the projection is changed to age (x -axis) versus income (y -axis). Now the selection boxes can be freely repositioned and resized to slice-and-dice through the currently projected attributes while still maintaining the earlier defined geographic regions.

Note that also dynamic network exploration is supported by being able to shift through time for both the nodes and edges if time is available as an attribute. See the video in the supplemental material¹ for a demonstration of the different interaction methods.

¹<http://www.stef.vdelzen.net/dissertation>

4.6.3 Exploration

Next to the available attributes of nodes, additional derived attributes can be added based on a selection of interest. This enables, next to multivariate exploration, also exploration of the structure of the network. For example, in structural understanding it is interesting to find the nodes that are distance 1,2,3... *et cetera* away from a certain node or group of nodes, or to identify the nodes that are not reachable from a certain group of nodes. For this, users can add a dynamic attribute that computes the distance (in terms of link hops) to a selection S_i . A value or range for this derived attribute can then be set to support structural exploration. The derived attributes are dynamically updated in real-time upon changing the associated multidimensional boundaries of the selection box by running Dijkstra's shortest path algorithm [78] for all n nodes involved, having run time $\mathcal{O}(n \times |E| \log |E| + |V|)$.

For each selection consisting of nodes, there are three types of edges involved: *within*, *between* and *background* edges. For *within* edges both involved nodes are within the same selection, see Figure 4.4(a). *Between* edges have one node contained in one selection and the other in a different selection, see Figure 4.4(b). For *background* edges one node is contained in the selection and the other node is contained in the *background* selection, see Figure 4.4(c).

More formally, for a current selection of interest S_i an edge e with source and target nodes n_s and n_t respectively, has type e_{type} :

$$e_{type} = \begin{cases} \text{within} & \text{if } n_s \in S_i \text{ and } n_t \in S_i; \\ \text{between} & \text{if } n_s \in S_i \text{ and } n_t \in S_j, j \neq i; \\ \text{background} & \text{if } n_s \in S_i \text{ and } n_t \notin S_j, j = 1, \dots, n. \end{cases} \quad (4.3)$$

Here n_s and n_t are interchangeable and for between and background edges, we further distinguish between incoming and outgoing edges. Users can define for each selection which types of edges should be shown. Initially, for each new selection *within* and *between* edges are shown and *background* edges are hidden. Further filtering on the edges is supported similar to node filtering using Scented Widgets. Despite filtering options, there may be many edges involved for a selection, which clutters the view

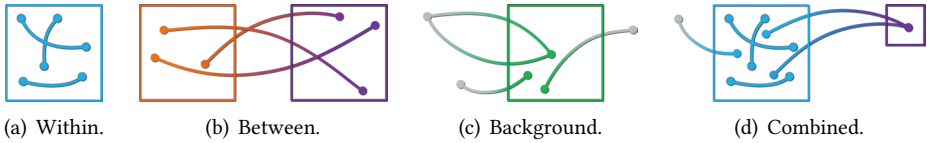


Figure 4.4: Different types of edges involved in a node selection. a) *Within* edges showing all internal connections of a selection; both source and target node are contained in the selection. b) *Between* edges show all connections between two selections; both source and target node are contained in different selections. c) *Background* edges show all connections from a selection to the background selection. d) *Combined*, showing all involved edges for a selection, within, between and background.

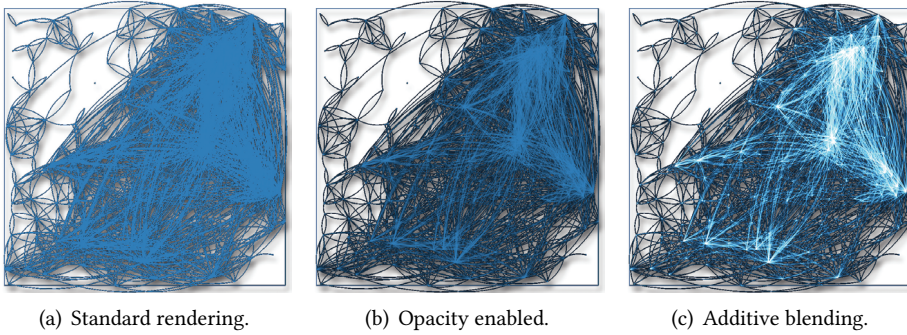


Figure 4.5: A selection of interest containing many edges that clutter the view making it impossible to identify involved nodes and local structure. We improve upon (a) standard rendering by enabling (b) transparent drawing of lines and enabling (c) additive blending.

and prevents the identification of involved nodes and hides network structure. We improve upon this by introducing the option to enable transparent drawing of edges in combination with additive blending, see Figure 4.5.

Upon the creation of a new selection by adding a box to the current projection in the detail view, a linked aggregated visual representation is created automatically in the high-level infographic-style overview, which is discussed in the next section.

4.7 High-level infographic-style overview

The high-level infographic-style overview provides users with abstraction, insight, enables communication to a broader audience, and is created semi-automatically based on the selections of interest. When a new selection box is created in the detail view, a linked visualization, sharing the same outline color, is added to the high-level overview.

The linked box shows aggregate information about the nodes in a selection. By default this is simply the number of nodes, shown textually, but also more detail can be shown. The visual representation can be changed to different multivariate visualization types. Currently we support scatter-plots, parallel-coordinate plots [146], histograms and (squarified) treemaps [46, 241]. For each of the visualizations the visual variables involved, such as what to show on the axes, can be changed to support further exploration and analysis (Figure 4.3(f)). We also support a small multiple exploration style similar to Van den Elzen *et al.* [273] (see Chapter 3) in which multiple visualizations are created, one for each value of a visual variable, to enable comparison and guidance in the exploration.

If the ranges of the selections of interest are updated then the associated visualizations in the overview are updated automatically to reflect the changes. The visualizations can be freely positioned and resized using standard editing controls found in visual editing programs. We considered using an automatic lay-out here, but since the number of

selections is typically small (2–6) and because the structure is highly dependent on the semantics, we opted for supporting manual lay-out.

The visible edges in the detail view are aggregated and also shown in the high-level overview. *Within* edges are shown as self-loops of the associated visualization. *Between* edges are rendered between the associated selection visualizations and finally, *background* edges are drawn gradually semi-transparent to the background, not attached to anything. The width of the edges is proportional to the count or sum of a selected link attribute. Initially this is the number of edges. Users are enabled to show the actual values in textual form rendered on top of the edges. Aggregated edges in the overview can be filtered by setting a range using a scented span slider. For the color we use a gradient from the colors of start selection to the end selection.

Users are enabled to freely zoom and pan the high-level overview. Level-of-detail zooming is implemented for the visualizations; if users zoom in on a specific visualization or enlarge the visualization, more detail becomes available, for example, the name of the attributes shown on the axes. Visualizations are drawn semi-transparent to enable comparison of charts by (temporarily) overlaying one visualization on top of another.

4.8 Examples and Use cases

Below we describe some example DOSA explorations of real-world multivariate network data. We show how a tightly coupled exploration is achieved by starting with either multivariate data or the underlying network topology and show this in context of the other to find correlations, anomalies and patterns.

4.8.1 US Migration and Census

United States county to county migration data was obtained from year-to-year address changes reported on individual income tax returns filed with the IRS [7]. Next, this data was augmented with geographic location of the counties and according state and finally, combined with county census data provided by the United States Census Bureau [1]. The final dataset consists of 3,221 nodes (counties) and 78,294 edges (migrations), 14 node attributes and 10 edge attributes.

By using a standard spring-embedder algorithm [112] to position the nodes of the network, we hope to see structure, such as hubs, communities, and disconnected components. However, we are presented with a typical hairball-like visualization from which no insights can be gained and exploration is impossible, see Figure 4.6(a), left. Next, we switch to a more familiar geographical plot by using a longitude, latitude projection. We add a selection box to this projection that encloses all nodes, to see whether network structure is revealed, which unfortunately, is not the case, see Figure 4.6(a), right. Therefore, the box is resized to a smaller region to enable more focus.

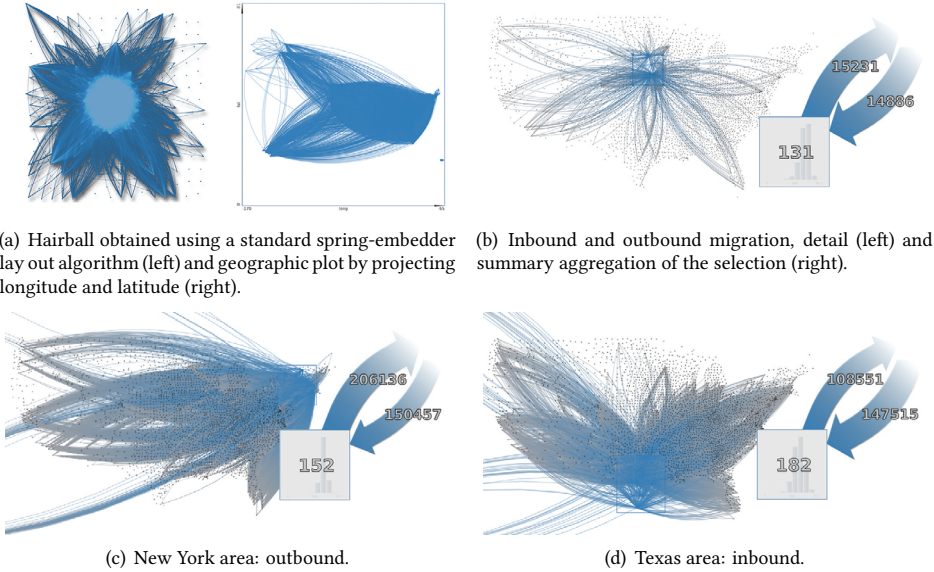


Figure 4.6: U.S.A. migration data exploration scanning for inbound, outbound and balanced regions.

4.8.2 Balance

We are interested whether there are regions or states that are more inbound, outbound or balanced. To support this exploration, we disable *within* edges and enable *background* edges such that we can directly see the total incoming and outgoing aggregated migrations for our current selection box in the high-level overview, see Figure 4.6(b).

Now, we can drag our box around in the detail view to quickly scan for unbalanced regions. We see that North-East, around the New York region, migration is more outbound, see Figure 4.6(c). In the South, the regions around Texas and Florida, migration is more inbound, see Figure 4.6(d), also Alaska is slightly inbound.

If we resize the box and extend the selection to compare West with East, we find that both are balanced. If we next compare North with South, by adding another selection box, we find that North is more outbound and South is more inbound. By adding two more selection boxes we can refine the division of the United States into four regions. Now we see, Figure 4.7, that all migrations are balanced except for the North-East region; there, migration is more outbound to both South-East and South-West selections.

We can conclude that North is more outbound due to people leaving from the North-East region. For the rest of the country migration is mostly balanced and no other anomalies are found.

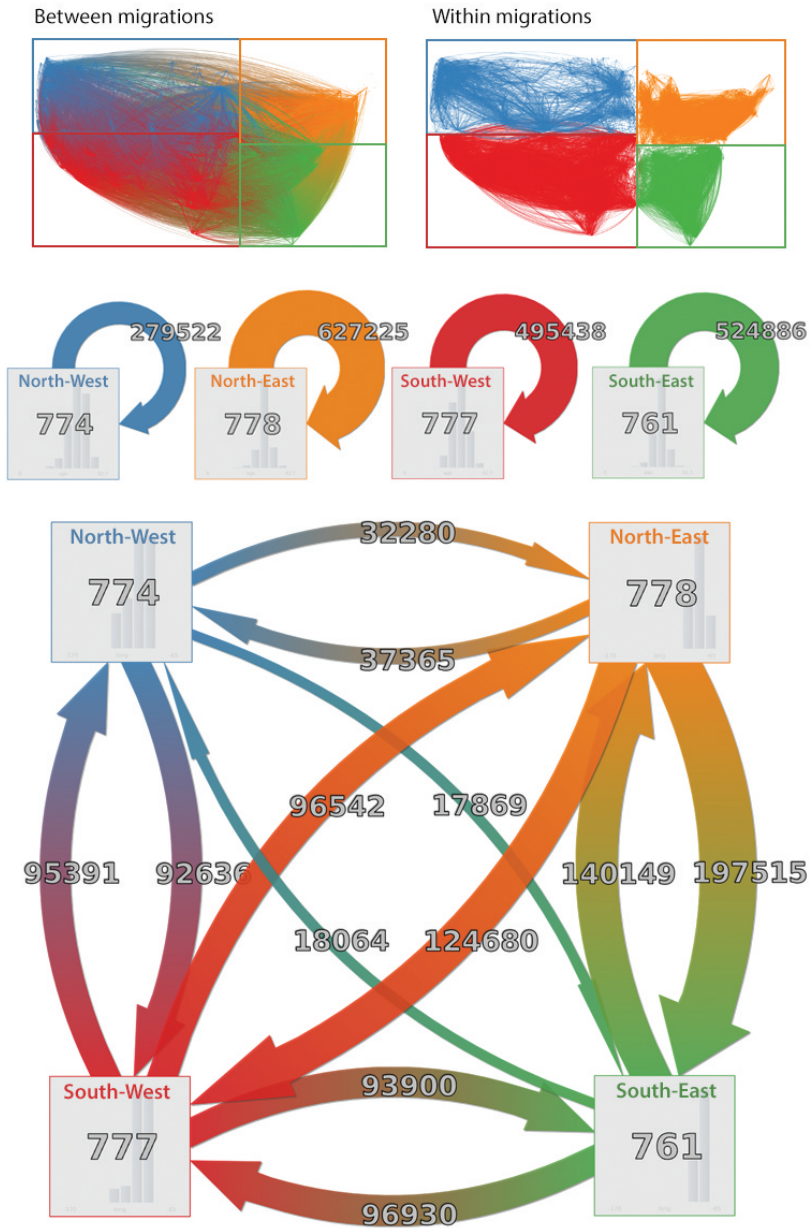


Figure 4.7: United States migration data exploration testing for predominantly inbound or outbound regions with detail (top) and overview (bottom). Number of counties in each selection, shown on the boxes, is approximately equal to achieve fair comparisons. The North-East region is outbound with migrations mainly going to the South-East and South-West regions. The rest of the migrations are fairly balanced. The North-West has the least internal migrations and North-East the most (middle).

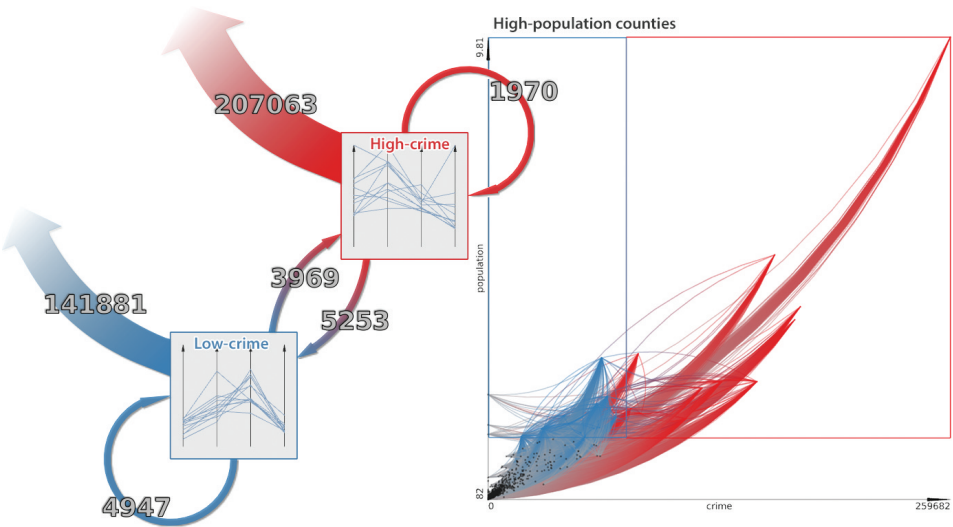


Figure 4.8: Migration of highly populated low and high crime regions.

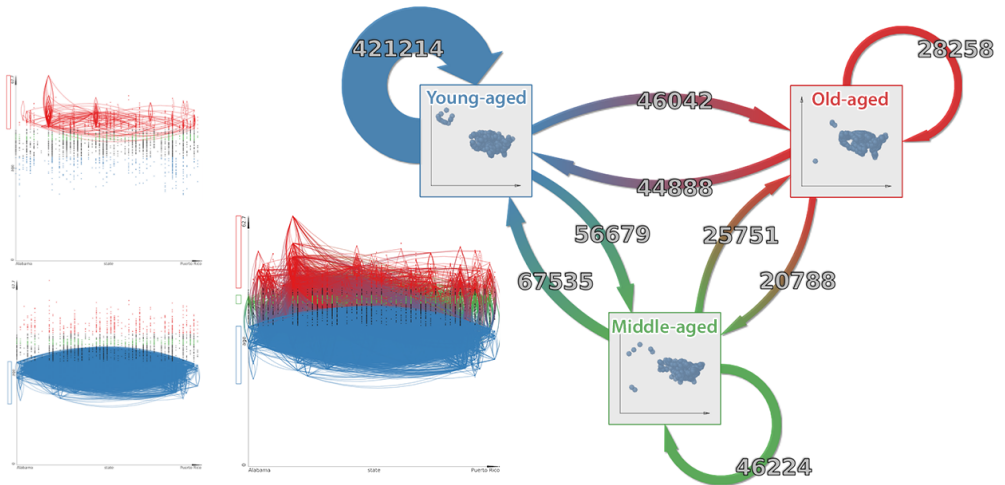


Figure 4.9: Testing for correlation between age and migration.

4.8.3 Crime

Next, we show how exploration of attribute and edge occurrence correlation is supported. For example, we want to investigate whether a high crime rate correlates with outbound migration:

- First, the nodes are positioned by selecting *crime-rate* and *population* as our current projection attributes, focusing on high population counties.
- We create two selections, dividing the top 25 highest populated counties, one low crime-rate and the other high crime-rate.
- For both we enable *within* and *between* edges.

We now see that migration from high-crime counties to low-crime counties is higher than the other way around. We also see that migration within low crime areas is twice as high as within high crime areas. If we also enable outgoing background migration we see that significantly more people are leaving from high crime counties compared to low crime areas, see Figure 4.8.

4.8.4 Age

A similar investigation can be made for testing correlation between age and edge occurrence (migration). We project state versus age and create three selections: counties where average age is low, middle and high. We keep the number of counties roughly equal in each selection to be able to make fair comparisons. From the high-level overview we see that people from low-age counties tend to move to other low-age counties. We also see that they prefer to move to middle-age counties compared to high-age counties. People from low-age counties tend to move most, followed by people from middle-age counties and finally people from high-age counties. From the overview we also see in the geographical visualizations, that most part of Alaska contains young-aged persons, also some middle-aged but is not dominated by old-aged persons. Hawaii, however, does not contain dominantly young-aged counties. Finally, Florida mostly contains old-aged counties. Also, interesting observations from the detail view can be made, we already concluded that people from young-counties tend to migrate to other young-counties and people from old-counties tend to migrate to other old-counties. In the detail view we see that there is an additional pattern; people from young-counties move to other young-counties mostly in a *different* state, while people from old-counties mostly migrate to other old-counties *within* the same state. See Figure 4.9 for an overview.

4.8.5 Visual path discovery

Now, we show how structural exploration is supported with simultaneous multivariate data analysis. Assume, we are interested in finding patterns A to B, B to C. We want to find a region (B) where people from the state New York (A) are migrating to, and also people from this region are migrating to the state Washington (C). We introduce two selections in a geographical projection; one selection S_1 filtered on the state attribute

New York, the other selection S_2 on Washington, both positioned to their according regions. Now we see the direct migrations between the two states, see Figure 4.10(top). We introduce a new attribute, distance to New York, $D(S_1)$ and add this as a constraint to S_2 , to only show nodes with distance 2 from S_1 , see Figure 4.10(middle). Next, we enable *background out* edges for S_2 and we are presented with all the counties with an inflow from New York and an outflow to Washington. We can now add another selection to be able to see the number of nodes and edges involved for one possible



Figure 4.10: Network path exploration, finding indirect routes from New York to Washington. Aggregated visual representations show number of contained nodes (counties). Aggregated edges show number of edges. From New York there are 102 possible routes to 17 counties in the Florida region, from these 17 counties another 48 routes lead to Washington. Top to bottom: (top) Direct links between New York and Washington, (middle) counties in New York that have exactly distance two to the counties selected for Washington, and (bottom) all outgoing links from the counties in the New York selection and a selection in the Florida region containing counties that connect New York with Washington.

route, see Figure 4.10. This process can be repeated to find, for example, paths of length 3. Note that while performing these structural operations we can still filter on node and edge attributes per selection for combined multivariate analysis.

4.8.6 Enron E-mail corpus

All e-mail traffic of Enron (former energy service company) corporation was made publicly available during the legal investigation of the biggest American bankruptcy due to accounting fraud [239]. The dataset is cleaned, private messages are removed, and augmented with employee function. Furthermore, sentiment analysis was performed on the e-mail body texts and added as multivariate link data. This dataset consists of 149 nodes (employees) and 185,506 edges (e-mails), 5 node attributes and 15 edge attributes.

Assume we are working at the human-resources department and want to explore the e-mail behavior of our company. First we start by projecting the nodes according to *jobtitle* (x -axis) and *degree* (y -axis). Note that we are mixing multivariate data with a structural property here. This provides an overview of the distribution of employees in the different *jobtitle* groups and also who is e-mailing most within these groups. Next, we are going to explore the e-mail behavior of the different groups. Therefore, we introduce two selections, one to select a specific group and the other containing the rest of the employees. For the latter selection we disable *within* edges to be able to focus on the *between* communication. We use the first group to shift through the different function groups and see that:

- CEOs are more sending e-mail;
- directors are more receiving;
- managers are heavily biased towards sending e-mail; and
- the managers stand out because they have a large self-loop in the overview.

From the overview visualization we see two persons having an unusual high degree. We identify the person with highest degree via details on demand in the visualization and refine our selection to only contain this person, see Figure 4.11. Now it becomes clear, by refining the selection to show *cc* and *to* e-mails, that the high self-loop is indeed due to this person who *cc*-ed himself 9,422 times and directly sent himself another 155.

4.8.7 C-Level communication

Now assume we want to inspect CEO communication behavior. Therefore, we introduce another few selections and only show communication from and to the CEOs. We see that CEOs are mostly communicating with Vice presidents and managers. However, we also see that regular employees are communicating with CEOs, which is not what we would expect. By refining our selection we see that it is only one person, Jeff Dasovich, who is heavily communicating with the CEOs and mainly broadcasting, see Figure 4.12(left). By filtering on the sentiment attributes we also see that his e-mails are mainly negative. After googling it appears that Jeff Dasovich is Enron's Government

Relation Executive, who had to communicate to the CEOs when things went wrong for Enron, mislabeled here as a regular employee. If we now keep this configuration and refine the selection to contain only e-mails with strong problem sentiment in it, we see it contains only one-way communication of the vice-presidents and Jeff Dasovich to the CEOs, see Figure 4.12(right). We refine the selection again on time and see that these e-mails were mostly sent during the critical period for Enron.

4.9 Discussion and Limitations

The basic ideas presented in this chapter are all simple in nature: (1) combined structural and multivariate exploration and analysis through visual queries using (2) selections of interest, based on (derived) node and link attributes, controlled by Scented Widgets and playful interaction on two (3) juxtaposed and linked views showing network detail using (derived) attribute projections and showing a high-level infographic-style summary overview simultaneously. However, combined they are novel and enable a strong visual query mechanism that is intuitive and effective.

In an earlier stage of the research, the prototype only contained a single view: the detail view. The aggregated visualizations in the overview were directly rendered on top of the selections of interest. However, this had several disadvantages; upon dragging the selection it was difficult to track changes in the visualization due to the motion and also internal edges of a selection were no longer visible due to the overlapping visualization. Therefore, it was decided that two juxtaposed views would benefit users in the exploration and analysis. This enabled also the possibility to have the easy to understand high-level overview as a means of presentation and communication to people who are not interested in the low-level details.

We also consider it a strong point of our system that it is simple in design and not much explanation is needed. Quite some effort is put into keeping interaction intuitive, uniform and minimalistic, *e.g.*, similar interaction methods in both views (select, drag, change size), similar interaction to deal with node and edge attributes (uniform Scented Widgets), uniform and combined interaction for structural and multivariate exploration based on attributes. Also, visual coherence between the different components is achieved by using color. Visual elements are kept to a minimum to reduce visual noise and support a broad audience.

There are, however, some limitations such as scalability with respect to the number of attributes and scalability with respect to the number of selections. We believe that in general users are content with about 5 or at most 10 selections of interest in order to answer their questions and still being able to understand the involved complexity. But if the number of selections of interest becomes large, for example due to automatic clustering or community detection algorithms, then both the detail and overview become cluttered. In the overview, edges below a certain threshold can be filtered as well as in the detail view, however, this is only partly a solution.

Currently we are relying on a node partitioning and nested or overlapping selections of interest are not possible. Enabling this allows for more powerful queries but also

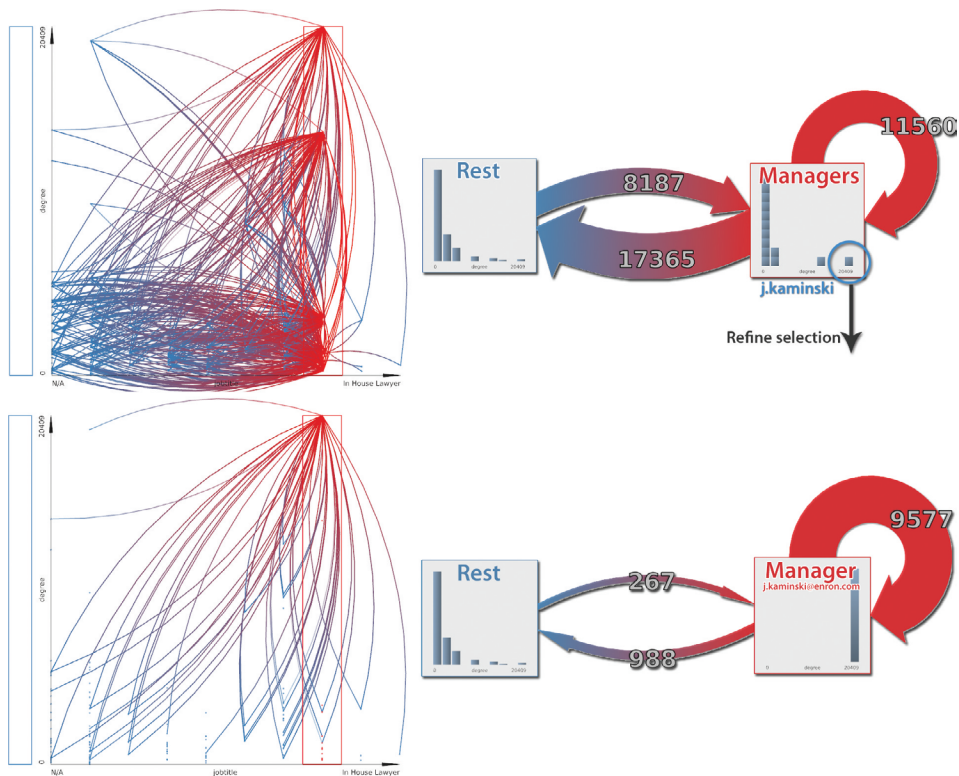


Figure 4.11: Enron e-mail communication exploration using two selections of interest: one representing the managers, the other the rest of the employees. Managers stand out due to a large self-loop (11,560 e-mails). After refinement the cause appears to be a single manager e-mailing himself all the time (9,577 e-mails).

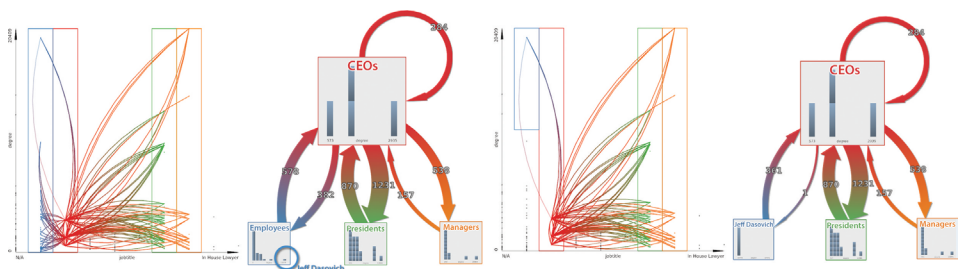


Figure 4.12: Typical C-level communication: CEOs are heavily communicating with Vice presidents and managers. However, also communication is present between CEOs and regular employees, which turns out is only one person heavily broadcasting to the CEOs (right).

increases interpretation difficulty and selection mechanism conflicts. Also, since we are relying on ranges as multidimensional boundaries for the selections, we only allow for box-shaped selection widgets. Brushing capabilities could be introduced such that

users can freely color nodes for a certain selection. However, the uniform selection mechanism based on Scented Widgets would break, as now nodes are no longer identifiable by range but only based on unique identifier. This implies that potentially a large number of gaps appear in the attribute ranges making interaction of the Scented Widgets a challenge.

If the number of attributes associated with the nodes and edges is large, the list of Scented Widgets becomes difficult to interact with due to a large scrolling area. Also, the number of available projections for the detail view quickly grows (n^2 , for n attributes) making it more likely that interesting features in the data are missed. A solution here could be feature selection, to only select the most interesting features, as a preprocessing step of the data before loading it into the tool either automatic or manually using visualization techniques [31, 87, 232].

4.10 Conclusions

4

We presented novel interaction methods for both domain-expert and casual users to explore and analyse multivariate networks concurrently on network topology as well as the multivariate data. This enables users to see outliers, patterns and trends for the combined elements. Furthermore, we support users in the simultaneous creation of a high-level infographic-style overview. This helps in understanding the network due to a simple image, provides abstraction and aggregation and presents a means for communication to a broader audience. Both interaction methods are facilitated by using selection sets as a central element and the juxtaposition of detail and overview. We have shown the effectiveness of our DOSA approach through several elaborated examples on real-world datasets. Furthermore, we have shown this method is not just limited to multivariate networks, but also functions when only multivariate data or network structure is available. Finally, due to the general and flexible setup, this method is domain-independent.

4.10.1 Future work

For future work it is interesting to enable the partitioning of the network not only on the nodes but also on the edges. This could enable richer exploration by showing aggregate visualizations also for the edges. However, intuitive interaction and facilitation of this is not trivial. Also, the adaption of the interaction methods would benefit from the enhancement of the detail view to support different visualizations such as a matrix visualization or ultimately generalize the techniques to any network visualization. Finally, functionality could be added to export the high-level infographic-style overview to an external editing tool such as Microsoft PowerPoint or Adobe Illustrator for further fine-tuning, editing and enrichment, *e.g.*, for publication.



Massive Mobile Phone Data

5

This chapter is based on [267]:

“Exploration and Analysis of Massive Mobile Phone Data: A Layered Visual Analytics approach.” S. van den Elzen, J. Blaas, D. Holten, J.-K. Buenen, J. J. van Wijk, R. Spousta, A. Miao, S. Sala, and S. Chan. *In Proc. 3rd Int. Conf. Analysis of Mobile Phone Datasets*, Cambridge, MA, May 2013. **(Best Visualization Award D4D 2013)**

5.1 Exploration and Analysis of Massive Mobile Phone Data

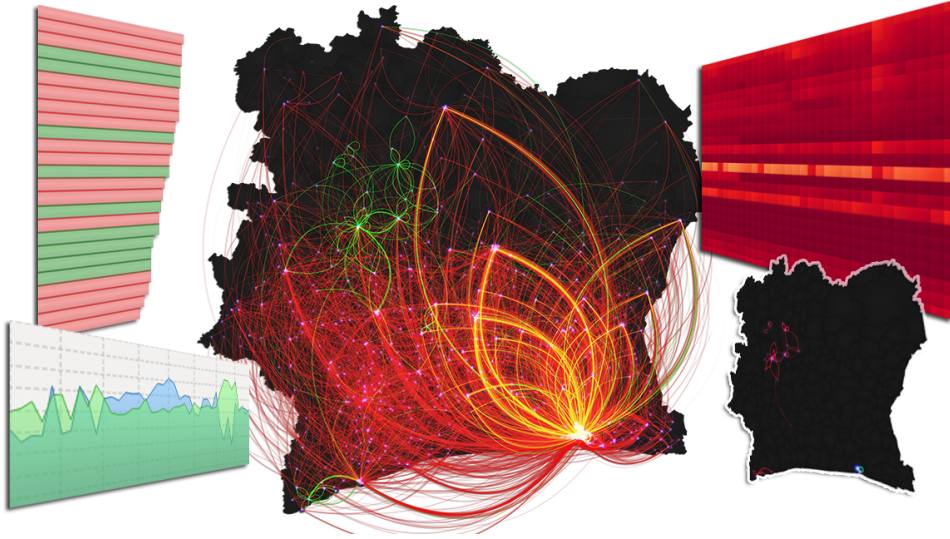


Figure 5.1: Different coherent components of the visual analytics solution for Massive Mobile Phone Data exploration and analysis in context of the Orange Data for Development challenge.

5

MASSIVE MOBILE PHONE DATA

WE present a system for the exploration and analysis of massive mobile phone data that enables users to gain insight. First we identify user tasks and develop a system following a visual analytics approach by tightly integrating visualization, interaction and algorithmic support. The system is then evaluated by exploring a massive mobile phone dataset containing 2.5 billion calls and SMS exchanges between around 5 million users located in Ivory Coast over a period of 5 months. As a use case, we show how major events are correlated with localized increase and decrease of calls.

5.2 Introduction

Four datasets on mobile phone communication were released in the context of the Orange *Data for Development* (D4D) challenge. The D4D is an open innovation *Big Data* challenge from Orange to the world research community, to help the societal development and welfare of the population by finding insights from mobile phone communication. The datasets are based on 2.5 billion anonymized Call Detail Records (CDR) of phone calls and SMS exchanges between five million of Orange's customers in Ivory Coast between December 1, 2011 and April 28, 2012 [35]. In this chapter we focus on the first dataset: tower-to-tower traffic. Data is provided to us in tab-separated-value (tsv) file format. For each hour in the timespan we are given the number of calls and duration (aggregated) between any pair of towers. Additionally, we are provided with the geographic location (longitude and latitude) of each cell tower. Initial data

cleaning was performed by Orange Labs in Paris, such as removing double entries, new subscribers, and communication to other providers. We further cleaned the data by removing entries that had missing tower identifiers.

In this chapter we describe how we support the analysis and exploration of massive mobile phone datasets by identification of user tasks and according requirements for a visual analytics prototype. Next, this is implemented and applied to the real-world mobile phone data that were provided. We present a system for the exploration and analysis of massive mobile phone data that enables users to gain insight on different levels of abstraction both in time and space. The prototype provides a smooth user experience despite the massive amount of data. We implement a visual analytics approach adhering to the visual analytics mantra: *analyse first, show the important, zoom, filter and analyse further, details on demand* [167].

Visual Analytics is the science of analytical reasoning facilitated by interactive visual interfaces [259]. It aims at an integrated approach combining visualization, human factors and automated data analysis using methodologies from information analytics, geospatial analytics and scientific analytics to effectively support the decision making process [168].

One opportunity for the analysis of such data is to detect local events (e.g., political, social, weather), assuming that these lead to changing call behavior. In this chapter we adopt this as the main use case. Correlations of call change behavior and local events are successfully identified using the prototype. Therefore, we believe *call change* occurs due to *major events*. We found both an increase and decrease in the number of calls over locally concentrated communication channels strongly correlated with events.

The chapter is organized as follows. First, user tasks and according requirements to the visual analytics approach are discussed in Section 5.3. Next, we discuss related work in Section 5.4. A layered visual analytics approach is presented in Section 5.5, where the different components of the system are discussed in detail. Section 5.6 portrays typical use cases utilizing our approach. Finally, conclusions and directions for future work are given in Section 5.7.

5.3 Design Principles

In this section we identify different user tasks, derive requirements, and discuss design decisions following from this. We believe that change in call behavior occurs due to major events. Therefore, to detect events, we focus on call change derived from the first D4D dataset (tower-to-tower communication).

The user tasks can be categorized into higher level tasks: **exploration**, **analysis** and **presentation** of massive mobile phone data. The main goal of **exploration** is to *gain insight* and to *form hypotheses*. The main goal of **analysis** is to *confirm or reject hypotheses*. While performing analysis, visualization is not only supportive but can also raise new questions, therefore users typically switch often between exploration and analysis during data exploration. **Presentation** is needed to *convey findings* to both expert users and a broader audience. In order to support this, familiar visualizations are needed.

Table 5.1: *Requirements to support users in the exploration and analysis of massive mobile phone data. This list is not exhaustive and can be extended further; however, we believe the system should at least support these user tasks.*

	Task <i>User wants</i>	Requirement <i>the system should</i>
Exploration	identification of: higher level communication channels; changes in call behavior, this because we believe that a change in call behavior occurs due to major events; and communities (cell towers with similar behavior over time).	provide an overview of communication channels both in space and time; provide an overview of call change behavior, again, both temporal and spatial aspects should allow for exploration; provide (algorithmic and visualization) support for community identification.
Analysis	perform comparison of: multiple levels of abstraction for time, space, and, visualization; multiple points in time; and multiple visualizations of similar or different data dimensions.	enable effortless switching between different abstraction levels; enable for simultaneous comparison of multiple time points; enable for simultaneous comparison of multiple views and data dimensions.
General	interactively browse through different portions of the data; being guided by complex patterns and correlations; and do all of this in real time.	provide appropriate visualizations for each abstraction level that; emphasize clues for further navigation to; provide a real-time smooth exploration user experience.

Table 5.1 provides an overview of more detailed tasks and requirements. In addition to these requirements we aim for effortless switching between the exploration, analysis, and presentation tasks. In summary:

- data has to be shown at various levels of aggregation, both temporally and spatially;
- data has a temporal, a geospatial, and a network character; all have to be shown;
- we use multiple linked views for this;
- details have to be shown on demand;
- where possible, use automated methods to simplify analysis and reduce the amount of data;
- features like overall call behavior, call change and communities have to be clearly visible;
- where possible, use familiar mappings and metaphors for easy understanding.

In Section 5.5 we present how the system implements these requirements and discuss the individual components and their integration and coherence.

5.4 Related Work

We briefly discuss related work to place our work in context, and to motivate the development of a new visual analytics prototype: no tool exists that fulfills our requirements.

Many approaches are explored using only automated methods offering no interaction and visualization, *e.g.*, [23, 84, 175, 264].

A visual analytics system, developed by Andrienko *et al.* [11, 15], for extracting place histories from mobile data, combining geovisualizations, geocomputations and statistical methods allows for the exploration of spatial, temporal and thematic components of the data. However, the main focus here is on social aspects and place extraction and does not allow for the exploration of call change inferred from events. Similar approaches, not exploring call change or event detection are discussed by Kwan and Lee [178] and Sagl *et al.* [222]. A system developed by Correa *et al.* [72, 236] also mainly focuses on social behavior patterns. Also, the geographical component is not taken into account and no algorithmic support is offered. Temporal communication patterns in mobile call graphs focusing on structural network change are discussed by Ye *et al.* [304]. Höferlin *et al.* [136] focus on individual trajectory movement exploration extracted from mobile data. A more general visual analytics system for the exploration of spatio-temporal data not tailored towards mobile data is presented by Von Landesberger *et al.* [285].

Egocentric temporal exploration of CDR data is performed by Qi *et al.* [303]. However, the geographical content is not taken into account and cannot be visualized. Furthermore, the method focuses on egocentric exploration and does not allow for an overview of the (higher level) call patterns. Egocentric exploration of temporal call behavior focusing on regions rather than individual towers is explored by Blondel *et al.* [34] in their web-based Geofast tool.

5.5 Visual Analytics Approach

In this section the developed prototype enabling the effective analysis and exploration of massive mobile phone data is presented. The prototype application follows a visual analytics approach using multiple coordinated views that tightly integrates visualization, interaction and automatic computation methods. A combination of visualization and automated methods is used, because purely visual methods fall short due to scalability issues; the data provided is large and screen space is limited. This can partially be overcome by interaction methods such as zoom, pan and filter techniques. However, this leaves less apparent patterns in the data hidden. Also, purely automatic methods fall short due to aggregation of results and loss of context. Furthermore, automatic methods are often highly focused and designed for one specific task, not allowing for the exploration and discovery of unexpected patterns. A system effectively integrating visualization, interaction and algorithmic support leverages the benefits of the individual parts.

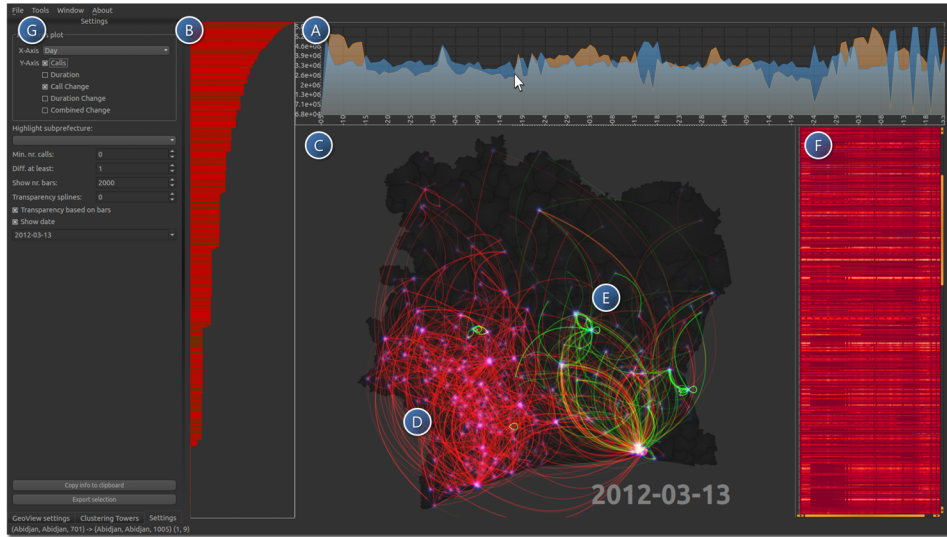


Figure 5.2: Graphical user interface: providing high level overview of different measures such as the number of calls and call change over time in the measure overview (A). Individual contributions of communication channels to the measure at a selected time interval are shown in the measure contribution view (B) and in the geospatial view (C) revealing localized call change behavior: large decrease (D) and large increase (E) of the number of calls. The matrix view (F) provides a high level overview of call behavior over time for the individual towers. Different settings and algorithmic support are offered by according controls (G).

Initial data transformation steps were taken to be able to implement the requirements as defined in Section 5.3. These steps are discussed next, followed by a detailed description and discussion of design decisions of the individual coordinated views and their integration into the system (see Figure 5.2 for a screen-shot of the graphical user interface).

5.5.1 Analyze first, show the important

The data provided consists of 3.9 GB (zipped) and 33.5 GB (unzipped) tsv files. Clearly, this does not fit into memory. In order to provide a real-time exploration experience to users different techniques are employed. Here we choose for a combination of *pre-computation*, *divide-and-conquer*, and *load-on-demand* strategy. The tower-to-tower communications, mainly focused on in this chapter, were provided in ten separate tsv files each spanning a period of two weeks. To obtain manageable data, both for overviews and detailed inspection, we performed the following preprocessing steps. We first processed the data using scripts taking an advantage of a line-by-line streaming approach that *divided* the large files into smaller files, each containing the data for one day. Furthermore, the data lines in the smaller files are sorted descending on number of calls between any pair of towers. Finally, separate files were created for calls and

duration. This process is repeated to create files on different abstraction levels such as weeks and months. Instead of loading everything into memory at once, smaller chunks can now be loaded on demand if detailed information is requested. The relatively small file sizes allow for real-time exploration. In addition, data can also partially be loaded, requesting only the most important data, because the files are internally sorted.

In addition to splitting and organizing each file, different metrics were identified for which an overview needs to be provided. These derived measures were then pre-computed to be shown as a line graph in the measure overview (more on this in Section 5.5.2). The pre-computed derived measures are *number of calls*, *duration*, *call-change*, *duration-change*, and, *combined-change*. Each of the measures are pre-computed for the different abstraction levels (*days*, *weeks* and *months*). The number of calls measure and duration measure aggregate per abstraction level the total number of calls and total duration respectively. The call-change, duration change and combined change are computed based on the extended Jaccard index [154, 257], rather than taking the absolute difference of the number of calls for two points in time. This has the advantage that a large change is reported when the number of calls goes from 100 to 1000 but also if the number of calls goes from 1 to 10 for a certain communication channel. Let E be the set of all cell tower pairs having communication on one or more points in time. For each pair of cell towers involved ($e \in E$), at two points in time t_x and t_y we compute the *individual* call change ICC_e :

$$ICC_e(t_x, t_y) = 1 - \frac{M_e(t_x)M_e(t_y)}{M_e(t_x)^2 + M_e(t_y)^2 - M_e(t_x)M_e(t_y)}, \quad (5.1)$$

where $M_e(t_x)$ gives the value of the according measure (here number of calls) at time point t_x for cell tower pair (communication channel) e . If $M_e(t) = 0$, 1 is used as measure to prevent a final value of 0. Next, all individual call changes ICC are summed and divided by the number of involved tower pairs $|E|$ to provide a final call change value CC for two points in time:

$$CC(t_x, t_y) = \frac{1}{|E|} \sum_{e \in E} ICC_e(t_x, t_y). \quad (5.2)$$

Duration change and combined change (calls + duration) are computed in a similar fashion. In addition to the change values CC we also store the individual change values ICC and again sort these descending. These files are later loaded-on-demand if additional information is required on the aggregated measures, e.g., to provide information on the contribution of each communication channel to the aggregated measure. By pre-computation of these measures we can provide multiple overviews that allow for a smooth real-time exploration of the data.

5.5. VISUAL ANALYTICS APPROACH

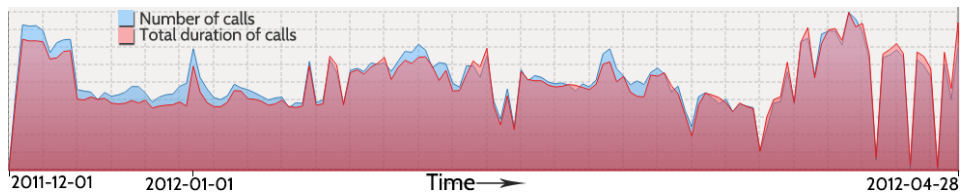


Figure 5.3: Measure overview simultaneously rendering multiple measures enabling high-level correlation exploration.



(a) Measure contribution view, providing information on most important contributors (communication channels) to the according measure and selected point in time.

(b) Non-clustered (left) and clustered (right) matrix view grouping towers with similar communication behavior over time.

Figure 5.4: Measure contribution (a) and temporal matrix views (b).

5.5.2 Measure overview

In the measure overview line graphs are shown. Users can select what to show on the x -axis and y -axis. On the x -axis different aggregation levels of time can be set; users are enabled to choose one from *days*, *weeks*, and *months*. On the y -axis (a combination of) different measures can be conveyed such as number of *calls*, *duration*, *call change*, *duration change*, and *combined change*. Showing multiple measures in the overview enables users to explore correlation between these. For example, in Figure 5.3, we see that the number of calls in the D4D-data highly correlates with duration.

The measure overview provides a high-level overview, and can be used to detect interesting points in time that require further investigation. Depending on the chosen measure one can focus on (any combination of) curves, peaks, and dips. For example, if call change is selected, users can identify points in time where change is high by focusing on peaks. The identified points can then be further explored in detail using the different linked views.

On mouse hovering the according measure value is highlighted and the actual value is shown. Furthermore, the aggregated measure for the selected time point is broken down into individual values that are shown in the measure contribution view.

5.5.3 Measure contribution view

The measure contribution view shows the individual contributions to the aggregated measure of the selected time point in the measure overview. The individual contributions of communication channels (antenna-to-antenna) are shown as horizontal bars (see Figure 5.4(a)). The horizontal bars are sorted from highest contribution (most important) to lowest contribution to the aggregated measure value. Each bar shows the region, department, sub-prefecture and identifier of both the sending and receiving cell towers. Furthermore, the number of calls (or a different selected measure) over this communication channel are shown along with the measure of the previous day for comparison purposes. This difference is also encoded as bar color; red or green indicates that at this point in time there were less or more calls compared to the previous point in time. By default only the fifty most important contributors are shown. Users are enabled to adjust this value to their likings. In addition we provide users with filtering options. Filtering is possible on the minimum number of calls required or the minimum difference of the number of calls between the previous and current point in time. This enables users to focus on communication channels with low, average or high activity.

All communication channels shown in the measure contribution view are also shown in the geospatial view for spatial identification. By hovering over an individual communication channel in the contribution view, the according channel is also highlighted in the geospatial view.

5.5.4 Geospatial view

The geospatial view displays the map of Ivory Coast. On top of this map all communication channels and involved cell towers are rendered that are currently shown in the contribution view. The communication channels are rendered as arcs. The direction of the communication is encoded clockwise. Also here, color depicts whether the number of calls (or a different measure) is lower (red) or higher (green) compared to the previous point in time. The opacity of the arcs depend on the contribution value, similar to the length of the bars in the contribution view; the more important a link is, the higher its opacity. This emphasizes the more important communication channels for easy identification. On mouse hovering, the according region, department and sub-prefecture are shown. Zoom-and-panning mechanisms can be used to navigate the map and focus on specific regions.

Technical details The arcs are rendered as quadratic Bézier curves. First the vector from source to destination point is determined. Next we compute the vector orthogonal to this vector with half the length and position it halfway between the source and destination point. Now we take the endpoint of this orthogonal vector as the control point for the quadratic Bézier curve. Due to the computation of the orthogonal vector, taking source and destination point into account, the clockwise direction is automatically inferred. The towers are rendered as white dots with a radial gradient from white opaque (innermost) to full transparent blue (outermost). Finally, additive blending techniques are used to render the towers and arcs on the map to create a

subtle aesthetically pleasing glow effect in dense areas. In addition, this allows users to differentiate between increasing and decreasing traffic; overlapping arcs where one is increasing (green) and the other decreasing (red) are rendered as yellow.

5.5.5 Matrix view

The matrix view provides a holistic overview of the behavior over time for each individual cell tower. On the vertical axis all cell towers are shown. The horizontal axis denotes time. Each row represents the behavior of one cell tower over time. On the intersection of a tower and point in time a rectangle is drawn. This rectangle represents a measure value, *e.g.*, number of calls for one antenna at one time interval. The rectangles are rendered using a heat-map technique; each rectangle is colored based on the according value, here we use a dark-red to yellow to white colormap; dark-red represents the lowest value, white the highest. Zooming-and-panning are provided to navigate and explore the matrix.

The matrix enables users to identify towers with similar behavior over time. However, because of limited screen resolution this poses serious difficulty on the task. Therefore, users are enabled to interactively apply clustering methods to the rows shown in the matrix. Once clustered, the rows of the matrix are re-ordered (see Figure 5.4(b)). Towers that belong to the same cluster (all having similar behavior) are grouped together. In addition, the clusters themselves are also sorted based on cluster-size. We offer cluster parameters to users, which can interactively be adjusted. The result of a parameter change is directly reflected in the matrix view by reordering. The parameters available to users are cluster method (*e.g.*, hierarchical, k-means, k-medians), distance metric (*e.g.*, Euclidean, Manhattan, Pearson, Spearman, Kendall), number of desired clusters, and time period to take into account while clustering.

If a clustering is applied on the matrix view, users are enabled to only show clusters of interest. Furthermore, the geographical location of the towers belonging to a cluster can be shown or hidden in the geospatial view.

5.5.6 Linking and integration

From each of the initial views additional views can be opened for the inspection of details. For example, if one communication channel is identified in the contribution view, an additional overview can be created showing the number of calls (or different measure) for the entire timespan to verify outlier behavior. Similar, the number of calls for a specific cell tower can be shown from the matrix view. The creation of new views enables comparison both in time and space to verify hypotheses. Finally, data for a combination of point (or period) in time, range of cell towers, and, range of regions, can be exported to a file for further investigation in external tools such as SynerScope's Marcato [5]. Also, facilities for easily searching the internet for events on a specific date and region are built-in. On double clicking in the geospatial view the platform-specific default browser is opened with according constructed search strings, containing the date and region in French.

5.6 Use cases

In the following sections, typical use cases are presented that demonstrate the power of the visual analytics approach to the exploration and analysis of massive mobile data in the context of the D4D challenge. First some background knowledge on Ivory Coast is discussed to provide a context. This background knowledge is assembled based on United Nations (UN) reports [227, 228, 229, 230, 231]. Next we provide general findings and interesting correlations are extracted from complex patterns found, while browsing through the data using the prototype.

5.6.1 Background knowledge

On November 28, 2010 elections were held to choose a new president for Ivory Coast. There were two candidates to be chosen from, the current president Mr. Gbagbo (leader of the FPI party) and the opponent Mr. Ouattara (leader of the RHDP party). On December 2, 2010 the Independent Electoral Commission announced that Alassane Ouattara garnered 54.1 per cent of the votes while Laurent Gbagbo received only 45.9 per cent of the votes. That same day, the constitutional council declared the electoral results to be invalid, due to missing the deadline for announcing the provisional results. The next day, December 3, 2010 the constitutional council proclaimed the final results of the presidential elections. This time, however, Laurent Gbagbo received 51.45 per cent of the votes while Alassane Ouattara received 48.55 per cent of the votes. The results of the first announcement were later certified as the rightful outcome of the elections by the UNOCI [229]. However, Laurent Gbagbo did not step down. This started the post-electoral crisis, resulting in violent attacks, killing of civilians, rape, torture, displacements, and, inhumane and degrading treatment. While human rights abuses have been committed by both sides, most of the killings have been carried out by elements of the forces loyal to Mr. Gbagbo [229]. The situation continued to deteriorate until former President Gbagbo was apprehended on April 11, 2011 [227]. However, pro-Gbagbo militias, mercenaries and FDS (former army) elements continued fighting. Some 50 of those elements surrendered to FRCI (new army) on April 29, 2011, the rest fled towards the Liberian border area, where they continued to kill civilians and loot property in south-western Ivory Coast. Clashes between the FRCI and pro-Gbagbo militias and mercenaries continued to be reported there, as well as violence against civilians in the west and south-west [227]. On December 11, 2011 legislative elections were held in a generally calm and peaceful manner, however, the country is still struggling to recover from the devastating crisis [228]. Here our data starts. We are provided with CDR-data covering the period December 5th, 2011 until April 22nd, 2012. Because media is government controlled and many reports are made that journalists are oppressed and newspapers are banned [227, 228, 229, 231], we solely rely on UN reports and reports of the International Crisis Group [148, 149, 150] as our source of major events that occurred during the data-period. During this period, the situation remains particularly fragile in western Ivory Coast, where large numbers of weapons, armed elements, former combatants, militias and dozogs (traditional hunters), as well as competition over the control of resources are significant sources of insecurity [230].

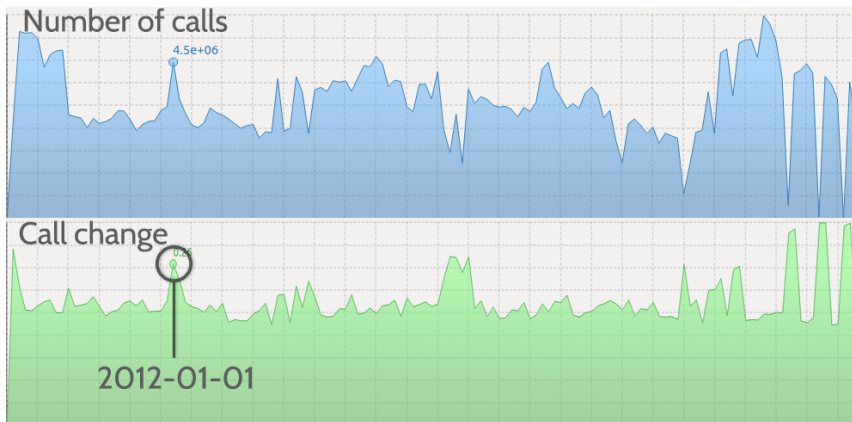


Figure 5.5: Peak in the number of calls and call change due to new year.

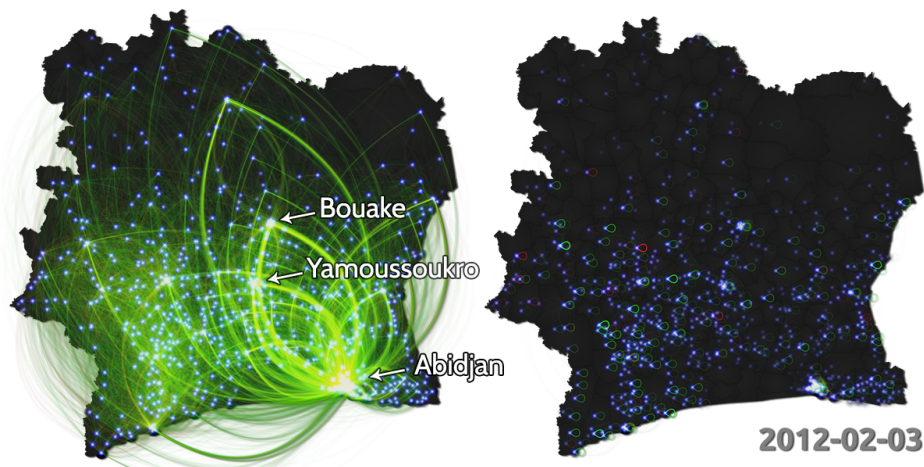


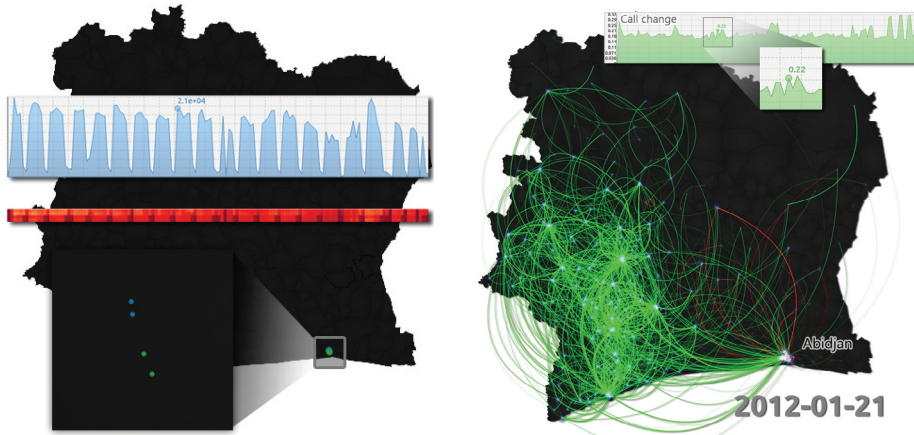
Figure 5.6: High-level communication patterns (left) and predominantly local communication (right).

Most of the incidents occurred in the west, although insecurity has increased in other parts of the country. Law enforcement, while present throughout the country, remains ineffective, and some areas are still under the protection of the *dozos*, which increases insecurity [230].

5.6.2 General findings

In the measure overview events that generate a peak in the number of calls and also in call change behavior are directly visible, such as the celebration of new year (see Figure 5.5).

From the inspection of the highest contributors to the number of calls on any day it becomes clear that the highest number of calls is very local. This appears in the



(a) Cluster of towers having a strong week-weekend pattern, located in Abidjan. (b) FPI meeting in Abidjan disrupted by supporters of the RHDP.

Figure 5.7: General events and local increase identification.

geospatial view as many predominantly self-loops (see Figure 5.6(right)). By plotting all contributors using a high transparency value for the individual edges, higher level communication is revealed (see Figure 5.6(left)). We see for example, that there is a strong link of communication between Bouake and Abidjan, but significantly less strong between Bouake and Yamoussoukro.

From the clusters in the matrix view, some clusters can directly be explained. For example, some clusters show a high week-weekend pattern, with less traffic in the weekends. These towers are based in Abidjan, more specific in the Plateau and Adjame region where most companies are located, giving less traffic in the weekends (see Figure 5.7(a)).

5.6.3 Local event increased call correlation patterns

During the following events there is a local increase of cell phone traffic. There is a clear correlation of call change and events that are directly visible when exploring the data. Below, these correlations are discussed in chronological order.

On January 21, 2012, there is a meeting of the FPI (pro-Gbagbo) in Abidjan. This meeting is violently disrupted by supporters of the RHDP (pro-Ouattara). One person was killed, several were injured and property was damaged. Also, national police officers were assaulted. On this day we see, Figure 5.7(b), that there is an heavy increase in telephone calls in the west (pro-Gbagbo), also noticeable is the increase in traffic from Abidjan to the western region (probably supporters calling their friends and family, informing them of the disruption).

On February 11, 12 and 13, 2012 clashes between communities are reported in Arrah. During these days (especially 12 and 13) there is indeed a local increase in the number

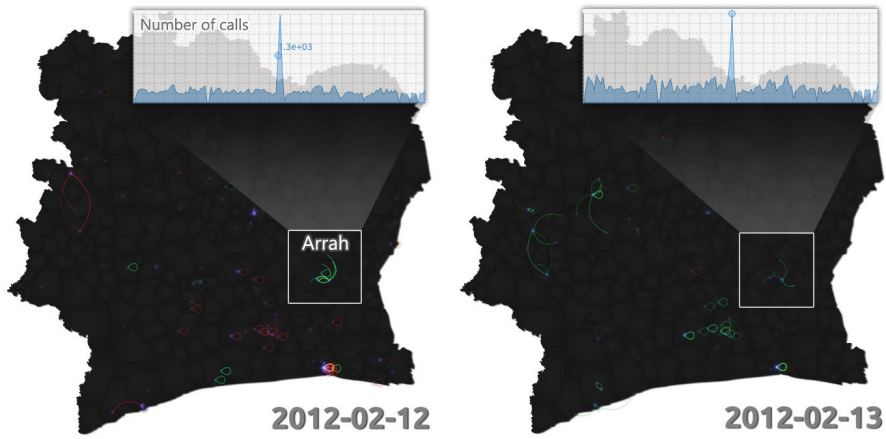


Figure 5.8: Clashes between communities in Arrah.

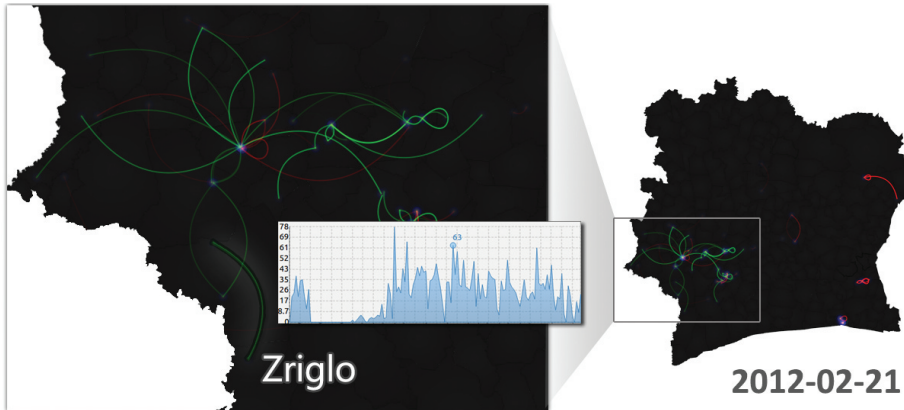


Figure 5.9: Attack on the village of Zriglo.

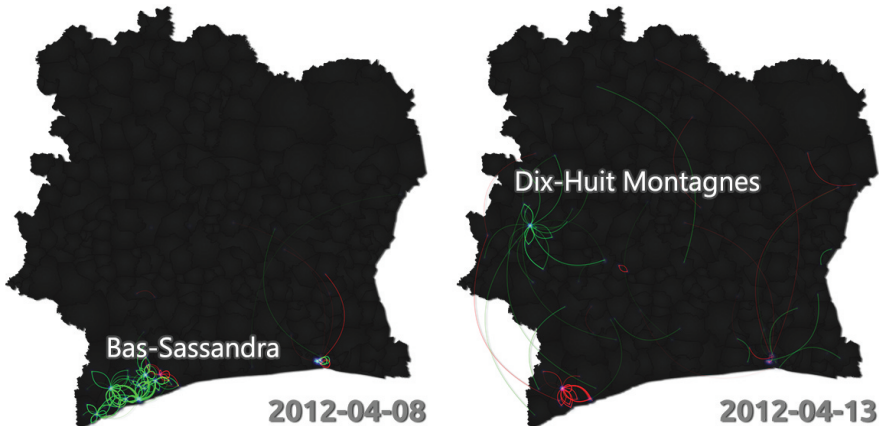


Figure 5.10: Increased phone calls correlated with rainfall anomalies.

of calls to the Arrah region, directly visible in the geospatial view (see Figure 5.8). If we bring up detailed information on the number of calls from the specific communication channels (antenna-to-antenna) there is indeed a remarkably high spike at these days, confirming something is going on.

On February 21, 2012, the village of Zriglo is attacked, killing six persons and wounding many more. There is indeed an increase in the number of calls to this village for this day. This also becomes apparent if the individual antenna-to-antenna communication is inspected, showing an unusual peak (see Figure 5.9).

Cocoa is a key commodity in Ivory Coast. The country is the world's largest producer of cocoa beans, accounting in 2010/2011 for a total 35% of the world's total production [147]. Because of this high economic value, cocoa has already been a driving factor for conflict in the country [60, 300].

Cocoa trees in Ivory Coast are harvested twice a year: a main harvest happens between September to December, while a second minor harvest happens from April to June [36]. April/May is a particularly important time for cocoa farmers, as two important events other than the aforementioned harvest happen: (a) it is one of the two times when pesticides are applied on cocoa plants; (b) if precipitation has been abundant farmers can establish new cocoa plantations or expand existing ones (starting field operations in May). Moreover, yam varieties growing in forest areas are planted in April and May [36]. Two main hypotheses can be thus made:

- Events happening in April/May (*i.e.*, harvesting, marketing and input supply) are likely to produce an increase in telephone traffic from the western cocoa-growing regions towards urban areas and other (market, logistics) hubs.
- Correlation with abundant (*i.e.*, above normal) rainfall in March/April is likely to produce additional increase in telephone traffic from the western cocoa-growing regions towards agricultural supply hubs.

Two positive rainfall anomalies that may have impacted agricultural activities and may be linked with increased phone calls have been identified in two regions: Bas-Sassandra and Dix-Huits Montagnes (see Figure 5.10). In the Bas-Sassandra region a positive rainfall anomaly during the first decade of April was reported (see Figure 5.11), and an increase in telephone calls on April 8, 2012, was noticed in two different areas: (a) the area of Sassandra and San Pedro subprefectures; and (b) the area of Tabou, Grand-Bereby and San Pedro subprefectures.

Higher data granularity was available only for the area of Sassandra. Rainfall was absent during the first decade of April, with the exception of a highly anomalous storm on April 3 reported in Sassandra [6]. Still this event does not explain the increase of telephone calls on 8 April. It is important to highlight that April 8 2012 matched with Easter. Some sort of correlation with religious events may hence be assumed.

In the Dix-Huits Montagnes region a positive rainfall anomaly during the second decade of April was reported (see Figure 5.12), and an increase in telephone calls on April 13 and 16, 2012. Particularly the increase in phone calls was localized in the Man subprefecture, which is an important center for cocoa production, and also is the most important production area of coffee in the whole country.

5.6. USE CASES

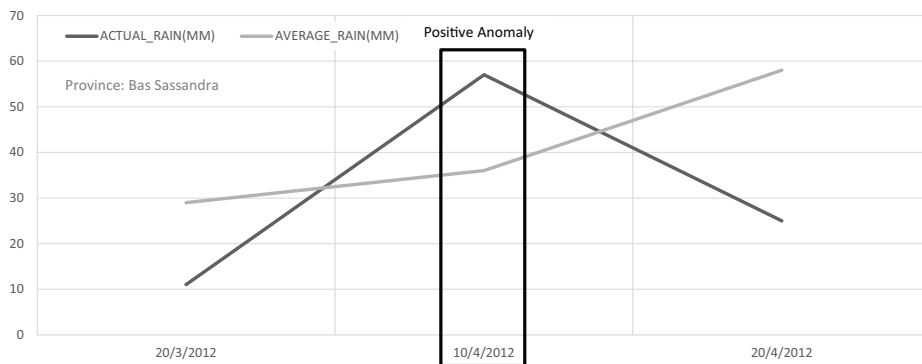


Figure 5.11: Rainfall activities in Bas-Sassandra region between 20 March and 20 April 2012 (10-day cumulated estimates) [4].

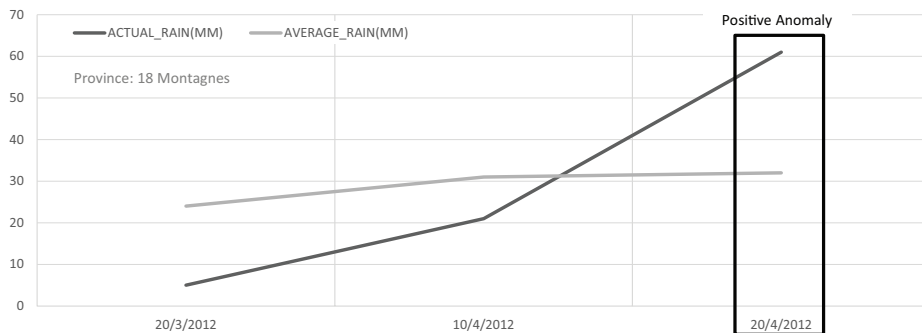


Figure 5.12: Rainfall activities in Dix-Huit Montagnes region between 20 March and 20 April 2012 (10-day cumulated estimates) [4].

5.6.4 Local event decreased call correlation patterns

In the events described below we found a strong local decrease of call activity (majority of complete shutdown of call activity). Again, these local decreases were clearly visible in the visualizations while browsing the data. Again, these events are discussed in a chronological fashion.

On December 15th, 2011 all cell towers in the western region appear to be shut down (see Figure 5.13(a)). A large number of antennas is connected to the electric grid of Ivory Coast. On this day half the country was shut down due to electric failure. Next, we identify additional towers by applying a hierarchical clustering on call behavior over time (see Figure 5.14). These cell towers have similar call behavior over time. If we inspect one of these towers (typically they all have this pattern), we see that these towers are not entirely shut down, but they remain to have an unusual low number of calls. Then, this low activity remains until the 4th of January, when they appear to be turned on again.

On January 15th, 2012, there were confrontations between communities in Gagnoa

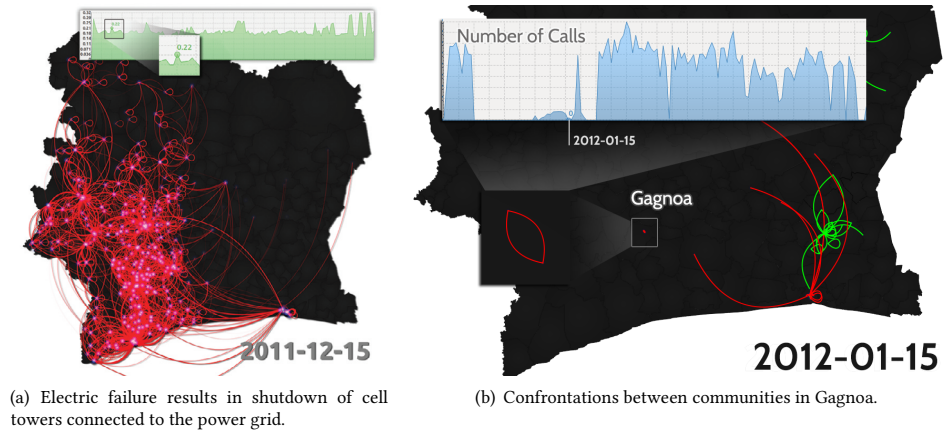


Figure 5.13: Local event decreased call correlation patterns.

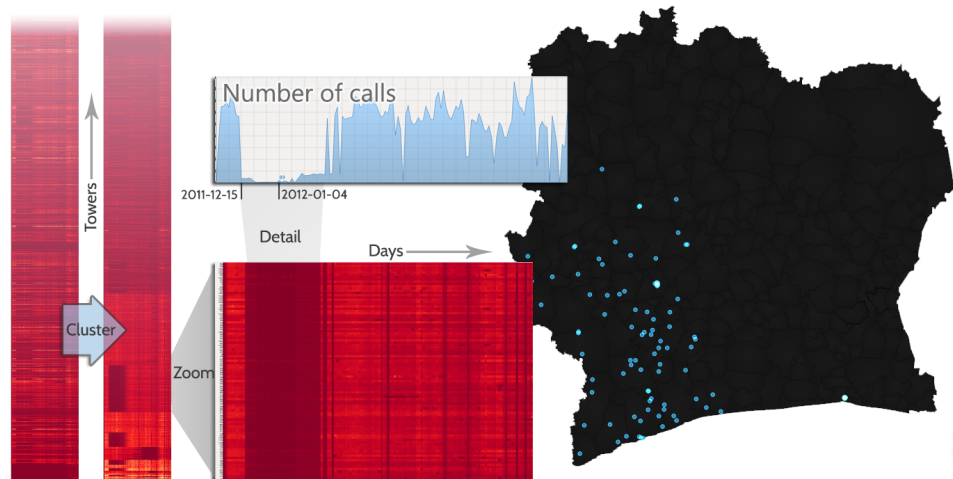


Figure 5.14: Cluster of towers in western region with unusual low activity for a significant period.

resulting in the deaths of 16 people, injuries to many more and the burning of several houses. On this day there is indeed a regional call change. The calls in this region drop to 0 (see Figure 5.13(b)). A possible cause is the fleeing of locals and damaging of the cell towers.

In early February, 2012 (no date mentioned) there were reports of confrontations between farmers and cattle breeders in Odienne. This led to injuries to several persons and the displacement of some 200 people. On February 5th, 2012 a shutdown of towers in the region around Odienne immediately show up in the call change graph (see Figure 5.15(a)). It should be underlined that farmer-pastoralists conflicts in the area of Odienne have been already reported by scholars [75], and the livelihoods of local farmers suffered additional significant stress in the recent two years. Particularly, in March

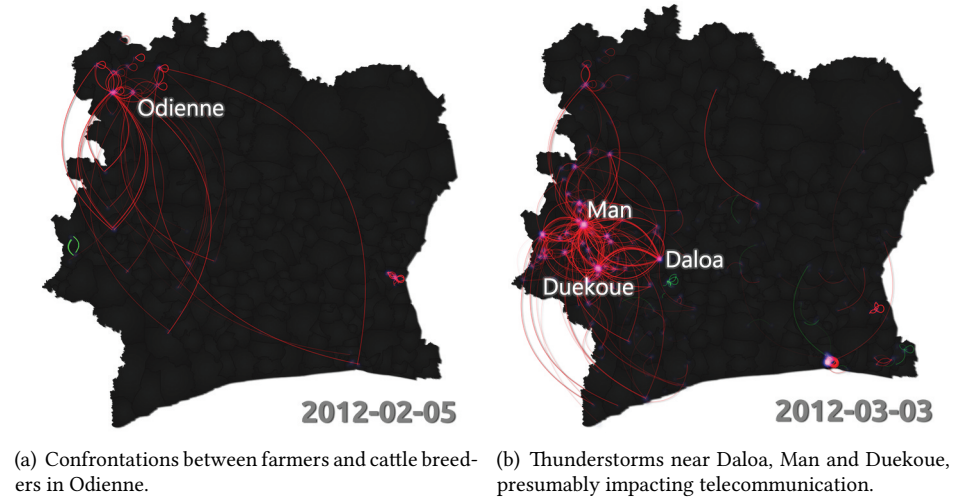


Figure 5.15: Local event decreased call correlation patterns.

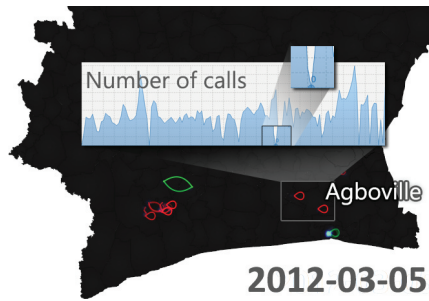


Figure 5.16: Bad weather conditions influencing local call activity around Agboville.

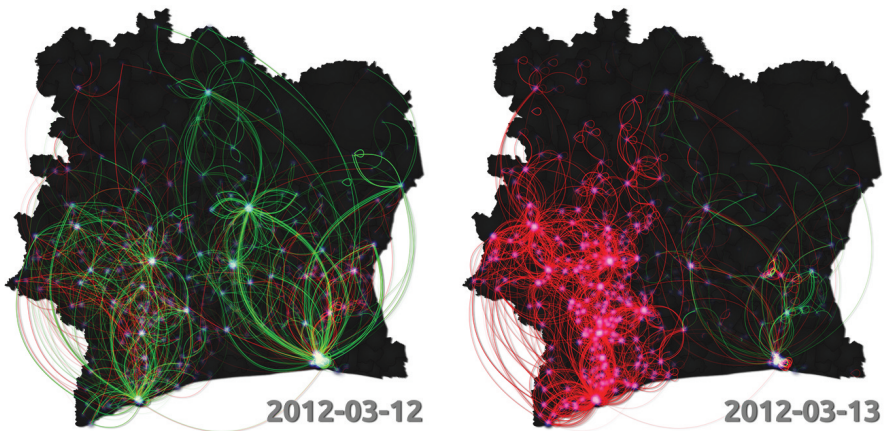


Figure 5.17: Day before (left) supposedly electric failure in the western region (right).

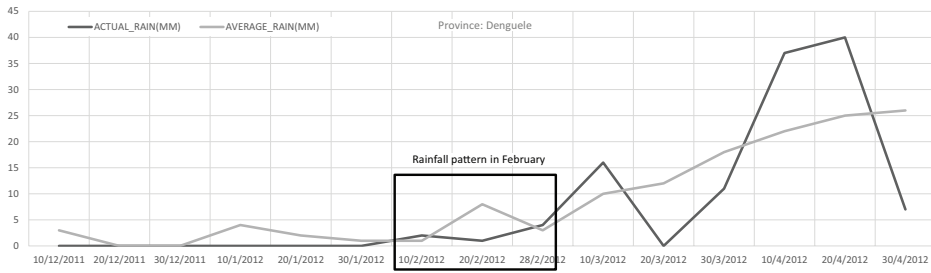


Figure 5.18: Highlight of rainfall pattern in Denguele region in February 2012. No rainfall activity was recorded on the whole region [4].

2012 FAO [2] classified Ivory Coast as a country in need of external assistance because of severe localized food insecurity, citing the northern regions as a food insecurity hotspot because of lacking support services and conflict-related damages to agricultural activities. We performed an assessment of disasters and weather for the time covered to test for correlation. However, no disasters were reported during the whole period according to the International Disaster Database [3]. Also, no significant weather events in terms of rainfall that may have disrupted telecommunications were recorded both in Gagnoa [6] and Odienne, based on Interpolated Estimated Dekadal Rainfall provided by NOAA/FEWSNet¹ [4] for the whole Denguele region (see Figure 5.18).

On March 3rd, 2012, near Daloa and neighboring big cities Man and Duekoue there was a drop in the number of calls. The number of calls on these communication lines dropped from the normal level of 100-200 to 10-20 on this day (see Figure 5.15(b)). On March 5th, 2012, in Agboville, we again see a decreased call activity (no calls) of at least two towers in the area, that directly pop-up in the call change behavior (see Figure 5.16). A thunderstorm was recorded on March 3rd, 2012 in Daloa [6]. This event may have seriously impacted telecommunications activity in the neighboring areas. The same weather conditions were reported in Abidjan on March 5, 2012 and we can fairly assume that Agboville (65 km distant from Abidjan) was affected as well by the thunderstorm.

On March 13th, 2012 cell towers in the western region are shut down (see Figure 5.17). This again might be the result of electric failure.

Finally, if we cluster towers on the number of calls over time, several more interesting clusters are revealed, all having a different shut-down period. To our opinion the shut down of towers can have a number of reasons: (a) this is missing data, Orange could or did not register calls, (b) external factors making communications more difficult, like weather and disasters, (c) sabotage on the antennas, (d) electric failure or diesel replenishment problems for off-grid cell towers, or (e) other technical problems.

The first explanation can be ruled out as it is stated in the D4D data information report that indeed there is missing data but this only covers a period of about 100 hours (± 4 days) [35]. However, we notice shut downs (or significantly lowered communication) for

¹Quantitative estimate of rainfall combining METEOSAT derived Cold Cloud Duration imagery and data on observed rainfall (GTS-Global Telecommunication System by the NOAA Climate Prediction Centre)

periods of more than 15 continuous days. As we have seen, weather conditions explain some of the local decreased cell tower activity. However, due to time constraints no explanation was found for the clusters of towers that have a significant period of lowered (or none at all) activity. These cases are currently being investigated in a collaborative effort with Orange.

5.6.5 Socio-economic development

By focusing on call change, local events can be detected as we have shown in the use cases. Both the local increase and decrease of call behavior provides insight in complex patterns. By focusing on clusters of cell towers having similar call behavior, events can be detected. These events are of different nature, such as weather (heavy rainfall in important cocoa area), social (new years eve), political (party meetings), disorder (clashes between communities) *et cetera* and can only be explained by domain experts by enriching the visual analysis process with external data to gain insight in complex correlations, anomalies and communities both in time and space. This in turn enables event detection which is an important first step towards prediction, improving early intervention through development, aid and other civil initiatives.

5

5.7 Conclusions

We aimed at developing a tool for the exploration and analysis of massive mobile data supporting all aspects of the process. We identified user tasks and requirements from which appropriate visualization, interaction and automated support techniques are selected. Next, we implemented these in a highly interactive prototype. We next showed the effectiveness of our visual analytics approach by applying the prototype on massive mobile phone data containing 2.5 billion calls and sms exchange between around 5 million users located in Ivory Coast over a period of 5 months, provided by France Telecom within the context of the Orange D4D challenge. From the typical use cases obtained while browsing the data, we extracted significant and interesting events by cross-correlating these using UN reports and weather information.

5.7.1 Future Work

In the context of the D4D challenge we mainly focused on the first dataset, containing detailed information of tower-to-tower communication, due to time constraints. It would be valuable to incorporate additional visualizations and automated techniques that enable also the exploration and analysis of the remaining datasets. Also, we believe exploration and analysis of non-aggregated data (*i.e.*, on person-to-person level) provides even more insight in complex patterns.



Massive Sequence Views

6

This chapter is based on [268, 269]:

“Reordering Massive Sequence Views: Enabling Temporal and Structural Analysis of Dynamic Networks.” S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. *In Proc. IEEE PacificVis*, pages 33–40, Feb 2013. **(Best Paper Award IEEE PacificVis 2013)**.

“Dynamic Network Visualization with Extended Massive Sequence Views.” S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. *IEEE Trans. Vis. Comput. Graphics*, 20(8):1087–1099, Aug 2014.

6.1 Dynamic Network Visualization with Extended MSVs

NETWORKS are present in many fields, such as finance, sociology, and transportation. Often these networks are dynamic: they have a structural as well as a temporal aspect. In addition to relations occurring over time, node information is frequently present such as hierarchical structure or time-series data. We present a technique that extends the *Massive Sequence View* (msv) for the analysis of temporal and structural aspects of dynamic networks. Using features in the data as well as Gestalt principles in the visualization such as closure, proximity, and similarity, we developed node reordering strategies for the msv to make these features stand out, optionally taking the hierarchical node structure into account. This enables users to find temporal properties such as trends, counter trends, periodicity, temporal shifts, and anomalies in the network as well as structural properties such as communities and stars. We introduce the *circular msv* that further reduces visual clutter. In addition, the (circular) msv is extended to also convey time-series data associated with the nodes. This enables users to analyze complex correlations between edge occurrence and node attribute changes. We show the effectiveness of the reordering methods on a synthetic and a rich real-world dynamic network dataset.

6.2 Introduction

Dynamic networks are present in many fields, such as finance, sociology and transportation: they have a temporal aspect such as time of transaction, time of connection, or time of packet transmission. Effective visual exploration of these networks is a difficult and as of yet unsolved problem, but is highly important to understand network behavior next to network metrics. Animation and small multiples are standard approaches to show the network behavior over time. There are obvious problems with animation, such as the difficulty to focus on many items simultaneously and the difficulty to track changes over (longer) time periods. For small multiples it is difficult to determine the number of multiples to use and to relate these to each other.

The Massive Sequence View, further referred to as msv is first introduced by Jerding and Stasko [158, 159] (as *Execution Mural*) and later extended by Holten *et al.* [71, 138]. The msv is a program execution-trace visualization technique that conveys both structural and temporal aspects and is an extension of the traditional Message Sequence Chart [152] in which time is explicitly mapped to the horizontal (or vertical) axis (see Figure 6.1(c)). Program classes are represented using (invisible) horizontal lines, positioned equally spaced along the vertical axis. The horizontal axis of the visualization represents chronological order $t_0 \dots t_n$. If there is a function call from class c_i to class c_j at time t_k , a vertical line with start- and end-points at the vertical positions of c_i and c_j , respectively, is drawn at horizontal position t_k . This is repeated for all function calls in the program execution-trace. By examining the trace, users can discover phases in the execution, relationships between classes, and, in general, how the objects accomplish the functional purpose of the program [158]. Phrased differently, the visualization enables the exploration of structural as well as temporal aspects. Furthermore, the msv,

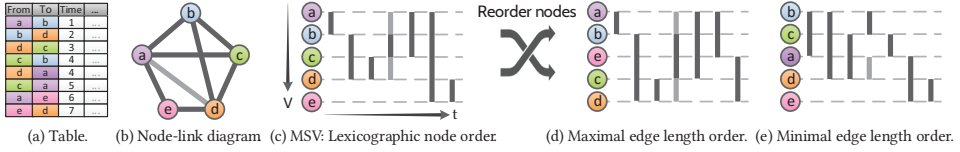


Figure 6.1: Different visualizations of dynamic network data. For the node-link diagram (b) time is flattened and not visible. The MSV (c) shows the relations over time. Note that there are no overlapping edges in the minimal edge length ordering (e).

extended with anti-aliasing techniques, filtering, and zooming, scales well with respect to both the number of classes and number of messages and is usually part of a multiple coordinated view system, such as in the work of Holten *et al.* [138] where one view conveys network topology and the MSV the dynamic aspects.

Due to the scalability and ability to explore both the structural and temporal properties of the program execution, the advantage is that it is also useful for general *dynamic network* exploration. Due to the chronological order of function calls there are no edge crossings in the traditional MSV. In a dynamic network there may be multiple edges at (approximately) the same point in time, implying overlapping edges in the MSV dependent on the order of the nodes (see Figures 6.1(d,e)). As a result a standard MSV suffers from visual clutter and important patterns can be hidden; classes (nodes) are shown in the same order as they are declared in the program header files [158]. However, classes can be listed in alphabetical order, by their appearance order in source files, or by user specification [159, 160]. Holten *et al.* [138] apply a user-defined hierarchy on the nodes representing high-level constructs such as modules, packages and classes. However, it is still unclear how to reorder the nodes such that certain features stand out. Also, in a program execution-trace all calls are significant and typically there is no noise. In general, dynamic networks are less structured compared to program execution traces. Not all edges are significant and therefore regarded as background noise. The more prominent, multiple occurring edges represent the typical behavior of the network. Therefore, reordering techniques are necessary to emphasize either typical or anomalous visual patterns. First, we determine what visual patterns are easy to identify using the MSV due to Gestalt principles and show how they relate to dynamic network data. Next, we present different reordering strategies to emphasize the visual patterns for easy identification.

In a first step we proposed reordering strategies on the nodes such that important visual patterns are emphasized, enabling better exploration of dynamic networks [268]. Ordering or sorting is an important and strong visualization-supporting technique already explored in different application domains such as parallel coordinate plots [183] and adjacency matrices [180, 301]. We believe the presented reordering techniques may be valuable and applicable not only to the MSV, but also to other visualization techniques such as node order in arc-diagrams [197] or one-dimensional graph layouts in general. In addition to the reordering strategies we further explore the design space of the MSV. More specific, we contribute extended methods to communicate optional node information, such as a hierarchical structure and time-series data. Furthermore, we introduce a *circular* MSV to further reduce visual clutter.

This chapter is organized as follows. First, related work is discussed in Section 6.3. We define dynamic networks and identify characteristic temporal and structural dynamic network patterns in Section 6.4. Different node reordering strategies to emphasize these features are presented in Section 6.5. The circular MSV is introduced in Section 6.6. Next, the model is extended to also take extra node information into account in Section 6.7. In Section 6.8, we apply the reordering techniques on real-world and synthetic datasets. Finally, limitations, conclusions, and directions for future work are given in Sections 6.9 and 6.10, respectively.

6.3 Related work

The two dominant approaches to dynamic network visualization are animation, *e.g.*, [106, 177, 214] and small multiples, *e.g.*, [93, 118, 216].

Animation has drawbacks, such as the difficulty to focus on many (moving) items simultaneously. Also, tracking changes over (longer) time periods demands high cognitive efforts. Therefore, if animation is used there needs to be a good layout stability to preserve the mental map, such as in the visual analytics approach for dynamic social networks [95] and the DGD system [207] utilizing the Foresighted Graph Layout with Tolerance [76]. For small multiples it is difficult to determine the number of multiples to use and to relate these to each other. Therefore, a better approach to dynamic network visualization is to provide a static overview of the entire time span of the network as is the case with the MSV.

Visualization techniques providing an overview of the entire time span include Timeline Trees [48], TimeSpiderTrees [51], TimeRadarTrees [50, 52], Parallel Edge Splatting [54], TimeEdgeTrees [55], TimeArcTrees [119], and Alluvial diagrams [218]. All of these techniques are also subject to a node ordering, however in these proposals there is no special order on the nodes. Although, in the Parallel Edge Splatting method which is closest to our technique, Burch *et al.* [54] note that more sophisticated sorting methods could optimize the visualization, but these are not implemented and a practical solution is not given. In the Parallel Edge Splatting technique time is discretized and the resulting graphs are placed next to each other on the horizontal axis similar to parallel coordinate plots. Nodes are connected by straight lines causing large amounts of edge crossings, which is tackled by applying edge splatting techniques. MSVs do not suffer from crossing edges, because the edges are drawn as vertical lines, there may be however overdraw due to edges occurring at the same time or when there are less pixels available horizontally for displaying each edge at a 1-pixel width. Therefore, the MSV is drawn using anti-aliasing techniques and grayscale shading to increase the perceived spatial resolution. We also use these anti-aliasing techniques to render the images.

The MSV is extended by enabling users to interactively control filtering and abstraction [161]. Brushing techniques are added and the color and size of function calls can be set, and classes can be selectively shown or hidden [159]. Eick and Ward [85] independently developed an interactive visualization for Message Sequence Charts with brushing, colors on the edges, and zooming. Holten *et al.* [71] used zooming to inspect

patterns on a fine-grained level. Furthermore, the msv guarantees the visibility of outlier calls when visualizing many calls using Importance-Based Anti-Aliasing [138]. Our implementation also includes zooming, filtering, and coloring capabilities.

De Pauw *et al.* [74] introduce a variation on the msv called execution patterns which is more true to the original Message Sequence Chart [152]. Renieris and Reiss [215] address the aspect ratio problem; if there are many calls and relatively few classes, the execution patterns and msv tend to get very wide and thin, which is solved by mapping time to 2D instead of 1D by laying out the visualization as a spiral.

6.4 Definitions and features

In this Section, we first give the dynamic network definition used throughout this chapter. Next, different Gestalt principles applicable to the msv are explored. As is shown using these principles, several visual patterns are easy to recognize and interpret during the analysis process.

6.4.1 Dynamic network definition

We consider a dynamic network as a directed graph $G = (V, E)$, with a node (vertex) set V and edge set $E \subseteq V \times V \times T$ with vertex tuples (v_a, v_b) and time attribute $e_t \in [t_{min} \dots t_{max}]$ for each edge (also referred to as a transaction). Transactions are considered to occur instantly; they have no duration. Furthermore, time-series data can be associated with nodes; nodes can have attributes that change over time or are static over the entire time span such as hierarchical structure information.

We define a *configuration* π_V to be a permutation on the set of vertices V , representing the vertical order of the nodes in the msv. Furthermore, i gives the index of vertex v_i in the current configuration π_V . Consider two vertices v_p and v_q ; we define an *edge set* E of vertex pair (v_p, v_q) as:

$$E(p, q) = \{e = (e_a, e_b, e_t) \in E \mid (e_a = v_p \wedge e_b = v_q) \vee (e_b = v_p \wedge e_a = v_q)\}, \quad (6.1)$$

where e_a and e_b give the source and sink vertex of edge e , respectively, and $e_t \in [t_{min} \dots t_{max}]$, i.e., the set of all edges over time between two nodes.

6.4.2 Gestalt principles

The Gestalt principles stem from the cognitive psychology field. These principles help to explain that there are easy to identify visual patterns that stand out in the msv. Consider the simple situation in which there is a burst of transactions between two nodes within a short amount of time. Such a burst is perceived as a block in the msv.



Figure 6.2: Gestalt principles applied to msv.

We are also interested in more complex features and how they show up in an msv. We identify these patterns and present different strategies to emphasize them and leverage the identification of the visual patterns in the visualization. Below, the relevant Gestalt principles *closure*, *proximity*, and *similarity* are discussed in the context of the msv.

Many closely positioned subsequent transactions are perceived as a solid block due to the **closure** principle (see Figure 6.2(a)). *We tend to perceptually close up, or complete, objects that are not, in fact, complete* [252]. The closure principle fills the space between subsequent edges as if there is none. To make this effect work the msv edges need to be sufficiently close, leveraged by the Gestalt principle *proximity*, which makes the individual edges appear as a single block. Objects that are close are perceived as a group due to the **proximity** principle (see Figure 6.2(b)). *When we perceive an assortment of objects, we tend to see objects that are close to each other as forming a group* [252]. In the msv the horizontal temporal dimension is used to make the edges appear as blocks. Groups of similar blocks are perceived to find periodicity, shifts, and anomalies. *We tend to group objects on the basis of their similarity* [252]. Blocks of similar shape can be identified due to the **similarity** principle (see Figure 6.2(c)).

From this point on we use the word *block* for a group of transactions adhering to the Gestalt principles closure and proximity; they are sufficiently close to be perceived as a single group, *i.e.*, the time between subsequent edges is close to zero given the current scale.

Two types of features exist in dynamic network data that define the network behavior: *temporal* and *structural* features. The msv enables users to explore both temporal and structural features of the network simultaneously. Below, different characteristic features, inspired by [55], are defined and discussed in the context of the msv.

6.4.3 Temporal properties

A **trend** is an increase or decrease in the number of transactions over time between two or more nodes, visualized in the msv by sequential blocks that have an increasing or decreasing width (see Figure 6.3(a)). A **counter trend** deviates from trend patterns by showing mirrored behavior. If the global trend is to have an increase in the number of transactions for a group of nodes, then a possible counter trend is the decrease of the number of transactions for a different (smaller) group of nodes (see Figure 6.3(b)). **Periodicity** is the periodic repetition of a transaction or burst of transactions among two or more nodes, apparent in the msv as equally spaced sequential blocks of similar shape (see Figure 6.3(c)). A **shift** is a sudden or gradual disruption in a periodic repetition of a transaction or burst of transactions among two or more nodes (see Figure 6.3(d)).

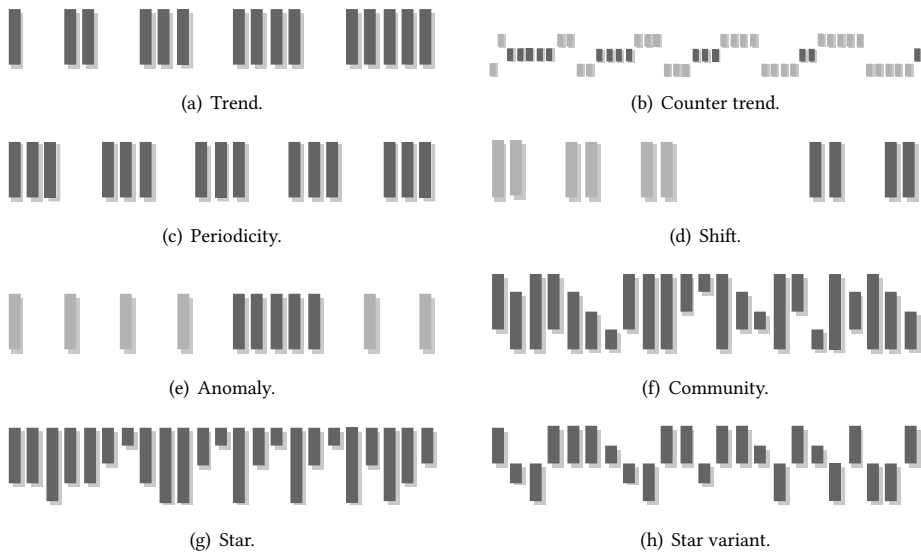


Figure 6.3: Temporal and structural properties of a dynamic network in the context of the MSV.

An **anomaly** is a temporal pattern that behaves differently compared to the mainstream temporal patterns, recognized as a sudden block between homogeneous behavior (see Figure 6.3(e)).

6.4.4 Structural properties

A **community** is a group of nodes that has a high number of internal transactions (between the nodes of the group) and a relatively low number of external transactions (from or to a node not belonging to the group). A community is a block-like structure consisting of multiple nodes (see Figure 6.3(f)).

A **star** is a group of nodes that all have transactions with one node. A star can be *incoming* (one node only receives), *outgoing* (one node only sends) or a *mix* of both. Star patterns are apparent in the MSV by an imaginary horizontal line with blocks of varying width and direction attached to it (see Figures 6.3(g,h)).

In order to make the temporal and structural features stand out there are a number of parameters available that we can manipulate. The first two are the width and the color of the edges. These can be varied to emphasize features of interest. Below, color is briefly discussed and it is explained why this feature alone is insufficient to emphasize blocks in the MSV. The remainder of the chapter therefore focuses on geometry related optimizations.

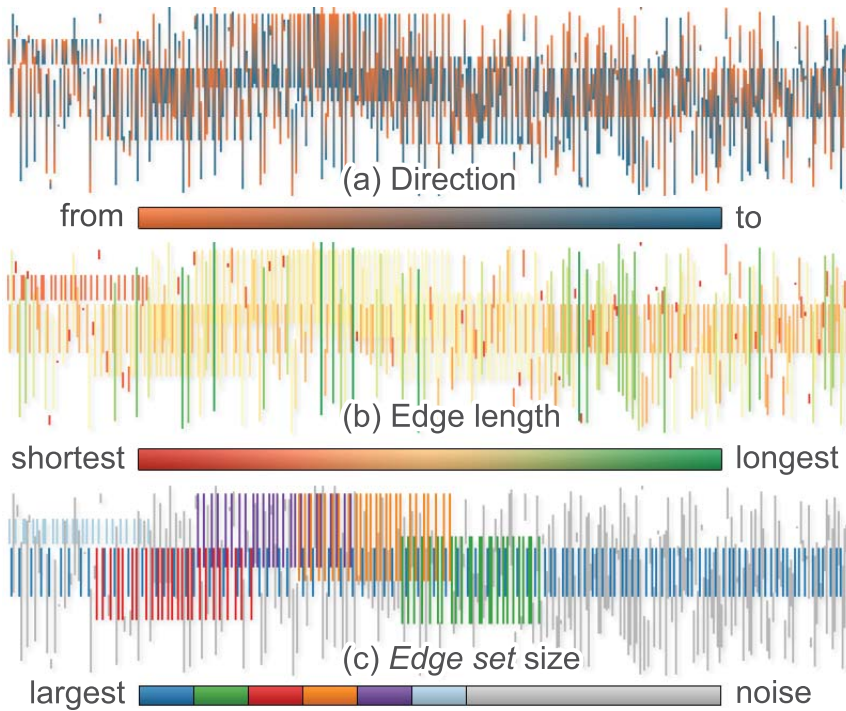


Figure 6.4: Different color encodings for the MSV edges.

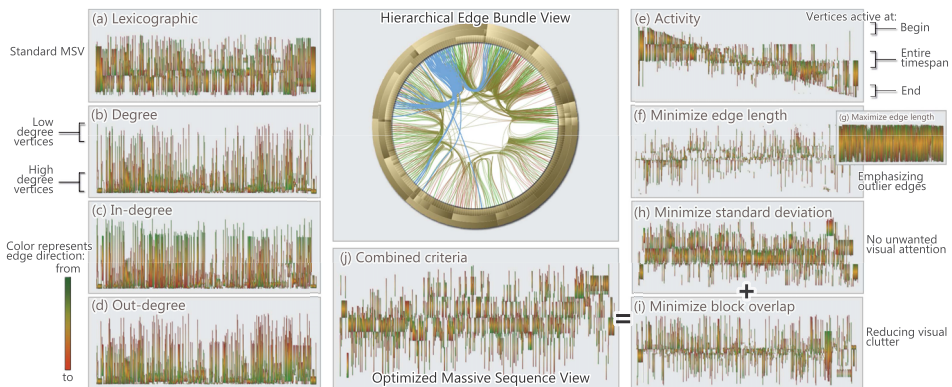


Figure 6.5: Massive Sequence Views as part of a multiple coordinated view application applied to real-world dynamic network data; a financial transaction dataset containing 302 accounts and 997 transactions. Each ordering provides a unique view on temporal and structural aspects of the data.

6.4.5 Color

As suggested by Holten *et al.* [138], color can be used to encode edge direction (see Figure 6.4(a)). Furthermore, color can be used to convey edge attributes such as (some) weight to reveal patterns in the data. However, to make *blocks* more apparent in the msv the spatial Gestalt principle of similarity can be enhanced by using color. We propose to color the edges based on edge length, since edges belonging to the same block all have the same length (see Figure 6.4(b)).

Also, the frequency of edges between node-pairs can be used to color the different blocks. If one, for example, is interested in only the most apparent feature, we can color the edges of the largest edge sets (see Figure 6.4(c)). The number of blocks (features) to be colored can be adjusted by the user to reveal more patterns. To leverage the perception of the colored blocks, the residual edges are given a dimmed color to prevent unwanted attraction. However, color only is not sufficient, because blocks can become very small or overlap each other heavily, making it difficult to interpret the visualization.

To make the temporal and structural features stand out in the msv we use the vertical order of the nodes. Inspired by matrix reordering techniques [189], we explore and present different reordering strategies for the msv in Section 6.5 using the Gestalt principles as a guideline to enhance user perception with respect to visual pattern recognition.

6.5 Reordering techniques

In the following, we present strategies to select and compute a node configuration such that different features stand out, simplifying the exploration of dynamic network behavior. A simple (naive) approach to reveal temporal and structural patterns in the dynamic network is to order the nodes based on a structural property, such as degree. Also, one can aim to optimize a visual edge property, such as the length distribution of the edges by reordering the nodes, revealing other patterns. However, a better choice is to take time into account to be able to prevent overlapping blocks / features, making them easy to identify and interpret. To achieve good solutions, visual edge properties can be combined with temporal block overlap prevention. Below we present different reordering strategies including their implementation, advantages, and disadvantages.

6.5.1 Structural properties

For comparison purposes, Figure 6.5(a) shows a standard msv with a configuration based on the lexicographic order of the node names. The degree of nodes is often an important attribute; Figure 6.5(b) shows the effect of sorting nodes for this. It shows what nodes have a high degree and, furthermore, the distribution of degree over time per node. It will, for example, reveal nodes that have a high degree at specific points in time but overall have a low degree; these nodes will be at the top of the msv and will clearly stand

out. To reveal more structure in the high- and low-degree nodes, the degree property can be split into in- and out-degree based configurations as shown in Figures 6.5(c,d).

To explicitly investigate temporal behavior, nodes can be ordered based on their temporal activity. This is achieved by taking the average x -position $\bar{x}(v)$ (representing time) of the edges containing node v :

$$\bar{x}(v) = \frac{1}{|E_v|} \sum_{e \in E_v} e_t, \quad (6.2)$$

with E_v the set of edges containing v : $\{e \in E \mid e_a = v \vee e_b = v \wedge e_t \in [t_{min} \dots t_{max}]\}$. Next, the nodes are ordered such that $\bar{x}(v_i) \leq \bar{x}(v_j)$, $\forall i, j$. This configuration, Figure 6.5(e), reveals at what point or period in time nodes in the network are most active. The nodes that are mostly active at the start are positioned near the top, nodes that are mostly active at the end are positioned near the bottom. Nodes that are active throughout the entire time span are positioned around the center. This sorting method gives an easy to interpret, natural visual flow of time that is rendered from top left to bottom right.

6.5.2 Visual edge properties

The length of the edges in the msV influences the visual attention that they get. In general, importance is linked to edge length implicitly [116], as larger edges attract more attention compared to shorter edges. The computation of edge length is as follows:

$$l(v_a, v_b) = \|a - b\| / (|V| - 1), \quad (6.3)$$

where a, b give the index of nodes v_a and v_b in the current configuration π_V (see Section 6.4). A configuration can be computed on the nodes such that the average length of the edges is minimized. Unfortunately, this minimization problem is an NP-hard combinatorial optimization problem, known as the *optimal linear arrangement problem* [113].

To be able to (interactively) approximate the minimal edge length we utilize *simulated annealing* [61, 172], because it generally gives good solutions for optimization problems. Furthermore, it can deal with arbitrary systems and cost functions, providing us with a generic framework to approximate optimal solutions for other strategies defined later on. Finally, it is straightforward to implement. We choose for simulated annealing over other optimization methods due to its flexibility and its statistical guarantee of global convergence to an optimal point. Also, simulated annealing does not put any restrictions on the properties of our model. We are aware that other methods can be used here, such as constraint programming, mixed integer programming, and, genetic algorithms. For the reasons above we have chosen simulated annealing, and consider use of other methods as future work. Further details of the used simulated annealing method with according parameter initializations are described in Section 6.5.5.

As cost function for the simulated annealing process we use the average edge length of the configuration:

$$\text{minimize: } \bar{l}(\pi_V) = \frac{1}{|E|} \sum_{e \in E} l(e). \quad (6.4)$$

By minimizing the edge lengths of the msv, edges that have a relative high appearance frequency, and are therefore considered as typical for the current network, are given a shorter length (see Figure 6.5(f)).

This reduces visual clutter and improves the overall readability of the visualization by reducing the cognitive load. Also, outlier edges are emphasized, due to increased edge length. However, using this method visibility of the background level of activity serving as a comparison baseline is compacted into narrow, difficult to perceive activity bands that provide an analyst with little clue as to what normal behavior looks like. With this optimization we prevent the generation of a cluttered, high-ink-usage visualization that mainly shows background activity while paying no attention to the emphasis of structural or temporal outliers.

When the edge length is maximized (see Figure 6.5(g)), it reveals nodes that are most dominant in the network, which are now positioned near the top and the bottom of the msv. Inversely, shorter edges are more likely to be the outlier edges in this configuration. However, maximization of edge length does not give useful results in general, as the display area becomes filled and features are hard to discern.

If general behavior is more important, and not outlier edges, we do not want (larger) edges to receive unwanted attention. To prevent the longer edges from dominating the visualization, the edges should have about the same length overall. This is achieved by using the standard deviation of the edge lengths as cost function to the simulated annealing procedure. The minimization problem here is:

$$\text{minimize: } \text{stdev}(\pi_V) = \sqrt{\frac{1}{|E|-1} \sum_{e \in E} (l(e) - \bar{l}(\pi_V))^2}. \quad (6.5)$$

Minimization of edge length standard deviation provides a nice and balanced alternative as far as providing an uncluttered visualization with reduced ink-usage is concerned, while furthermore providing users with a uniform, non-distracting presentation (see Figure 6.5(h)). In such a visualization, highly communicative sub-communities stand out as clearly visible activity bands, while outliers occupy sparsely populated bands.

Both minimization of edge length and standard deviation improve the readability and simplify the analysis process. However, they do not take the *overlap* of blocks into account, making it difficult to identify and interpret different possibly overlapping features.

6.5.3 Time involvement to prevent block overlap

To improve the readability of the visualization and the interpretation of the dynamic network behavior, blocks should be easy to identify. To achieve this and make features stand out from the noise by leveraging the Gestalt principles, the *overlap* of blocks should be minimized. To this end, a measure is introduced to determine the overlap of blocks for a configuration and is used as cost function for the simulated annealing procedure for minimization.

Suppose there is a burst of edges $E(a, b)$ between nodes v_a and v_b and also a burst of edges $E(c, d)$ between nodes v_c and v_d , assuming $a < b$ and $c < d$. This gives two possibly overlapping blocks (see Figure 6.6(left)). For the simple case, we can state that $\text{overlap} \approx \text{vertical overlap} \times \text{horizontal overlap}$, here the coverage C is $(b-c) \times (t_3 - t_2)$. This generalizes to:

$$C(a, b, c, d) = C_v(a, b, c, d)C_h(a, b, c, d), \quad (6.6)$$

where the vertical overlap

$$C_v(a, b, c, d) = \max \left(\left(\min(b, d) + \frac{p}{2} \right) - \left(\max(a, c) - \frac{p}{2} \right), 0 \right), \quad (6.7)$$

with p a user-configurable padding parameter $0 \leq p \leq 100$. With each edge set $E(a, b)$ we associate a density profile $f(a, b, t)$ using a Kernel Density Estimation approach:

$$f(a, b, t) = \sum_{e \in E(a, b)} h(t - e_t), \quad (6.8)$$

where $h(t)$ is a kernel and $t \in [t_{\min} \dots t_{\max}]$. Here we use a Gaussian for $h(t)$:

$$\text{gauss}_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}. \quad (6.9)$$

The horizontal overlap is computed by taking the integral of products of all edge-pair kernel density estimations:

$$C_h(a, b, c, d) = \int_{t_{\min}}^{t_{\max}} f(a, b, t) f(c, d, t) dt. \quad (6.10)$$

The total score for overlap is now:

$$C(\pi_V) = \sum_{(i, j, k, l) \in V} C_v(i, j, k, l) C_h(i, j, k, l), \quad (6.11)$$

where the indices i, j, k and l are chosen such that $|E(i, j)| > N$ and $|E(k, l)| > N$ to discard sparse edgesets, which are in general not of interest, and also to speed up the calculations. Furthermore, it must hold that $i \neq j$ and $k \neq l$ to discard self loops and finally $\neg((i = k) \wedge (j = l))$ to prevent taking self overlap into account. Note that

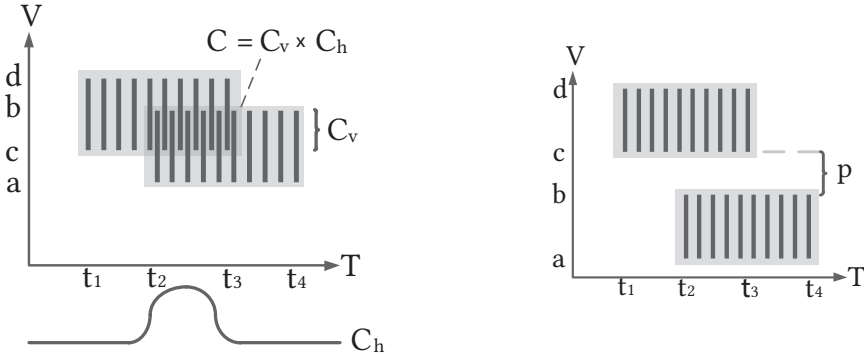


Figure 6.6: Blocks $E(a, b)$ and $E(c, d)$ with vertical overlap C_v , horizontal overlap C_h , total overlap C , and padding parameter p .

C_h can be computed and stored independently from the current configuration to speed up the final coverage computation by using a lookup table. Next, the average coverage C of all node-pairs is taken as the cost function for the simulated annealing process to minimize the overlap of blocks (see Figure 6.6(right)).

The choice for σ in the convolution process of function $f(a, b, t)$ (see Formula 6.9) is related to what is considered a block in the data. In general, $5 \leq \sigma \leq 10$ gives good results in practice. If the edges of a block in the data are sparse, then a large σ should be used to consider the edges as blocks; if blocks in the data are dense, then a small σ is appropriate to detect them. Alternatively, kernel size can be set by users based on domain knowledge. If, for example, the data is known to contain sparse blocks, or if there is an interest in finding sparse blocks, then a large σ should be used. Furthermore, the parameter may be based on known blocks in the data, for example weeks or hours. Based on this domain knowledge σ can be chosen to make these blocks stand out in the visualization. One route to minimizing block overlap is the reduction of the height of blocks, which makes it hard to see the individual blocks (see Figure 6.5(i)). This can partially be solved by increasing the padding parameter of the vertical overlap function C_v . The padding parameter, $0 \leq p \leq 100$, prevents blocks to be stacked close to each other, making them appear as a single feature. Overall, setting p to 10 is sufficient to prevent stacking of feature blocks.

Typical real-world data with a temporal component is often characterized by a stable or slowly changing contextual (background) level of activity serving as an important indicator of normal behavior. Against the background of this normal “data rhythm” structural and temporal outlier patterns should be depicted as clearly and preattentively as possible without compromising the overall visibility of the background rhythm that serves as an important and immediate comparison baseline. When exploring, e.g., financial transaction data, an investigator or auditor is often interested in the difference between the presence of highly communicative sub-communities, i.e., groups of bank accounts; in this case, block overlap minimization provides an ideal mechanism to focus an analyst’s attention on different and separate groups of accounts. However, in addition to minimizing the block overlap, blocks should have a height as large as possible to stand out from the noise for easy identification and interpretation.

6.5.4 Combining methods

Both methods, minimizing block-overlap and minimizing standard-deviation, have their advantages and disadvantages. A well-chosen combination of measures benefits the advantages of both and is therefore preferred: visible non-overlapping blocks with no unwanted visual attention to specific edges, leveraged by the Gestalt principles for easy interpretation. Unfortunately, both methods cannot be combined in a simple manner, due to the difficulty of normalizing both measures. If we want to combine the two metrics they should be normalized, otherwise one metric will weigh much stronger. For the normalization of the metric we need the maximum value to divide each value by. However, we cannot easily compute the maximum values, because this is the inverse of the minimization problem. Fortunately, a set of optimal configurations, more specifically, the Pareto optimal solution set, can be acquired using the multi-criteria ϵ -constraint optimization method [126]. Users are given a trade-off between the two optimization criteria by taking different ϵ and swapping the sequence of the optimization functions (see Figures 6.7 and 6.8). We presented interaction techniques to effectively explore visualizations depending on multiple parameters, such as these, using a small multiple approach [273] (see Chapter 3). The multi-criteria ϵ -constraint optimization is implemented as follows. First, the constraint criterion is optimized in isolation; next, the other criterion is optimized, with the restriction that the first optimization value is no worse than an ϵ -fraction of the first solution.

A combination of both block overlap minimization and edge length standard deviation minimization provides analysts with a visualization that allows them to 1) perceive the background rhythm as an immediate comparison baseline, 2) discern highly communicative sub-communities as well as temporal and structural outliers within uniformly sized activity bands that are densely or sparsely populated, respectively, and 3) perceive all of this within a uniform, non-distracting layout that depicts temporal events with significantly reduced visual clutter and ink usage (see Figure 6.5(j)).

6.5.5 Simulated annealing

Simulated annealing is used for cost function optimization of the presented reordering strategies. The outcome of the simulated annealing process depends on several parameters such as the initial temperature and cooling schedule that computes a new temperature each iteration [258]. During each iteration a new configuration is computed; a random node is positioned at a random new position. The new configuration is accepted if the cost function value for that configuration is better than the previous configuration cost value, in accordance with the Metropolis-Hastings algorithm [128, 193]. If the cost value of the configuration is worse, it is accepted with a probability equal to the Boltzmann factor of the cost function difference. For the cooling function we use a geometric cooling schedule to have a fast cool-down, and, hence lower running times. See Section 6.9.3 for a discussion on computational scalability. We ran different experiments on all three measures: edge-length, standard deviation, and block overlap, to determine default values for temperature and cool down function. The experiments were run for varying initial temperatures and cool down functions in order to select the ones that gave good results within a short amount of

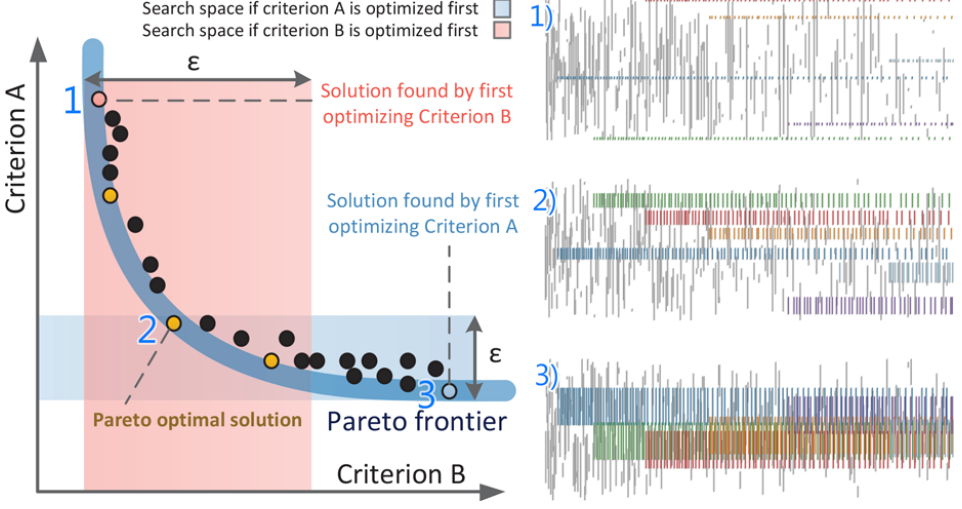


Figure 6.7: Pareto frontier with search space for both optimization criteria A and B, showing influence of sequence of optimization criteria and constraint parameter ϵ in the multi-criteria optimization process along with some example solutions for minimizing block overlap (1), standard deviation (3) and combined criteria (2).

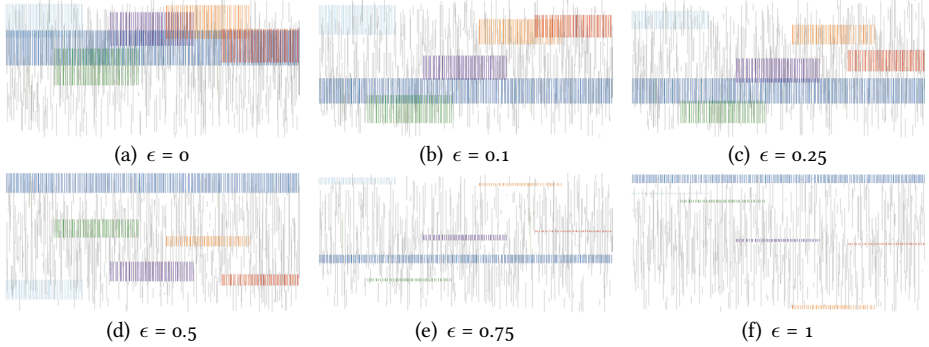


Figure 6.8: Edge length standard deviation μ minimized, next block overlap is minimized with constraint to stay within an $1 + \epsilon$ ratio of μ . Edges colored according to edge set size.

time, revealing the trade-off between number of iterations and solution quality. For the experiments we define the cooling schedule as $T_{i+1} = cT_i$, $i > 0$, where c is some constant $0.98 \leq c \leq 0.999$. The initial temperature T_0 is varied with values $0.1 \leq T_0 \leq 1000$ of stepsizes $T_0 * 10$ and c with stepsizes 0.005, i.e., we tested five different T_0 and c values: $\{0.1, 1, 10, 100, 1000\}$ and $\{0.98, 0.985, 0.99, 0.995, 0.999\}$ respectively. For each of the 25 combinations of T_0 and c we ran the simulated annealing algorithm 1000 times and took the average of the solutions. We found that an initial temperature of 100 (maximum distance between two solutions) and a constant c of 0.999 provide good solutions. Therefore, these are set as default parameters. The number of iterations (tried configurations) with this setup is 18411. Depending on the search space of the dataset (the number of nodes) these default values should be adjusted.

6.6 Circular Massive Sequence Views

The geometry of the msv can be further optimized by using a polar representation for the edges. After the reordering process there might be edges that span from the top to the bottom of the msv. By allowing edges to wrap around, visual clutter can be reduced even further. One way to achieve this is to use a three-dimensional cylinder and project edges to this instead of the two dimensional plane projection used right now. However, additional interaction techniques have to be introduced to rotate the cylinder along its length axis. Additionally, a large part of the edges would not be visible due to the orientation. A better solution is to draw circular edges using a polar representation (see Figure 6.9(a)). Here, time is encoded by the radius of the circle. Edges appearing at the beginning are drawn near the center of the circle, whereas edges occurring at the end are rendered at the outer boundary of the circle. This introduces a bias with respect to attention given to the edges, because over time edges are drawn using an increasing amount of ink. However, we are free to emphasize transactions occurring at the beginning or the end of the time span by reversing the time direction. Using the circular representation an edge can always be drawn in two ways, using either the longest or shortest path between two nodes. The edge length computation $l(v_a, v_b)$ (see Formula 6.3) between two nodes v_a and v_b is adapted accordingly:

$$l'(v_a, v_b) = \frac{\min(\|a - b\|, |V| - \|a - b\|)}{|V| - 1}. \quad (6.12)$$

By rendering edges always using the shortest path in addition to the previously introduced reordering strategies on the nodes, visual clutter is reduced even further. For dynamic networks the number of edges (over time) is typically much larger than the number of nodes. Therefore, the standard msv tends to get very elongated, but the circular msv reduces this effect. Both the standard msv and circular msv have their advantages and disadvantages (see Figure 6.9(b)). Communication between two nodes is more apparent in the circular variant, however, more screen space is needed. For the standard msv bursts in time are better discernible. Also, comparisons of time moments is harder using the circular msv. This could be improved by overlaying time grid lines, however this would introduce visual clutter. If the standard and circular msv are linked via standard brushing and linking techniques, advantages of both can be utilized to gain maximal understanding of the dynamic network behavior.

6.7 Extending the model

Dynamic networks, or networks in general, usually contain more information than just nodes and edges. In this section we explore two of these aspects, node hierarchy information and node time-series information, and how these can be incorporated in the node reordering strategies for the (circular) msv, both computationally as well as visually.

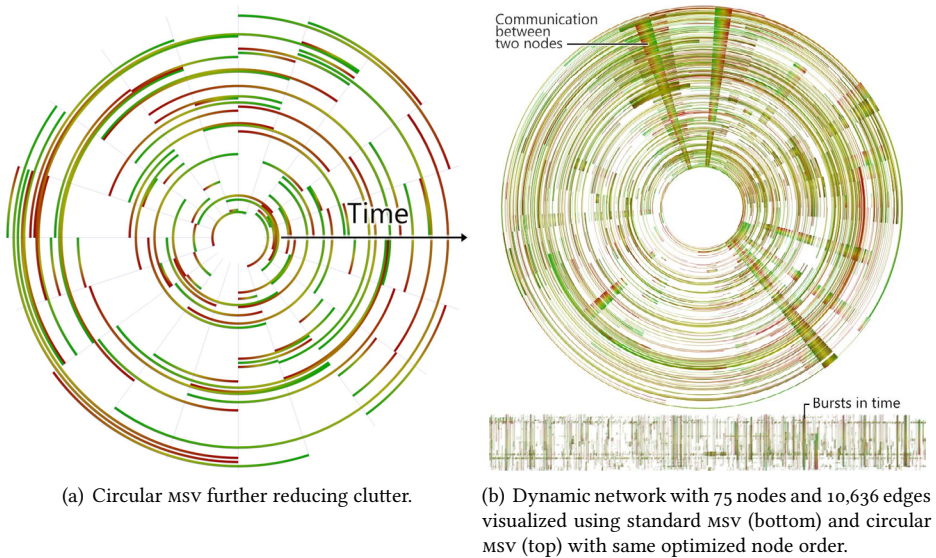


Figure 6.9: Dynamic networks visualized using circular MSV.

6.7.1 Node hierarchy

Often a natural node hierarchy is present, or generally can be deduced, which imposes structure on the nodes [138]. For example, organization structure in case of e-mail communication or geographical bank location form a natural hierarchy in case of a financial transaction network. By constraining the position of the nodes in the MSV to adhere to their position in the hierarchy, anomalous transactions can be detected and analyzed. Also, transactions are generally expected to adhere to certain (domain) rules or implicit agreements imposed by the hierarchy. For example, people in an organization mainly communicate to other people that are on a similar level of the hierarchy, implying natural communities in the data. By applying these hierarchical constraints in the node reordering process the anomalous unexpected communication patterns stand out. In addition, the hierarchy can support collapsing and expanding behavior [138] which improves visual scalability of the MSV. We propose three reordering strategies, in a similar fashion as Holten and Van Wijk [139], for hierarchy-constrained node reordering:

- *top-down*, first apply a reordering strategy on the highest level, then recursively apply reordering strategies on lower levels of the hierarchy;
- *bottom-up*, start with reordering strategies on the lowest levels of the hierarchy, then apply reordering strategies on higher levels of the hierarchy;
- *iterative combination*, iteratively apply top-down and bottom-up strategies until the solution does not improve significantly anymore according to a user-defined convergence factor.

We found that generally the iterative application of the top-down and bottom-up strategies gives the best results. However, this is computationally (slightly) more intensive. In general, after about 3 to 5 iterations the solution does not improve significantly anymore. The convergence factor nicely provides a tradeoff between solution quality and computation time. Figure 6.13 shows an example result for the iterative hierarchy-constrained reordering technique.

6.7.2 Node time-series data

In addition to (static) hierarchy node information, time-series data on the nodes is also frequently available. Inspired by CircleView [169] we extended the (circular) msV to additionally show time-series information on the nodes (see Figure 6.10). This enables users to discover correlations between node attributes that change over time and the occurrence of edges involving these nodes. Similar to CircleView we use a heat map to depict time-series values for an attribute over time. In contrast, we do not use a slice of the circle but instead use a single line. This is done to leave space for the edges and be able to analyze both aspects simultaneously. However, showing both pieces of information simultaneously introduces clutter. This can be solved by providing users with controls for the opacity of both the edges and node time-series rendering. This enables users to shift the emphasis towards the information of interest for analysis, and still provides the option to see both at the same time, revealing complex correlation patterns (see Figure 6.10).

6.8 Use case

We apply the reordering strategy to a real-world dynamic network dataset showing the effect on the msV. This dataset also contains node hierarchy information and we apply the constraint reordering technique to reveal unexpected communication patterns. Finally, the dataset contains time-series data on each of the nodes. We combine all techniques and show how these can be used to visualize and analyze dynamic networks in order to reveal complex patterns and correlations.

6.8.1 Dataset

For the evaluation of all techniques we used the rich Social Evolution Dataset from the Reality Commons project made available by the MIT Human Dynamics Lab [188]. The dataset contains, among other information, mobile phone calls between 75 residents of a dormitory over a period of 8 months. The dataset also contains a static, one-level hierarchy; for each resident we know the advancement within their studies, *i.e.*, freshmen, sophomore, junior, senior, graduate or unknown. Furthermore, we have time-series information on the reported health status (flu symptoms) of the residents. The dataset contains 75 nodes and 10,636 edges.

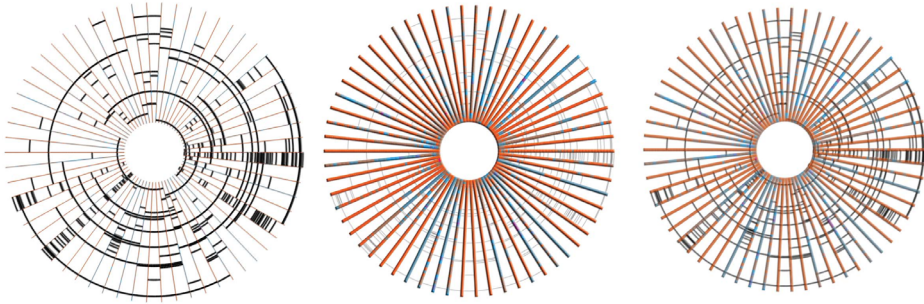
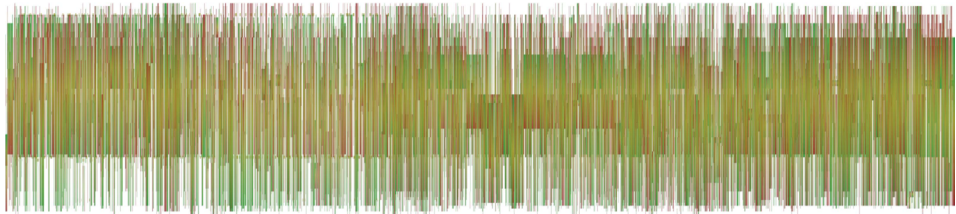
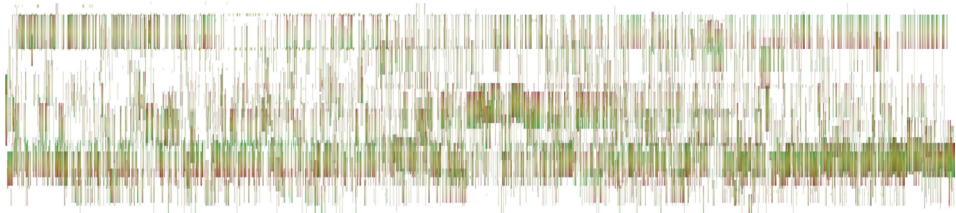


Figure 6.10: Time-series data associated with nodes visualized using heatmap approach. Emphasis on edges (left), node time-series data (middle), and simultaneous display for complex correlation discovery and analysis (right).



(a) Non-optimized msv.



(b) Reordered msv.

Figure 6.11: Standard msv (a) and combined reordering strategy applied to msv (b).

6.8.2 Reordering strategy

First we apply the combined reordering strategy, minimizing standard deviation and block overlap, to the msv (see Figure 6.11). From the reordered msv we observe several highly communicating communities (Figure 6.11(b), bottom), bursts of communication at points in time (at start and halfway), communication between two nodes (top), and observe that communication increases over time.

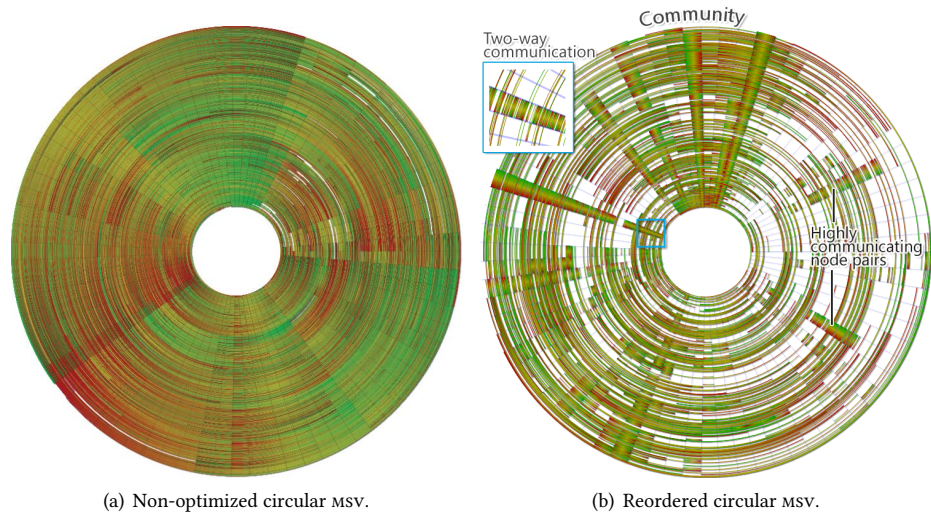


Figure 6.12: Circular MSV (a) and minimizing edge length node reordering strategy applied to circular MSV (b).



Figure 6.13: Standard hierarchy-constrained MSV, with hierarchy applied according to appearance in data file (a) and combined reordering strategy applied to hierarchy-constrained MSV (b).

6.8.3 Circular MSV

By applying the edge length minimization strategy on the circular MSV we gain more insight in the community. A number of pairs of nodes that are communicating heavily over time stand out (see Figure 6.12(b), top). Furthermore, we observe a highly communicating node pair that is communicating sparsely with other nodes (left). The communication at the beginning of the time span is mainly two-way communication, after which there is a phase of no communication and then the communication seems to be unidirectional for the rest of the dataset. Finally, we spot several other heavily communicating nodes mainly active halfway the time span (right).

6.8.4 Hierarchy

To gain more insight in the highly communicating channels we group the nodes by applying a hierarchical structure on the geometry (see Figure 6.13). Now nodes are constrained during the reordering process. The non-optimized hierarchy reveals some structure, however, the bigger insights come from the reordered MSV. Here we see that the Freshman and Junior students are the least communicating groups. We also observe that the Graduate students mainly communicate with other Graduate students, sparsely with Senior students, but not with others. There are some communication bursts in which students from the Unknown group communicate with students from the Senior group. These observations are not directly apparent in the unordered hierarchy-constrained MSV (see Figure 6.13(a)) due to visual clutter or the reordered MSV without hierarchy (see Figure 6.11(b)).

6.8.5 Node time-series

By visualizing time-series associated with the nodes we can observe correlations between communication behavior and health status (see Figure 6.14). We see for example that there is a burst of communication from one student when that student becomes ill (bottom inlay box). We also observe two students that are heavily communicating with each other over time except for one period in which one of the students reports not being healthy (top inlay box).

6.9 Limitations and Workarounds

Limitations with respect to visual scalability, network topology and computational scalability are briefly discussed and possible solutions are suggested for each category.

6.9.1 Visual scalability

The MSV scales well visually thanks to anti-aliasing techniques and blending [138]. When the number of nodes is large and visual scalability becomes an issue, filtering

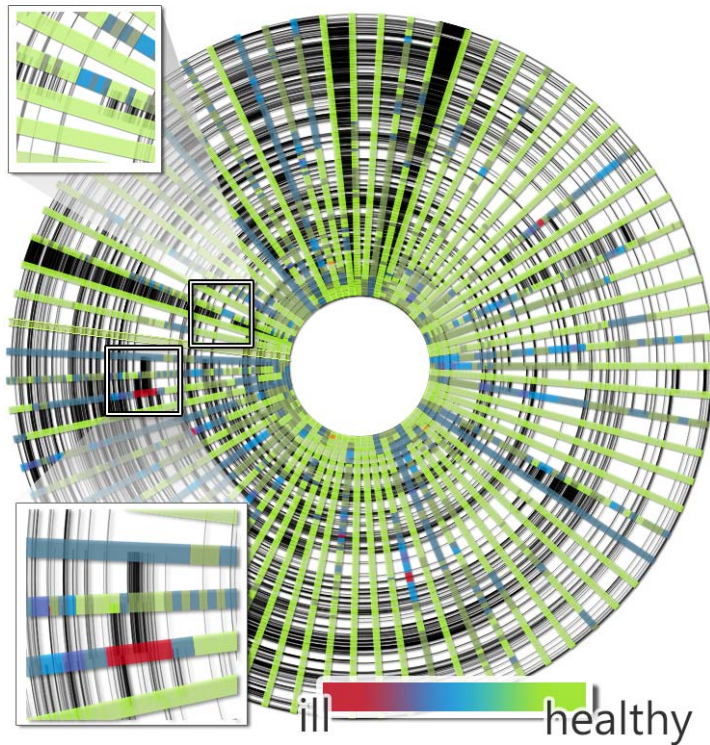


Figure 6.14: Time-series data associated with each node. The visualization reveals correlations between edge occurrence (calls) and attributes (flu symptoms). Edges are rendered black to increase contrast and prevent color interference hence, simplifies the analysis.

techniques can be used to only show subsets. Furthermore, clustering techniques can be applied to create new nodes, one for each cluster, and only show these with aggregated edges, or solely show one cluster to inspect internal communication.

If visual scalability reaches its limit due to the number of edges, despite the blending and anti-aliasing techniques, zooming or period selection can be employed. After zooming, the nodes can be reordered again to have an optimal ordering for analysis of the zoomed selection. However, this breaks mental map preservation due to the change in ordering between zoomed selections. Similar techniques, such as splitting time in smaller periods, can be employed if reordering does not provide enough insight due to the network having a high number of conflicting edges.

6.9.2 Network topology

For networks that contain many overlapping edges at the same point in time, reordering techniques will not help to render each of the individual edges without overlap due to a high conflict rate. However, to have a sense of the number of edges occurring at

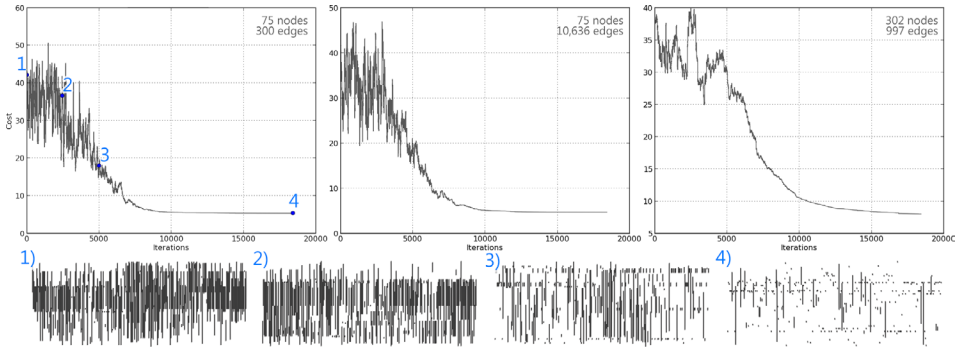


Figure 6.15: Cost (average edge length $\in [0..100\%]$) versus number of iterations in the simulated annealing process for various size graphs. The quality of the visualization at different iteration points are shown for the first plot.

the same time, we provide users with a control to interpolate the horizontal time (x -axis) position between real-time position and chronological order. For the computation of the chronological ordering of edges with the same time, different attributes can be used. We group edges with similar source and sink nodes to emphasize the block effect. Furthermore, the interpolation can also be employed to interpolate between real-time position and a different time-axis position, based on an edge attribute for further analysis.

Also, interaction techniques such as linking and brushing can be used to highlight the edges in a different view, for instance a node-link diagram or edge bundle view to better convey the network topology.

6.9.3 Computational scalability

The simulated annealing process offers a trade-off between time of simulation and solution quality. We found that good quality solutions are found in about 5 to 20 seconds by using the parameters as described in Van den Elzen *et al.* [268] for moderate size graphs, $|V| \leq 1,000$, $|E| \leq 10,000$. Figure 6.15 shows charts of average edge length (percentage of total height) versus number of iterations in the simulated annealing process. The charts show that the process converges here to a global minimum and does not get stuck in a local minimum. These charts are representative for all discussed cost functions. Furthermore, convergence speed not only depends on the structure of the graph, more specifically the connectedness, but also on the number of nodes, due to more options, rather than the number of edges.

To speed up computations, motivated by a high number of nodes or in case of time-critical applications, only important nodes can be considered in the optimization process based on an attribute or metric such as degree or betweenness. Furthermore, parallel versions of the simulated annealing algorithm can be used [211].

6.10 Conclusions

In this chapter the msv is utilized to visualize the dynamics of networks. Different characteristic temporal and structural patterns are defined in terms of the msv and according reordering strategies are proposed, making the patterns stand out, such that they are easy to identify and interpret. We present three main reordering strategies, based on node structural properties, visual edge properties, and prevention of block overlap in time. Furthermore, we present a practical solution to combine the different reordering techniques approximating the optimal solution, using simulated annealing and ϵ -constraint multicriteria optimization, providing good solutions within a short amount of time. We believe the reordering techniques are powerful techniques and are valuable to other visualization techniques, in particular 1D network layouts such as arc-diagrams.

The effectiveness of the different reordering strategies is shown by applying them on generated and real-world datasets. From these use cases it becomes apparent that each proposed individual reordering strategy is useful and worth inspecting as well as the combined reordering strategies. The new node orders improve readability, reduce cognitive load, and bring forward temporal and structural features present in the data, leveraged by Gestalt principles for easy identification and effective exploration and analysis.

The standard msv is extended to enable the analysis of optional node data. To further reduce visual clutter and use space more efficiently we introduce the circular msv. Time-series information associated with the nodes can be analyzed using a hybrid approach of the circular msv and a representation based on CircleView [169]. We investigated how hierarchical node structure can be taken into account and present a solution that integrates this information into the reordering strategies using an iterative top-down and bottom-up process.

The msv does not reveal detailed topology of a network very well. Hence, the msv should be used as part of a coordinated view application with at least one other view conveying network topology, or as part of a tool-chain. In that case, the techniques presented in this chapter are a useful component towards enabling temporal analysis and understanding of dynamic networks by using the presented reordering strategies for the msv.

6.10.1 Future work

There are various directions for future work. The temporal and structural feature list is not complete and could be extended to, for example, cliques, cycles, motifs, and paths in the network. A different direction is to extend the msv itself to allow for a better topological representation of the dynamic network. Furthermore, the metrics as defined here, especially the overlap metric C might be generalized to apply to 2D network layouts as well. This can then in turn be used to develop 2D dynamic network layouts.

The optimization process might be adapted to improve running times using operations research methods such as constraint programming or mixed integer programming. A challenge for the algorithms or graph drawing community is to find approximation algorithms with polynomial running times that produce solutions with guaranteed bounds.

Finally, the presented techniques should be evaluated using user studies to strengthen the claims.



Reducing Snapshots to Points

This chapter is based on [270]:

“Reducing Snapshots to Points: A Visual Analytics Approach to Dynamic Network Exploration.” S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. *IEEE Trans. Vis. Comput. Graphics*, xx:xxxx-xxxx, 2015. *to appear* (**Best Paper Award IEEE VAST 2015**).

7.1 A Visual Analytics Approach to Dynamic Network Exploration

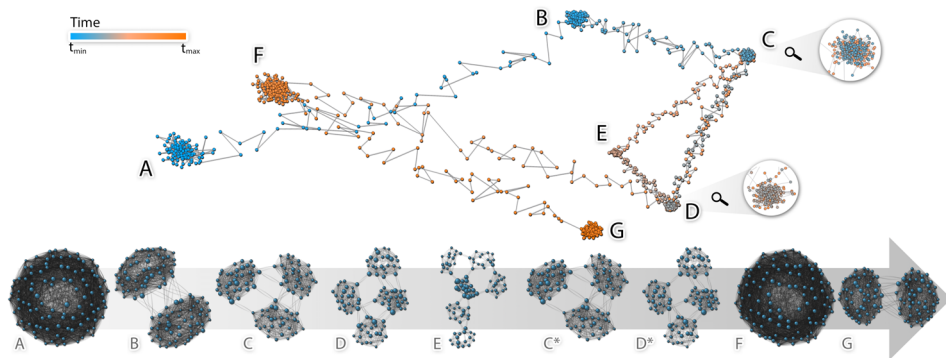


Figure 7.1: Reducing snapshots of the dynamic network to points enables the exploration of network evolution. The projection of snapshots (top) reveals that the network contains 7 stable states (A-G), with 2 recurring states (C,D), and shows the transitions between them. A representative network for the snapshots in each stable state is shown (bottom).

W^E propose a visual analytics approach for the exploration and analysis of dynamic networks. We consider snapshots of the network as points in high-dimensional space and project these to two dimensions for visualization and interaction using two juxtaposed views: one for showing a snapshot and one for showing the evolution of the network. With this approach users are enabled to detect stable states, recurring states, outlier topologies, and gain knowledge about the transitions between states and the network evolution in general. The components of our approach are *discretization*, *vectorization and normalization*, *dimensionality reduction*, and *visualization and interaction*, which are discussed in detail. The effectiveness of the approach is shown by applying it to artificial and real-world dynamic networks.

7

7.2 Introduction

Networks are ubiquitous, as they describe relations between objects. Often these networks are large and dynamic; they change over time. Some examples of dynamic networks are (tele-)communication networks, social networks, financial networks, and transportation networks. Understanding the evolution of dynamic networks is a challenge. Typical insights to be gained are the discovery of states in dynamic networks that characterize the network over time. The identification of stable states, recurring states, outlier states, and transitions between these states helps in understanding the network. For example, the network can change gradually from one state to another, it could alternate between multiple states, or it might not be stable at all. An approach for the identification of these states and obtaining insight in the evolution of the network in general is needed.

The challenge is twofold; how to visualize, interact with, and analyze a large static network for one point in time (*snapshot*), and second, how to visualize, explore and analyze many of these snapshots. For the analysis of a static network (one snapshot) many methods are available, such as node-link diagrams and visual adjacency matrices [180]. In this chapter we address the second issue, and present an approach to explore and analyze many snapshots of a dynamic network. Currently, there are two major visual approaches to analyze network evolution: mapping *time-to-time* and mapping *time-to-space* [27]. Two implementations of these techniques are *animation* and *small multiples*. For small multiples one problem is to find an optimal balance between using few images, lacking temporal detail, and many smaller images, which are hard to interpret. Furthermore, it is demanding to focus on many items simultaneously and relate these to each other. The use of animation also has problems, such as the difficulty to track changes over (longer) time periods, and how to visually encode these changes.

Here we present a novel method that enables users to trace the network over time: the reduction of snapshots to points. More specifically, we contribute a visual analytics approach for dynamic network exploration, enabling:

- the visual identification of stable states, recurring states, and outlier states;
- the transitions between states, and;
- the analysis of network evolution in general.

The approach consists of four steps: 1) discretization; 2) vectorization and normalization; 3) dimensionality reduction; and 4) visualization and interaction. These steps are discussed in detail, and guidelines and defaults for parameters are provided. Together these steps enable dynamic network exploration.

The chapter is organized as follows. First, related work is discussed in Section 7.3. The visual analytics approach is presented in Section 7.4. In Section 7.5 we apply our approach to artificial and real-world dynamic networks. The approach is discussed in Section 7.6 and finally, conclusions and directions for future work are given in Section 7.7.

7.3 Related Work

For static networks many visualization methods are available such as node-link diagrams, visual adjacency matrices, and hierarchical edge bundles. The two dominant methods for dynamic networks, using these techniques, are animation and small multiples [30, 265].

In animation the different instances of the network at each timestep are shown as a movie, e.g., [106, 177, 214]. This approach leads to a high cognitive load, as users need to focus on many moving or changing items simultaneously. Also, tracking (multiple) changes over time is difficult. To reduce this burden, animation is often combined with

a timeline control, for example in [20, 220]. If a node-link diagram is used as network visualization in the animation, then the stability of the network layout for subsequent timesteps influences the viewers mental map. Therefore, mental map preservation is an important area of research in graph drawing [76, 95, 207].

In the small multiples technique the different timesteps are presented as juxtaposed visualizations using a filmstrip or grid layout [93, 118, 216]. In general, it is difficult to determine the number of multiples to use in the visualization. Furthermore, the multiples might be far apart from each other, making it harder to relate them to each other for the discovery of patterns. Also, visualization space for each multiple becomes smaller as the number of multiples grows. This decreases the readability of the individual visualizations. If small multiples are used interactively, a solution to this is the use of large singles for detailed inspection [273].

A variation on the small multiples approach is superposition; stacking the multiple instances of the network and optionally connecting related nodes of sequential timesteps [22, 83, 120].

Next to the generic techniques of animation and small multiples, sophisticated visualizations that provide an overview of the entire timespan of the network have been proposed [48, 50, 51, 52, 54, 55, 119, 218, 269]. However, these specialized visualizations are often difficult to interpret, difficult to reproduce, and generally pose restrictions on the network type.

Visual analysis techniques have been proposed for the exploration of dynamic networks, e.g., Von Landesberger *et al.* [286] also employ dimensionality reduction techniques with a focus on contagion in networks. Hadlak *et al.* [125] cluster temporal attributes associated with the nodes and edges. Steiger *et al.* [250] extend upon this by enabling the analysis of repeating time-series patterns. In contrast to our method, the structural connection between nodes and edges is not considered in both methods. Moreover, for our method we do not need temporal attributes to enable exploration and analysis.

Also, visual analytics solutions for the exploration of dynamic networks are proposed, e.g., [9, 21, 92]. Falkowski *et al.* detect and show communities over time that are connected based on similarity. This method, in contrast to ours, focuses on community detection and not on states in the dynamic network. Outlier states as well as the absence of communities will not be picked up by the community detection. Furthermore, it is difficult to relate combinations of communities to identify recurring or stable states. Other methods, not related to dynamic networks, also propose visual analytics solutions for the identification of recurring states, e.g., [14, 281].

We only briefly discuss related work here, as dynamic network literature is plenty. For further reading we refer to recent dynamic network visualization surveys providing a broader perspective [17, 27, 170, 287]. In summary, current methods are either based on mapping time-to-time such as animation, or mapping time-to-space with (variations of) small multiples. Both techniques have their own issues and shortcomings. In this chapter we explore our simple yet powerful method that deviates from these traditional techniques, by reducing snapshots of the dynamic network to points for exploration and analysis.

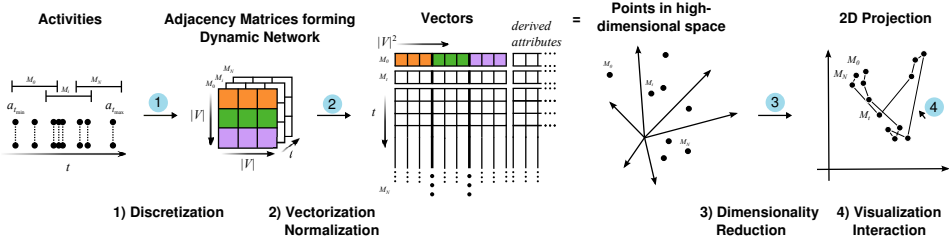


Figure 7.2: Visual analytics approach for the exploration of dynamic networks with different steps 1) discretization, 2) vectorization and normalization, 3) dimensionality reduction, and 4) visualization and interaction.

7.4 Reducing Snapshots to Points

First, the global concept is introduced by providing an overview of the approach. Next, dynamic networks are defined, followed by a description of the four steps of the approach: discretization (Section 7.4.3), vectorization and normalization (Section 7.4.4), dimensionality reduction (Section 7.4.5), and finally visualization and interaction (Section 7.4.6). Figure 7.2 provides a high-level overview of the approach.

7.4.1 Visual Analytics Approach

The visual analytics method consists of four steps, starting with dynamic network discretization. In this step the dynamic network data is prepared for analysis. The basic idea is to create a series of snapshots of the dynamic network at subsequent points (or short periods) in time based on event log data, such as transactions or communications. Each snapshot contains an instance G_i of the dynamic network.

The next step is simple but simultaneously crucial to our approach and is to the best of our knowledge not considered in previous literature: the network snapshots are vectorized and considered as points in high-dimensional space. This effectively represents the network edges at a point (or interval) in time as a row feature vector; each such point in this high-dimensional space represents the network at a different time-interval. The position of the snapshots in this space provides insights in the evolution of the network: snapshots where networks are similar will be positioned closer to each other and form clusters. Similarly, for timesteps where the network is different compared to more common network snapshots, the points will be outliers. Clusters of points indicate stable or recurring network states. Points lying in between clusters provide insight in how the network evolves from one state to another.

To enable analysis and exploration of the dynamic network snapshots we apply dimensionality reduction techniques and project the points to two dimensions in the third step. Finally, we introduce two juxtaposed linked views: a projection of the snapshots and a second view that provides a network visualization for a selected snapshot.

7.4.2 Dynamic Network Model

We model a dynamic network Γ as a sequence of N snapshots:

$$\Gamma = (G_1, G_2, \dots, G_N), \quad (7.1)$$

where a snapshot is a directed graph $G_i = (V, E_i, t_i)$, with node (vertex) set V and edge (link, transaction, event) set $E_i \subseteq V \times V$ with from and to vertex tuples (v_m, v_n) , and where t_i denotes the i -th timestep. The set of all edges in the dynamic network is the union of edge sets of all snapshots:

$$E = \bigcup_{i=1}^N E_i. \quad (7.2)$$

Furthermore, we consider weighted graphs. Let w be a function defined on the edge set as $w : \mathbb{N} \times E \rightarrow \mathbb{R}$ assigning a real-valued weight $w(i, e)$ to edge $e \in E_i$.

7.4.3 Discretization of Dynamic Networks

In practice, datasets containing timestamped activities (e.g., log-files, email-traffic) are more ubiquitous than predefined sets of snapshots of a dynamic network. If these activities involve two objects each, like for instance sending a message from one address to another or a financial transaction between two accounts, it is natural to view such a dataset as a dynamic network. An activity log is modeled as:

$$A = (A_1, A_2, \dots, A_M), \quad (7.3)$$

where $A_j = (a_j, s_j) \in E \times \mathbb{R}$ is an activity with edge $a_j(v_m, v_n)$ and time-stamp s_j . The activity log is transformed to a dynamic network by creating a sequence of snapshots for analysis. We assume the time points t_j are equidistant and $t_{j+1} - t_j = \Delta t$ is constant, representing days, hours, *et cetera*. Each snapshot G_i has an associated sampling time-window $[t_i - \omega/2, t_i + \omega/2)$ with width ω . Now the edge-set E_i of a snapshot is:

$$E_i = \{a_j \mid s_j \in [t_i - \omega/2, t_i + \omega/2)\}. \quad (7.4)$$

The weight of an edge e in snapshot i is simply the number of edges in the edge set, i.e., set cardinality:

$$w(i, e) = |\{A_j \mid a_j = e \wedge e \in E_i\}|. \quad (7.5)$$

To create the snapshots we have different choices for Δt . In the most extreme case each snapshot contains a single activity and at the other end of the spectrum there is only one snapshot containing all activities. Neither of these two options is desirable. In practice it is best to choose Δt based on domain knowledge or such that they result in a logical division of the timespan of the data, e.g., divide a day in hours, half-hours or quarters of an hour. A visualization of the events over time (e.g., a Reordered Massive Sequence View [268, 269]) could help in determining an appropriate window-length.

Here we choose to keep Δt constant. It may be justifiable to have varying lengths, however, this makes interpretation later on more difficult. To prevent missing patterns due

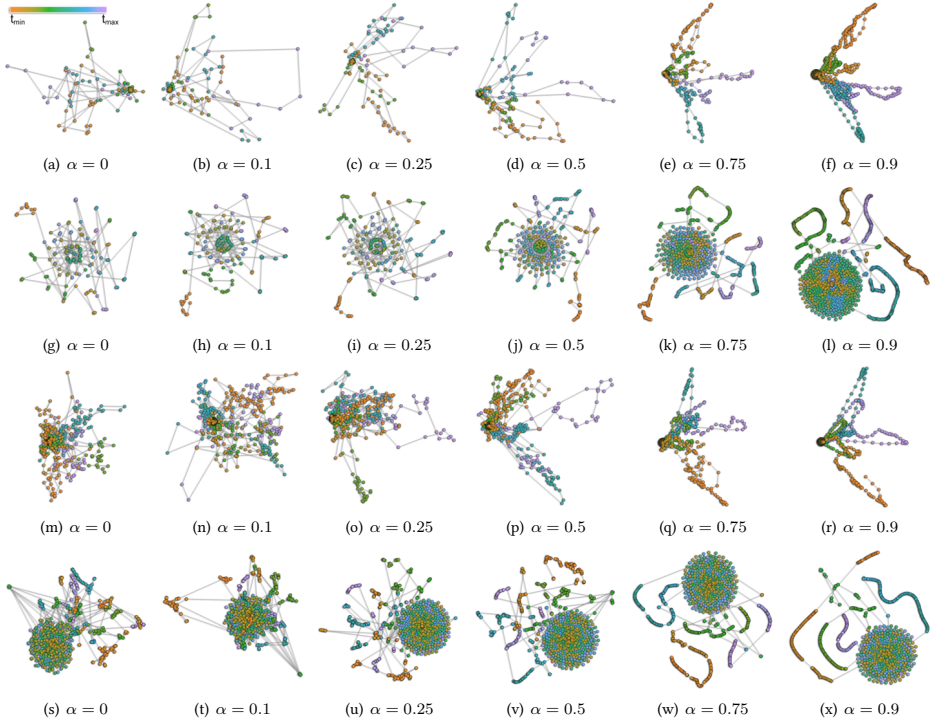


Figure 7.3: Effect of different overlap values α for the snapshots in the discretization process. Images show a two dimensional projection for the thiers-2011 [100] dataset computed using linear dimensionality reduction PCA (first and third row) and non-linear dimensionality reduction t-SNE (second and fourth row). The window width ω of the snapshots is kept constant in the first two rows, Δt (and hence the number of samples) varies. The number of snapshots a-f and g-l are, 151, 167, 201, 301, 602, 1500. In the third and fourth row the number of snapshots is kept constant at 1000 and the window width ω is varied. Generally, the more overlap, the more structure is visible in the resulting projections. Subsequent snapshots are connected with lines to reveal patterns over time. Points are colored according to time with a perceptually linear colormap (top left).

to hard boundaries, time-windows are allowed to have an overlap α with neighboring time-intervals using a sliding window, *i.e.*, we use $\omega > \Delta t$. The window acts as a moving average, effectively smoothing the data by adding new edges and dropping old edges as time progresses. The overlap $\alpha \in [0, 1]$ of successive windows is defined as:

$$\alpha = (\omega - \Delta t) / \omega. \quad (7.6)$$

Instead of uniform weighting of instances across time intervals, *i.e.*, using a box kernel, smoother kernels could be used. However, the simple scheme used here gives already satisfying results and is easier to explain to non-experts. Figure 7.3 shows the influence of different fractions of overlap on the final projection for an example dataset: Figures 7.3, 7.4, 7.5, and 7.6 use the same dataset, which is described in detail

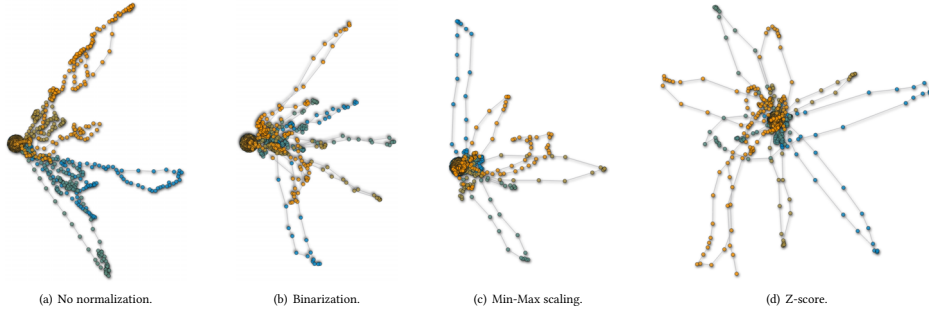


Figure 7.4: Linear dimensionality reduction technique PCA applied on dynamic network snapshots of the thiers-2011 [100] dataset a) without normalization, b) binarization, c) min-max scaling, and d) z-score normalization. Z-score normalization is the preferred choice for PCA since we are interested in the components that maximize the variance, however, no normalization already reveals much of the structure and evolution of the dynamic network.

in Section 7.5.2. We suggest a fractional overlap of 0.5 or more for a smoother image, however this is dataset dependent.

Each snapshot contains a number of edges, that when combined form a (static, flattened) weighted network. A snapshot G_i can be represented as an *adjacency matrix* M_i of size $|V| \times |V|$ with:

$$M_{i,jk} = w(i, (v_j, v_k)). \quad (7.7)$$

As a result, we have a set of N matrices with snapshots that together define the dynamic network. In summary, we introduce discretization to create snapshots for activity-based datasets, and as an added bonus we can control the number of snapshots to better scale and deal with large datasets. Overlap is introduced to smooth snapshots of subsequent time-steps, which improves the interpretation of the final projection. Also, overlap reduces missing temporal patterns with otherwise hard boundaries. The resulting snapshots of the dynamic network are used in the next step of our approach by interpreting each snapshot as a point in high-dimensional space.

7

REDUCING SNAPSHOTS TO POINTS

7.4.4 Vectorization and Normalization

Now that we have snapshots of the dynamic network, the goal is to analyze these and gain insight in the evolution. We enable this by simply reducing each snapshot to a point. In the following we describe how this is achieved.

All N network snapshots represented by $|V| \times |V|$ adjacency matrices are rearranged to $1 \times |V|^2$ row-vectors which are points in $|V|^2$ -dimensional space. The row-vectors are stacked to form a $N \times |V|^2$ matrix (see Figure 7.2, step 2). The columns of this matrix represent edges of the network and the rows represent different snapshots. A column is a feature vector, which represents the behavior of one edge over time. Instead of the adjacency matrix, also other attributes derived from the matrix can be used. Examples

are simply the number of (active) edges and vertices, or the degree distribution of the snapshot.

The next step in the approach is to reduce and project the high-dimensional points to two dimensions. This is done using linear and non-linear dimensionality reduction methods, a heavily studied topic in statistics and data-mining. Before applying the methods the matrix can be normalized to achieve better projection results; that is, we normalize each dimension of the high-dimensional space. We performed experiments with three different normalization methods: binarization, min-max, and z -score, see Figure 7.4. With binarization, 1 is assigned to each cell with a value $M_{i,jk} > 0$ and 0 otherwise. Min-max normalization scales the data to a fixed range $[0, 1]$. Z -normalization transforms the data to have zero mean and unit variance by subtracting the mean and dividing by standard deviation. For a linear dimensionality reduction method, such as PCA, z -normalization is the preferred choice, since we are interested in the components that maximize the variance. Min-max normalization leads to smaller standard deviations, thus suppressing the effect of outliers.

For the cases we considered, there was no clear best method, and the choice seems to depend on the dataset and patterns to look for. Normalization might emphasize specific patterns in the data for easier identification, for example, z -score normalization suppresses outliers while min-max scaling preserves outliers. We achieved good results without normalization and leave further exploration for future work. Also, it is not strictly necessary, because every feature is measured on the same scale and unit. However, if derived attributes are added it is advised to apply normalization first.

7.4.5 Dimensionality Reduction

Our network snapshots are points in high-dimensional space but multiple dimensions are hard to comprehend and difficult to visualize. Therefore we use dimensionality reduction techniques [277], which project points to lower dimensional subspaces such that data characteristics of the features in the lower dimensional subspace approximate geometric characteristics of the data in the original high-dimensional space. Our goal is to reduce the feature space to two dimensions and project the snapshots as points for visualization and interaction.

There are many linear and non-linear dimensionality reduction techniques that can be used, such as *Principal Component Analysis* (PCA) [141, 203], *Multidimensional Scaling* (MDS) [176], or *t-Distributed Stochastic Neighbour Embedding* (t-SNE) [276]. The computation times for both PCA and t-SNE can be strongly reduced by using improved variants such as *Randomized PCA* [127, 217] and *Barnes-Hut-SNE* [275], respectively. This makes them usable for large datasets and enables interactive real-time analysis. PCA is a linear dimensionality reduction technique, *i.e.*, the resulting dimensions are linear combinations of the original dimensions such that the variance of the data is described best. This restriction can be overcome by applying a kernel-trick [235] to achieve non-linearity [226].

A recent comparison reveals that non-linear dimensionality reduction techniques perform well on selected artificial tasks, but PCA still has a better performance on real-world

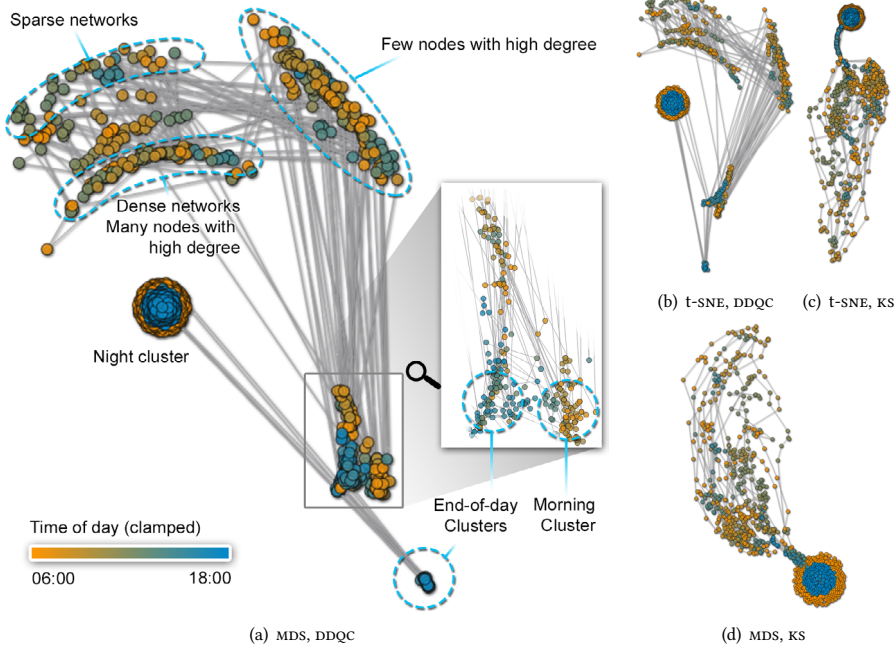


Figure 7.5: Non-linear dimensionality reduction using a,d) MDS and b,c) t-SNE for degree distribution distance measures c,d) Kolmogorov-Smirnov and a,b) Degree Distribution Quantification and Comparison [12], applied on the thiers-2011 dataset. Both non-linear dimensionality reduction methods result in similar images. This projection reveals a large night cluster, morning and end-of-day clusters that are connected to the night state. Three more network states are identified; a cluster with a network state where a few nodes have a high degree, a cluster with a sparse network state, and a cluster with a dense network state.

datasets [277]. Therefore, we choose for (Randomized) PCA as initial dimensionality reduction technique, but also provide MDS and t-SNE in our application, using source code provided by the authors [275, 288]. Both MDS and t-SNE have the advantage that besides the standard Euclidean distance between network snapshots, more sophisticated network distance measures can be directly employed with the use of a pre-defined distance matrix. For example, we can precompute distances based on degree distribution. As a measure for this, the widely used Kolmogorov-Smirnov test [174], or the more recent Degree Distribution Quantification and Comparison (DDQC) measure [12] can be used. This provides us with a more structural comparison of the network over time, see Figure 7.5.

The dimensions of the resulting dimensionality reduction algorithm are often difficult to interpret, especially for the non-linear dimensionality reductions such as MDS and t-SNE. However, this is not an issue here, because we are not interested in the dimensions but rather in the clusters and outliers of the resulting projection of the dynamic network snapshots. For analysis and exploration, the snapshots, now reduced to points, are visualized in the resulting two dimensional space.

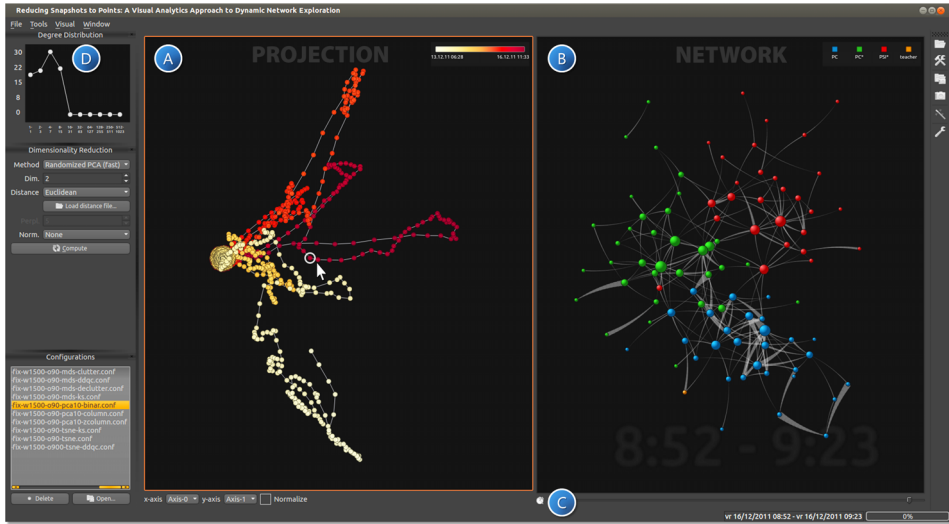


Figure 7.6: Graphical user interface of the prototype implementing the different components of the visual analytics approach with (A) the projection view with snapshots of the dynamic network reduced to points, (B) the linked network view with a node-link diagram of a selected snapshot in the projection view, and (C) linked time control, and (D) auxiliary degree distribution view for a selected snapshot.

7.4.6 Visualization and Interaction

To enable the exploration of the dynamic network we use two linked juxtaposed views implemented in a prototype application. Figure 7.6 shows the graphical user interface of the prototype with the *projection* view and the *network* view. See the video in the supplemental material¹ for a demonstration of the different interaction methods.

Projection View The projection view visualizes each snapshot of the dynamic network as a dot. Dots that are close to each other indicate that the snapshots have a similar network state. Sequential dots in time can be connected with a line to emphasize transitional patterns between clusters. This also enables the identification of snapshots close in time, but far apart in the projection view and vice versa. Dots are assigned a user defined colormap that is initially set to index in time. By coloring dots according to time more structure in the clusters can be identified. If a cluster has a (nearly) uniform color, this indicates a stable state in the network, *i.e.*, the network stays the same for a longer period. If a cluster consists of multiple colors, it reveals a recurring state. Next to color-mapping by global time index as described above, other options are available such as coloring dots according to hour of the day to reveal day patterns. Finally, dots could also be colored according to a derived attribute such as network density to reveal structure.

Exploration is supported by zoom-and-pan techniques. Zooming facilitates the close up inspection of clusters of snapshots. To simplify the identification of clusters

¹<http://www.stef.vdelzen.net/dissertation>

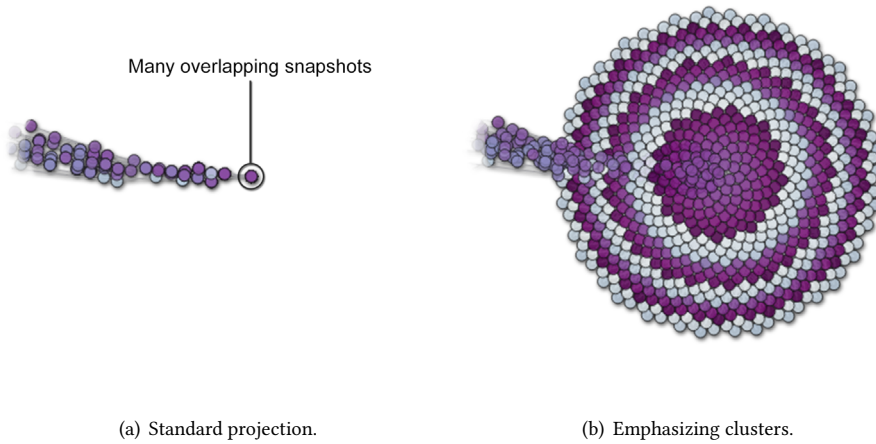


Figure 7.7: Simplifying the identification of states in the network by emphasizing clusters in the projection view by repositioning nodes with equal coordinates. a) Many snapshots containing similar network states that are given equal coordinates and b) after repositioning for better identification of the network state.

we de-clutter the projection view by repositioning points with equal positions using Phyllotactic arrangement techniques [284]. This results in the emergence of clusters that emphasize the states in the network, see Figure 7.7.

We use two dimensional projections to prevent clutter and to keep navigation techniques simple. However, different axes resulting from the dimensionality reduction can be selected to project the snapshots. For example, it may be fruitful to check dimensions beyond the first two dimensions if PCA is used. Additionally, time can be selected as one of the axes of the projection. Upon repositioning of the points due to switching between different projections the users mental map is preserved by using animation to support the impression of one unified space and also simplifying the tracing of points. Dots can be selected and highlighted to show the associated network snapshot in the network view.

Network view The network view visualizes the network of the selected snapshot using a node-link diagram where each dot represents a node of the network and each line an edge. The network configuration is computed using a user selectable force-directed graph layout [112].

The nodes in the network view can be set to be visible and static for each snapshot or be repositioned on each newly selected snapshot. The advantage of recomputing the layout is that it often results in a clearer visualization of the structure of the network, however, this might break the preservation of the mental map. Stability of the nodes is supported by initializing each node position to the current position when computing a new force-directed layout. If nodes are repositioned, animation is used to make tracking of changes easier. Also, a static layout based on the union of all snapshots can be used.

Nodes in the network can be colored according to an associated multivariate attribute value. Similar to the projection view, exploration is enabled by zoom-and-pan techniques. Furthermore, the start- and end-time of the highlighted snapshot is rendered in the network view to provide context. Simple linking and brushing techniques are implemented to enable visual querying. Brushing nodes in the network view highlights snapshots in the projection view according to user selectable rules: a snapshot is highlighted if a) one or more nodes have a link, b) all nodes have a link, c) one or more nodes have a link with each other, d) all nodes are linked to each other. This enables users to visually perform queries on the behavior of particular nodes over time.

In addition to the projection view and the network view we implemented two auxiliary views, a linked *timeline* control and *degree distribution* view. The timeline view provides a context and enables to scan through the dynamic network. The degree distribution view shows the degree distribution of the network that is highlighted in the projection view and shown in the network view. This enables the comparison of snapshots on a more structural level and proves useful if alternative distance measures are used based on degree distribution, such as in Figure 7.5.

7.5 Use Cases

We have applied our approach to artificial and real-world datasets. We present results and discuss choices for parameters. The visual analytics approach helps in understanding the network dynamics by revealing stable states, recurring states, outlier states and provides insight on the evolution of the networks in general.

7.5.1 Artificial Dynamic Networks

We created different artificial dynamic network datasets to test and evaluate the visual analytics approach. We show the results of our approach on two of such datasets here, a third example is shown in Figure 7.1. In our network creation model we define the number of nodes and number of stable states in the dynamic network. A stable network state is created by using the small world model of Newman, Watts, and Strogatz [198, 295]. Next, small variants of the resulting network are created for a number of user defined timesteps. After creation of a number of these stable states, transitional network states are created by interpolating between the stable states over a predefined number of timesteps.

First, we create a dynamic network with 100 nodes, 9900 edges, and 650 timesteps. The dynamic network contains four stable states and three transitions; at the start the network consists of one large small-world community. Next, the community falls apart and two separate communities emerge. Then one of the smaller communities is divided into two even smaller communities. Finally, one of these joins the largest community again. All states are stable for 125 timesteps and transitions take 50 timesteps.

If PCA is applied to the network snapshots we clearly see four clusters of snapshots and the transitions between them (see Figure 7.8). By highlighting the snapshots in the

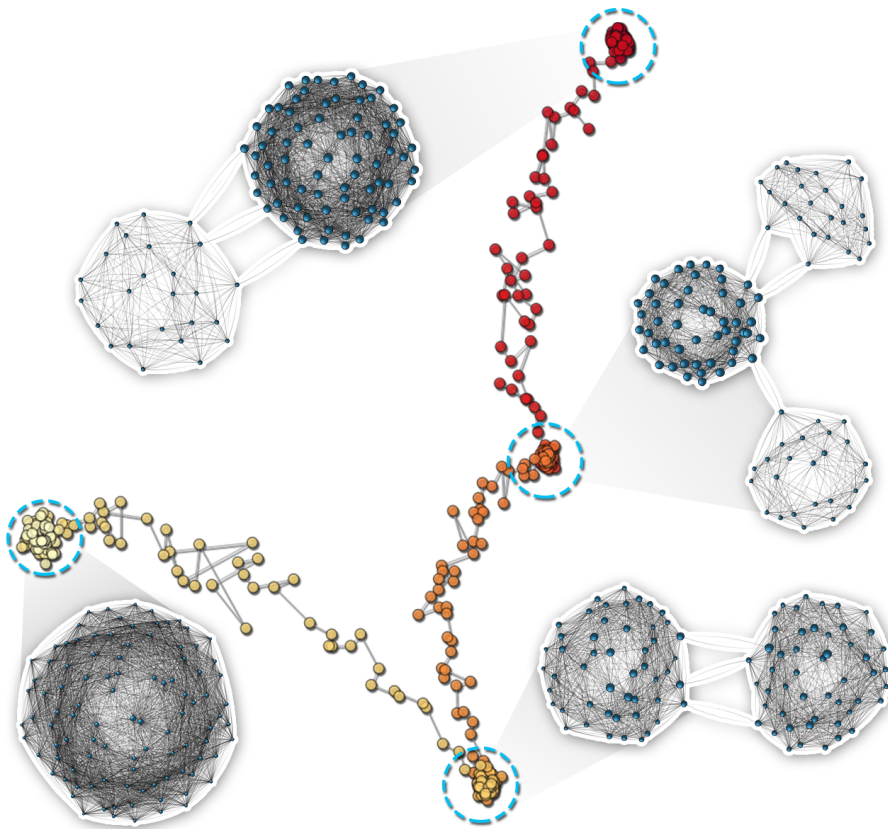


Figure 7.8: Linear reduction with PCA reveals four stable network states and transitions between them. Gray insets show representative snapshots.

clusters we discover the structure of the network in the network view. If we select time on the x -axis we see that the last three stable states are more similar than the first stable state as their distance to each other on the y -axis (first principal component) is smaller, see Figure 7.9(a). If we apply z -normalization, the structure within clusters becomes clearer, and the clusters themselves are emphasized. However, the transitions between the clusters become less clear, see Figure 7.9(b). Next we switch to t-SNE, Figure 7.9(c), and the clusters of stable states become even more clear, however, we do not see the transitions between the stable states anymore. This might be resolved after fine-tuning of the algorithm parameters.

The second dynamic network, again 100 nodes and 9900 edges, contains four states, five transitions and has a recurring state in the network that occurs at different points in time. The network starts with one big community (the recurring state) and then falls apart in two smaller communities. Next one big community is formed again after which the network again falls apart into two different smaller communities. Then the network splits into three communities and finally merges back to one big community again. The

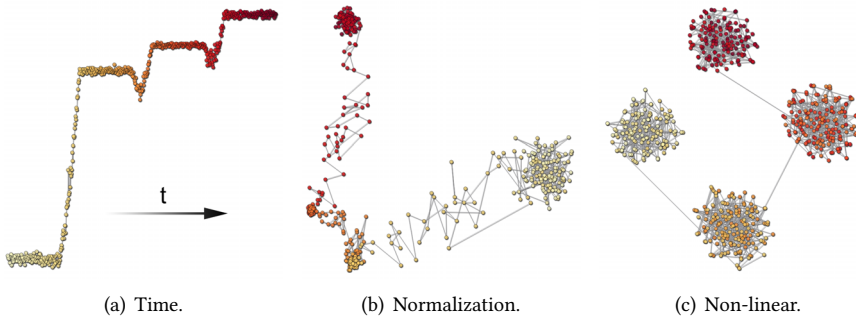


Figure 7.9: Alternative projections of the dynamic network snapshots: a) time vs. 1st principal component, b) PCA with z-normalization, and c) non-linear dimensionality reduction with *t*-SNE.

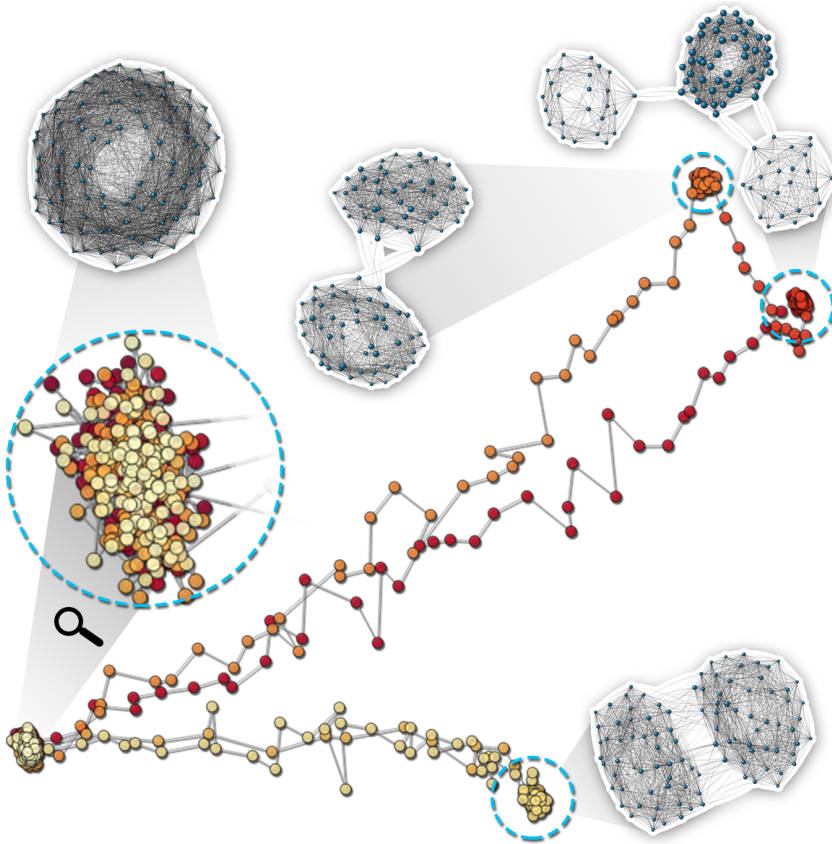


Figure 7.10: Linear reduction PCA reveals four network states, three stable and one recurring state. The largest blue circle shows a zoomed-in version of the recurring state and the gray insets show representative snapshots.

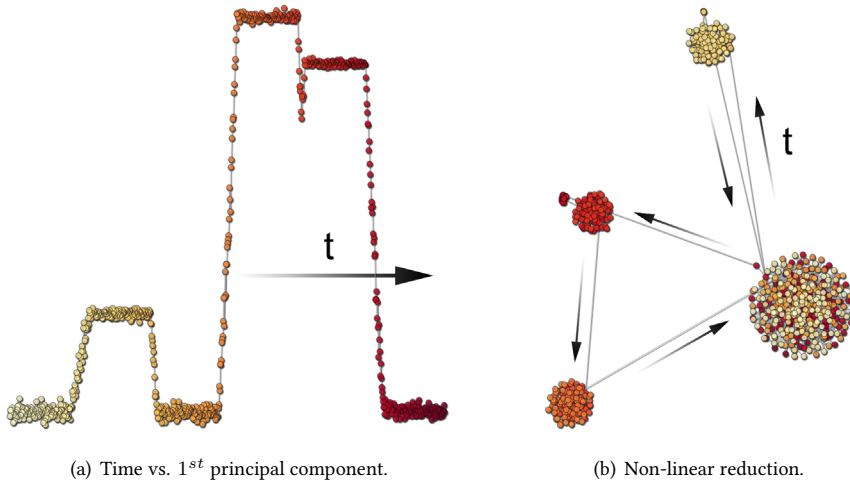


Figure 7.11: *Alternative projections providing more insight in the same dataset as in Figure 7.10.*

sequence consists of 900 timesteps.

First we again apply PCA on the snapshots enabling us to identify the four different stable states. We see the transitions between the states and identify one state as recurring because the cluster of snapshots consists of different colors, see Figure 7.10. If we switch to time on the x -axis we more clearly see the stable states and their duration in Figure 7.11(a). Furthermore, we now see when in time the recurring state occurs. By switching to t-SNE we see the stable states as clusters (see Figure 7.11(b)). The transitions between the stable states are not visible here, but we are still able to identify the sequence of states due to the connected subsequent snapshots. Also, the recurring state is more clear here due to a better spread of the points in the cluster.

7.5.2 High-School Contact Patterns

Two different datasets were collected from the SocioPatterns initiative [8]. Both datasets contain timestamped events of face-to-face contact between persons based on wearable sensors in context of determining how infectious diseases spread within a population. The first dataset contains face-to-face contacts of high-school students between three different classes during 4 school days in 2011 (see also Figures 7.3, 7.4, 7.5, and 7.6). The second dataset contains 7 school days of face-to-face contacts (from a Monday to the Tuesday of the following week in 2012) between students of 5 different classes, again logged at the French high-school Lycée Thiers in Marseilles [100]. The results we obtained with the two datasets are similar in nature with identical findings, therefore we only discuss the richer 2012 dataset below.

The dynamic network consists of 180 nodes (students), 45,047 contacts, and 10,104 unique edges. We create snapshots by choosing a window width ω of 60 minutes with an overlap α of 0.9. This results in 6 minutes (Δt) of new edges and dropping old edges for each subsequent snapshot. The total number of produced snapshots is 2015.

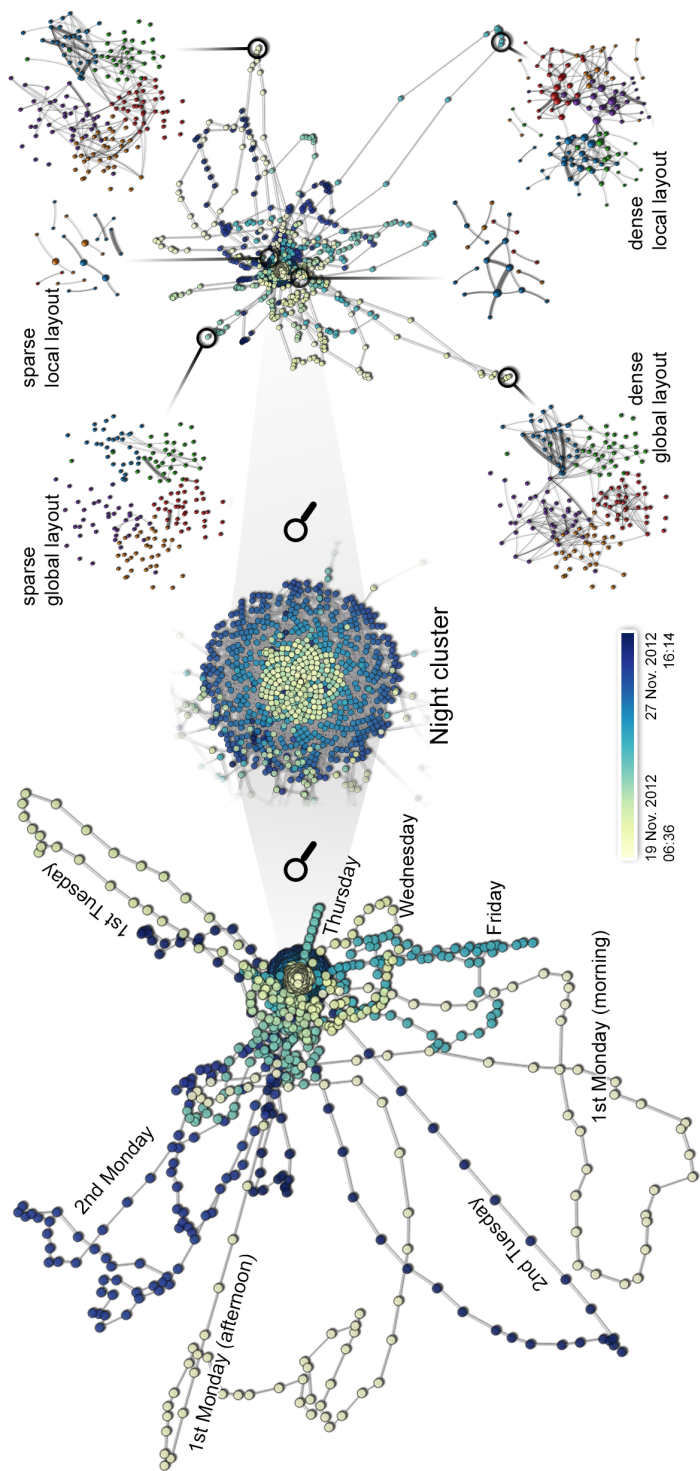


Figure 7.12: Linear reduction PCA without (left) and with (right) z-normalization for the thiers-2012 dataset. The middle cluster shows a zoomed-in version of the recurring night state. Days are identified by color and form a loop that starts and ends at the night cluster. The annotated networks show the network structure for the black-circled snapshots in the projection.

First we apply PCA to the snapshots (see Figure 7.12). Here we use a linear color-scale and the first thing we notice is that each day can be identified by a separate color. Furthermore, we see a big cluster of snapshots in the middle. This cluster consists of different colors, so this indicates a recurring state in the network. After inspection by highlighting, we see that this cluster represents the network during night, *i.e.*, no activity. Each day starts at the night cluster and returns to it with a loop.

Next, we use z -normalization and apply PCA again (see Figure 7.12, right). Similar to before, the recurring night cluster is present and the days are visualized as loops. There is a more clear separation now between clusters far away and close to the night cluster. The snapshots in the clusters further away from the center contain more dense networks whereas the clusters closer to the center (night) contain sparse networks. If snapshots are colored according to hour of day, we see that the dense network states often occur at the start of the day and in the afternoon the network is predominantly sparse.

If we switch the x -axis to time and the y -axis to first principal component, the nights and weekend are clearly visible in Figure 7.13(a). It also indicates that all days have a similar variance as the day patterns have similar shape.

We continue the exploration by switching to dimensionality reduction using t-SNE. Again, the night cluster appears and the days are shown as separate trails. Here we conclude that likely each day is different from one another, because no trails are overlapping or forming clusters, see Figure 7.14(left). We also see more structure in the days by means of gaps in the trails. This indicates a daily rhythm, presumably caused by breaks between classes and lunches. If we color dots by hour of day we see that the gaps occur at similar time-periods in Figure 7.14(right). Upon closer inspection we identify a cluster of snapshots that is present in each day between 8 and 9. At this cluster of snapshots the network is very dense and the clusters before and after are both sparse. Now that snapshots are colored by hour of day we see that one trail of snapshots is different compared to the rest. The Wednesday trail is shorter and afternoon snapshots

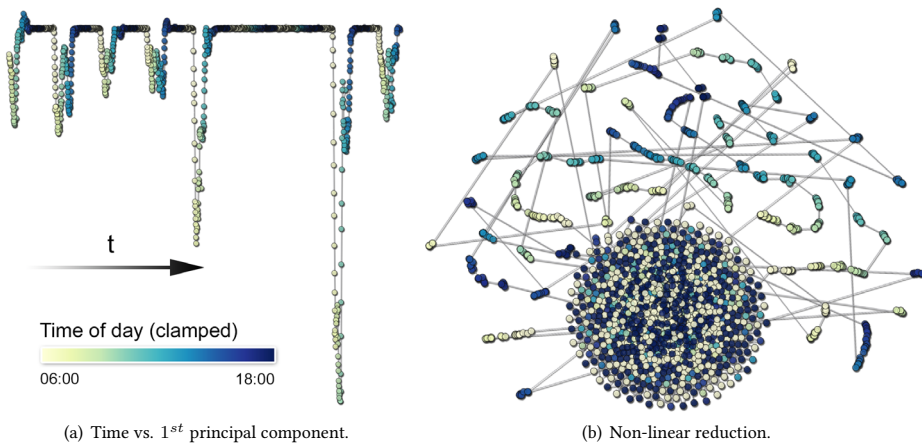


Figure 7.13: Alternative projections with snapshots colored by hour of day.

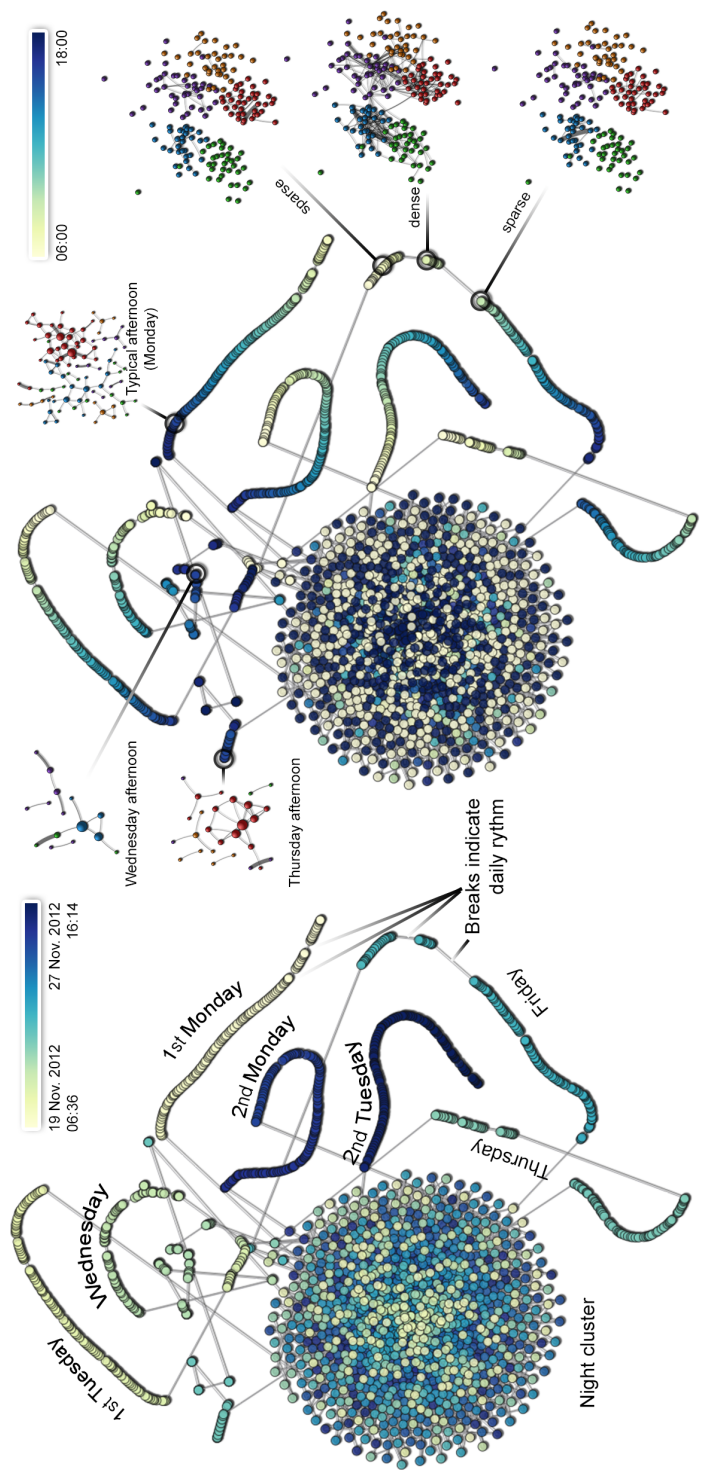


Figure 7.14: Non-linear dimensionality reduction of snapshots colored globally (left) and by hour of the day (right) revealing patterns and providing insight in stable states and the dynamic network evolution. The annotated networks show the network structure for the black-circled snapshots in the projection.

are not attached. After inspection we see that the network state at Wednesday afternoon is much sparser compared to the other days. This is explained by the fact that students take exams on Wednesday which typically last the whole afternoon without breaks [100]. Also, a cluster of snapshots is identified representing the network state at Thursday afternoon. Here, this network state represents the communication of one class only.

Finally, we apply z -normalization before applying the t-SNE dimensionality reduction and this results in many small groups of snapshots that generally span a time period of an hour (see Figure 7.13(b)). This indicates that classes probably last an hour. Furthermore, we see that mornings are more consistent and afternoons differ, indicated by longer trails in the morning and smaller clusters in the afternoon.

7.6 Discussion

The basic steps of the visual analytics approach presented in this chapter are all existing techniques: 1) discretization, 2) vectorization and normalization, 3) dimensionality reduction, and 4) visualization and interaction. However, their combined use for the analysis of dynamic networks is novel and provides new views and insights. The cornerstone of our approach, both simple and effective, is to reduce snapshots of a dynamic network to points.

There are many different choices for the exact implementation and parameter settings of our approach, for example, how to choose the time step and window overlap in discretization, the dimensionality reduction technique, and what (derived) features to use. Many options depend on the characteristics of the dynamic network to analyze and might be best determined in a joint effort between domain expert and a visualization professional or data scientist. Still, we consider this flexibility a strong point as it enables to use the approach for a broad range of datasets and perspectives on these, and opens possibilities for future work.

We found that in general non-linear dimensionality reduction techniques work best for the reduction of snapshots to points. This has the downside that it is harder to interpret the resulting dimensions. Though, this is not that important since we do not need a precise explanation but rather the ability to visually identify clusters and outliers.

There are two types of visual scalability concerns in our method; scalability with respect to network size and number of timesteps involved. In our case we found that our approach works well for a network with 180 nodes, 10,104 edges, and 2015 timesteps. We believe our method is able to deal with both aspects of scalability, up to a certain limit depending on memory- and time-availability. If the network at each snapshot is large, then for the visualization in the network view a (sorted) visual adjacency matrix [180] or hierarchical edge bundles [137] have to be used rather than a node-link diagram. Also, the choice for window-length of the snapshots commonly influences the network size at each snapshot; in general, if window-length is chosen smaller, the resulting networks at each snapshot will also be smaller compared to larger window-lengths. Also, scalability for the number of timestamps can be controlled in the discretization step. If only a few

timesteps are available the method will be less effective, since we rely on the visual identification of clusters. Therefore, one should aim for at least 20 timesteps.

Computational scalability of the dimensionality is addressed by using scalable variants of PCA and t-SNE, as described in Section 7.4.5. This enables near real-time interaction and being able to deal with large datasets.

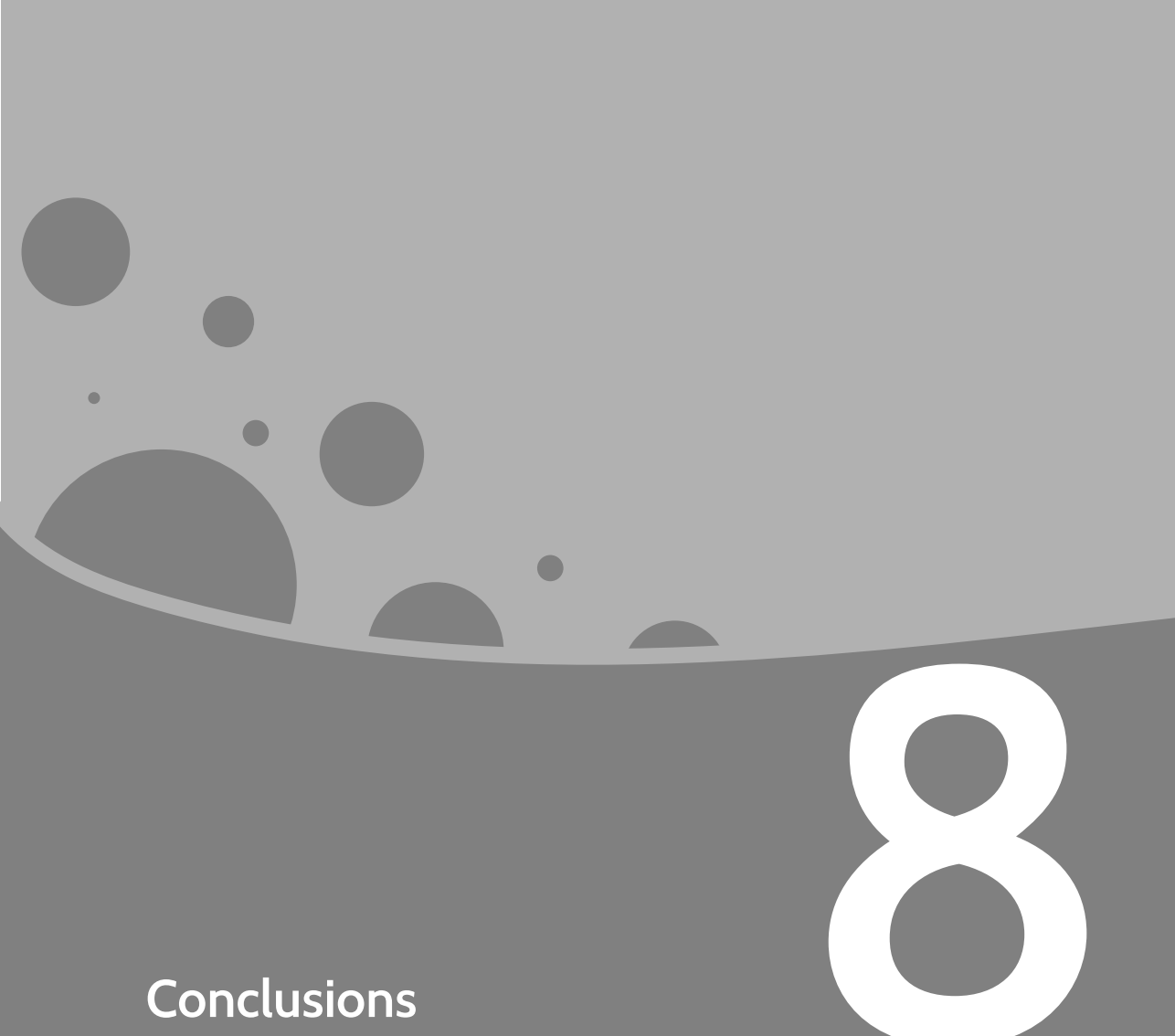
7.7 Conclusions

In this chapter we presented a novel visual analytics approach for dynamic network exploration. The approach consists of four different steps: *discretization*, *vectorization and normalization*, *dimensionality reduction*, and *visualization and interaction*. The crucial step in the approach that enables the exploration and analysis of dynamic networks is to reduce snapshots to points. The individual components are not new, however, the combination and their application to dynamic networks is novel. This enables users to visually identify stable and recurring states in the network and provides insight in the transitions between them. As a proof of concept the approach is implemented in a prototype. The approach is highly flexible and adaptable to users needs, e.g., how to create the snapshots and what dimensionality reduction technique to use. For the creation of snapshots we suggest to have at least 50 percent overlap of subsequent timesteps. However, this is merely a guideline and depends on the dataset at hand. We experimented with different dimensionality reduction techniques and found that standard PCA without normalization gives good results but non-linear reduction methods such as t-SNE work best. However, the general approach is independent from implementation details and with this work, deviating from standard approaches such as animation, timeline and small multiples, we hope to have inspired new work in the area of dynamic network exploration. We have shown the effectiveness of our approach by applying it to artificial and real-world dynamic networks and were able to get insights and understanding on the evolution of the networks.

7.7.1 Future Work

For future work there are several directions. First, the parameters involved in the different steps of the approach might be improved. Finding the best distance measure and dimensionality reduction technique is a challenge and typically depends on the dataset.

Second, the approach now relies on visual identification of clusters and outliers, which might be improved with automatic clustering in high-dimensional space, outlier detection [16], or graph similarity [33]. The visualization and interaction step might also benefit from high-dimensional visual analysis methods, e.g., a dual setup between item space and dimensions space [266]. Finally, the visualization setup might profit from other linked views showing network complexity properties.



Conclusions

8

8.1 Conclusions

In the previous chapters we presented interaction techniques, visualization methods, and recommendations and guidelines for the exploration and analysis of multivariate data, multivariate networks, dynamic networks, and combinations thereof.

Due to the complexity of dynamic multivariate networks, to develop a single visualization that depicts the structural changes of the networks, shows all multivariate attributes involved, reveals temporal and structural patterns, and exposes states in the network is a real challenge. To tackle this, we believe that interaction plays a key role in enabling users to gain insight and understanding of dynamic multivariate networks. Next to interaction, multiple perspectives on the data should be shown and each of these views should be linked to maximize contextual understanding. Furthermore, we argue that the workflow of an analyst should be reflected in the prototype application, automatic support for explanation and presentation should be provided, and interaction metaphors and visual variables should be used consistently across views and interface elements.

In the next paragraphs the conclusions of each of the individual chapters are repeated and briefly discussed followed by a discussion of re-usability and integration. Next we reflect on our contributions by extracting general guidelines for effective visualization techniques and interaction design to explore and analyse dynamic multivariate networks. Finally, directions for future work are identified and briefly discussed.

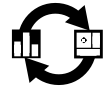
Small Multiples, Large Singles In Chapter 3 we introduced a new visual exploration method for multivariate data analysis using small multiples. We presented a model based on the alternation between large singles and small multiples. The small multiples are produced by applying split operations on large singles. We propose different split operations each having their own use. Furthermore, we introduce a navigation mechanism based on explicitly showing the visual history of the exploration path. The effectiveness of the exploration method is tested using a formal user study comparing four different interaction methods. We found users needed fewer steps in answering the questions and also explored a significantly larger part of the state space in the same amount of time, which gives them a broader perspective on the data, hence lowering the chance important data features are missed. Moreover, the small multiple exploration method offers comparison and guidance simplifying the exploration process. Users were more satisfied and preferred exploration methods using small multiples, but if a visual history should be integrated is still an open question and needs further investigation.



The presented method is not limited to multivariate data but can also be applied to dynamic multivariate networks. The dynamic aspect of the network, usually time, can be represented using a single attribute. Analysis is enabled by, *e.g.*, showing the network in a large single using a two dimensional embedding for the entire timespan. Next, the timespan can be refined and explored by splitting the large single in small multiples each showing a different time interval. During the exploration, multivariate attributes can be explored and different embeddings can be shown. In addition, integrated automatic methods in cooperation with the small multiples technique can be used to help with

the exploration. In Chapter 3 we have shown how visual analytics, e.g., interactive clustering, proves useful in multivariate data analysis. We can likewise explore the effect of parameters of network algorithms with similar split operations.

From Detail to Overview via Selections and Aggregations Chapter 4 presented novel interaction methods for both domain-experts and casual users to explore and analyse multivariate networks concurrently on network topology as well as the multivariate data. This enables users to see outliers, patterns and trends for the combined elements. Furthermore, we support users in the simultaneous creation of a high-level infographic-style overview. This helps in understanding the network, provides abstraction and aggregation, and presents a means for communication to a broader audience. Both interaction methods are facilitated by using selection sets as a central element and the juxtaposition of detail and overview.



We have shown the effectiveness of our DOSA approach through several elaborated examples on real-world datasets. Furthermore, we have shown this method is not just limited to multivariate networks, but also can be used when only multivariate data or network structure is available. Finally, due to the general and flexible setup, this method is domain-independent.

Although this method is targeted at multivariate networks we can extend this to also enable dynamic network exploration. Similar to the small multiple approach, time can be encoded as an additional attribute. By filtering using the selections of interest, a higher-level overview of the evolution can be produced for presentation and communication.

Massive Mobile Phone Data Exploration In Chapter 5 we aimed at developing tools and techniques for the exploration and analysis of massive mobile data supporting all aspects of the process. We identified user tasks and requirements from which appropriate visualization, interaction and automated support techniques are selected. Next, we implemented these in a highly interactive prototype.



We showed the effectiveness of our visual analytics approach by applying the prototype on massive mobile phone data containing 2.5 billion calls and SMS exchanges between around 5 million users located in Ivory Coast over a period of 5 months, provided by France Telecom within the context of the Orange D4D challenge. From the typical use cases obtained while browsing the data, we extracted significant and interesting events by cross-correlating these using UN reports and weather information.

In this chapter we touched upon the combined elements of *dynamic*, *multivariate*, *network* and *big data*. It shows and discusses the various challenges involved and attempts to solve these with a visual analytics approach. Besides the temporal and structural elements there is a geospatial element that plays a key role in the exploration. All three elements (spatial, structural, temporal) can be explored with coordinated multiple views in one unified framework.

Extended Massive Sequence Views In Chapter 6 the msv is utilized to visualize the dynamics of networks. Different characteristic temporal and structural patterns are defined in terms of the msv and according reordering strategies are proposed, making the patterns stand out, such that they are easy to identify and interpret. We present three main reordering strategies, based on node structural properties, visual edge properties, and prevention of block overlap in time. Furthermore, we present a practical solution to combine the different reordering techniques approximating the optimal solution, using simulated annealing and ϵ -constraint multi-criteria optimization, providing good solutions within a short amount of time. We believe the reordering techniques are powerful techniques and are valuable to other visualization techniques, in particular 1D network layouts such as arc-diagrams.



The effectiveness of the different reordering strategies is shown by applying them on generated and real-world datasets. From these use cases it becomes apparent that each proposed individual reordering strategy is useful and worth inspecting as well as the combined reordering strategies. The new node orders improve readability, reduce cognitive load, and bring forward temporal and structural features present in the data, leveraged by Gestalt principles for easy identification and effective exploration and analysis.

In addition, the standard msv is extended to enable the analysis of optional node data. To further reduce visual clutter and use space more efficiently we introduce the circular msv. Time-series information associated with the nodes can be analyzed using a hybrid approach of the circular msv and a representation based on CircleView [169]. We investigated how hierarchical node structure can be taken into account and present a solution that integrates this information into the reordering strategies using an iterative top-down and bottom-up process.

The msv does not reveal detailed topology of a network very well, hence it should be used as part of a coordinated view application with at least one other view conveying network topology, or as part of a tool-chain. Therefore, we believe the techniques presented in Chapter 6 are a step towards enabling temporal analysis and understanding of dynamic networks by using the presented reordering strategies for the msv.

Reducing Snapshots to Points In Chapter 7 we presented a novel visual analytics approach for dynamic network exploration. The approach consists of four different steps: *discretization*, *vectorization and normalization*, *dimensionality reduction*, and *visualization and interaction*. The crucial step in the approach that enables the exploration and analysis of dynamic networks is to reduce snapshots to points.



The individual components are not new, however, the combination and their application to dynamic networks is novel. This enables users to visually identify stable and recurring states in the network and provides insight in the transitions between them.

As a proof of concept the approach is implemented in a prototype. The approach is highly flexible and adaptable to users' needs, e.g., how to create the snapshots and what dimensionality reduction technique to use. For the creation of snapshots we suggest

to have at least 50 percent overlap of subsequent timesteps. However, this is merely a guideline and depends on the dataset at hand.

We experimented with different dimensionality reduction techniques and found that standard PCA without normalization gives good results but non-linear reduction methods such as t-SNE work best. However, the general approach is independent from implementation details and with this work, deviating from standard approaches such as animation, timeline, and small multiples, we hope to have inspired new work in the area of dynamic network exploration. We have shown the effectiveness of our approach by applying it to artificial and real-world dynamic networks and were able to get insights and understanding on the evolution of the networks.

This method focuses on understanding the evolution of the network. Here, static multivariate data is taken into account by using visual variables of the node-link diagram visualized for one snapshot. Nodes are colored to the according static node information. Edge width is varied for the dynamic edge data. Other multivariate data could also be included using different visual variables or glyphs by enriching the node-link diagram. Also, the visualization could be changed to a more specialized visualization depending on the associated multivariate data. Finally, the computed projection is mainly based on structural and temporal node-link patterns through vectorization of the adjacency matrix. A future extension could be to also, or solely, take multivariate data into account. We briefly touched upon these possibilities in Chapter 7 but a better understanding is needed to obtain general guidelines and recommendations.

8.1.1 Integration and re-usability

The techniques presented in this dissertation can be used in isolation, however, in combination the techniques result in powerful synergistic solutions. Some integrations are already discussed in the individual chapters, e.g., the integration of the small multiples, large singles technique in the DOSA approach of Chapter 4.

To position our solutions and to provide a guideline for future combination or integration of the techniques we decompose the title of this dissertation and for each aspect we depict the relevance of each chapter. This gives us an overview of aspect and chapter importance and what technique can be used to tackle a specific or combined problem. Moreover, this enables the comparison of different solutions.

The brightness of each block in Figure 8.1 shows the relevance for the intersections of aspects and solutions. Each chapter touches upon each of the aspects and for each solution at least two aspects are highly relevant. Overall, *visualization*, i.e., the creation of new visual encodings is deemed less important and more attention is given to *interaction* techniques. The *dynamic* aspect of networks as well as the *multivariate* aspect are given about equal attention. All chapters are highly relevant for the aspects *networks*, with the exception of Chapter 3, which provides a more generic solution for multivariate data. Figure 8.1 can be used to select suiting solutions for a given problem, e.g., DOSA can be combined with *extended MSVs* or with *reducing snapshots to points* if the network is both dynamic and multivariate.

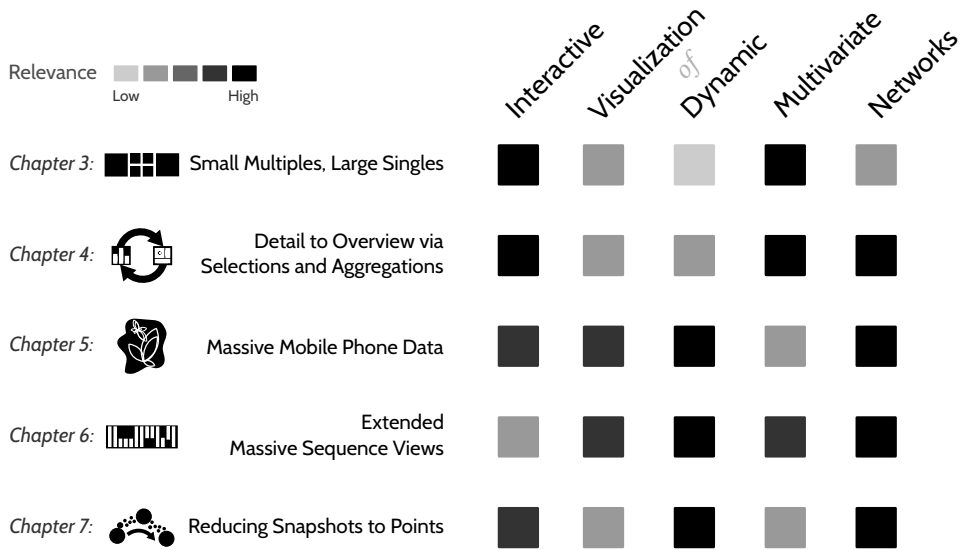


Figure 8.1: Overview of the solutions presented in each chapter with according relevance for each aspect.

A mock-up illustration of an example integration of the techniques in one holistic system is shown in Figure 8.2. Starting from two juxtaposed views similar to the DOSA approach, with one view showing the detailed network and the other showing a high-level overview, we extend this by integrating different solutions. The detailed network view can be used to show:

- the topology (structure) of the network using two-dimensional embedding algorithms;
- the multivariate data by superimposing edges on a scatterplot (Chapter 4);
- a projection of reduced snapshots of the network (Chapter 7); or
- the geospatial position of nodes with superimposed edges (Chapter 5).

Selections of interest are used for visual querying and play a central role. In the infographic-style overview we can show different visualizations for the selections depending on context; (aggregated) multivariate data can be shown here or a layout of the (aggregated) network if the selections of interest contain snapshots of the network. Next to topology and multivariate data we can show a selection of reordered *massive sequence views* revealing temporal patterns. We can even introduce a separate linked view for this if temporal patterns are important. The MSVs can also depict associated node hierarchies using icicle plots. Similarly, a circular MSV can show associated time-series for nodes, either as a linked view, or on demand with a tooltip style. Each step of the exploration path is preserved using a visual history that enables explanation, collaboration, and presentation. This mock-up is merely an example of how the different techniques can be integrated. By designing and integrating visualization concepts one should adhere to the principal design principles and techniques as introduced in

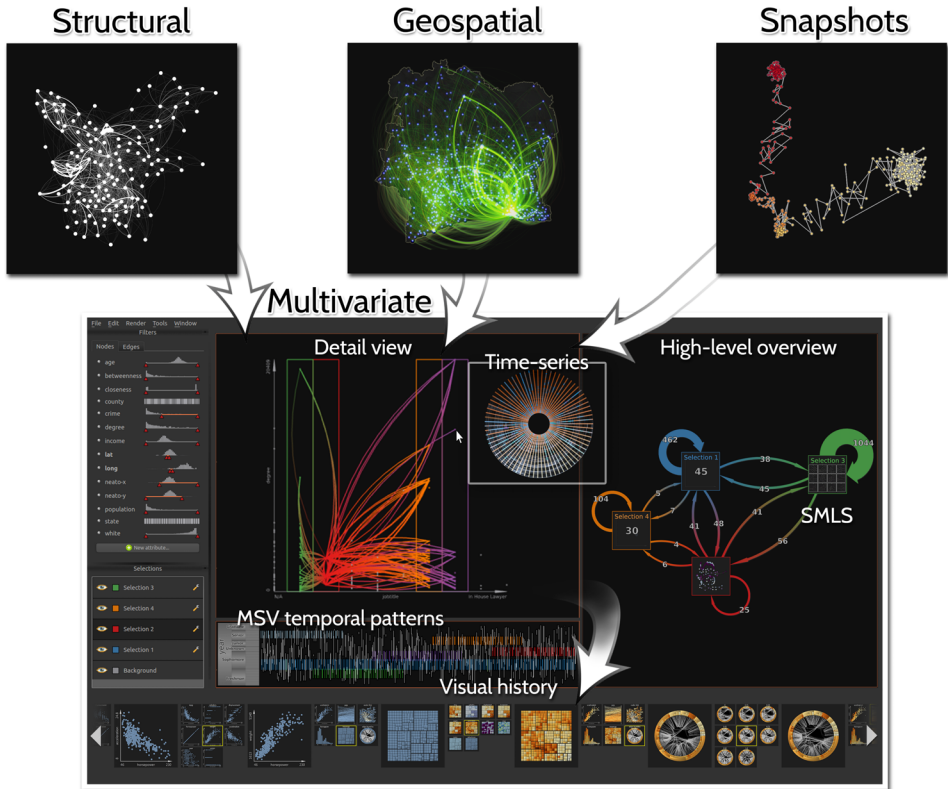


Figure 8.2: *Mock-up illustration of an example integration of the techniques presented in this dissertation into one unified framework for the exploration and analysis of dynamic multivariate networks.*

Chapter 1. In the following section we extract design rules from our own solutions and experience. The discussed design rules are applicable to general visualization design.

8.2 Reflections

In the introduction, Chapter 1, we discussed some of the fundamental visualization and interaction techniques and design principles. In the work presented in this dissertation and the developed prototypes to realize and evaluate the concepts, we strived to adhere to the discussed principles and concepts. Below we reflect on our work, both presented in this dissertation and earlier work, we extract several more design and interaction technique principles that are specific to our work and are not found in previous literature. We believe, when applied and integrated correctly, these concepts contribute to effective exploration and analysis in general; they are not specific to the exploration and analysis of dynamic multivariate networks, but applicable to interactive visualization design in general.

8.2.1 Design follows user workflow

The arrangement of the views, as well as the visualization design itself and available interaction techniques should follow the natural workflow of the user.

This design principle is most apparent in the work presented in Chapter 3: Small Multiples, Large Singles. Typically, in multivariate data analysis, users do not know what they are looking for in the data. A common interactive visualization such as a scatterplot does not help users in finding interesting patterns, due to the sheer amount of possible attribute combinations. In order not to miss anything, all combinations have to be tried. If we also take data filtering and visual encoding into account the number of possible combinations explodes. Thus, the natural workflow of the user here is to experiment with and explore different settings, then deepen and focus the exploration if something interesting is found. Furthermore, the different visualizations are compared by repeatedly switching settings in search of compelling patterns. Here we help users by providing a visualization and interaction technique that naturally follows this workflow. All different parameter values for data filtering, mapping and visual variables can be shown in an interactive small multiple setting that provides users with guidance and comparison. Intriguing patterns are directly visible and can be inspected in detail, next the exploration continues with that visualization. This workflow experience is enhanced by moving the exploration path containing all previous visualizations to the left upon important decision actions.

Also, the interaction and visualization design of the work presented in Chapter 4, DOSA, closely follows the user workflow. Here, users explore multivariate networks by starting with the creation of selections of interest. These can next be adapted using *direct manipulation*. Next, the projection can be changed while the selections of interest are preserved and thus can be viewed in a different context. Multivariate aspects of the selections of interest can be inspected in a second linked juxtaposed view providing a high-level overview. From this, the exploration is continued by further adjustment of the selections of interest in the detail view. Once interesting findings are gathered and hypotheses are tested, a high-level overview might be necessary to communicate a message to a broader audience. This presentable visualization is created simultaneously and automatically for the user already during the exploration of the data. The high-level overview can be edited with direct manipulation interaction operations, such as rearrangement, and showing and hiding details. Also, here the natural workflow of the user – going from exploration and analysis to presentation – is supported.

In Chapter 5 we deviate from this pattern of following the workflow of the user. In this chapter we first defined a minimum set of requirements that the application at least needs to support. However, from this it did not become clear how a typical exploration or workflow is performed. More specifically, we could not identify where users want to start the exploration. The data has three components that are all equally important, space, time, and structure. It is the combination of these components that reveals complex correlations. From these components there are multiple, logical starting points for exploration and analysis. It can start from a temporal pattern, a structural pattern, a spatial pattern, or a combination of these. Furthermore, on a conceptual level, the exploration can start with an overview and go into detail, or it can start

from a detail and then expand this to an overview. This not only depends on the data properties but might also depend on personal user preference and the topic of interest. Thus, contrary to logical sequential exploration steps, in Chapter 5 we rely on a flexible multiple coordinated view solution that tightly integrates spatial, temporal and structural perspectives on the data. This enables users to start the exploration with the properties of interest and can either start with an overview (measure view) or an interesting detail pattern (temporal matrix).

8.2.2 Automatic support for explanation and presentation

Automatic support for the explanation of findings and the presentation of the insights to a broader audience should be a principal element in the solution.

The solutions presented in Chapters 5 and 7 start with an overview of the data to begin the exploration by showing a *measure* view and a *projection* view, respectively. However, related to *following the user workflow* we believe it is not always best to implement the *information visualization mantra* (see Chapter 2): starting with the overview might not be appropriate. In addition, findings, and the exploration path leading to it, often need to be communicated to a broader audience and (semi-)automatic support for this should be provided.

An example implementation of this design principle is most apparent in the work of Chapter 4, where the multivariate network exploration does not start with an overview, but with interesting attributes and selections of interest. Likewise, the work of Chapter 3 starts with interesting attributes and the user is guided in the exploration with small multiples. Also, for the work presented in Chapter 6 on the extended massive sequence view we believe there is not a single best overview. The different reordering strategies all provide interesting overviews, showing multiple perspectives on the dynamic network. Each of the overviews enables the identification of distinct temporal and structural properties that are less apparent in the other overviews.

Earlier work by the author, not discussed in this dissertation, on the interactive construction and visualization of decision trees [272], also does not start with an overview but again with interesting attributes. These attributes are presented to the user and sorting here plays a crucial role to restrict inspection to only the most relevant attributes. Also, this interactive visualization adheres to the previous design principle of following the natural workflow of the user. This is achieved with a tight integration of automatic and manual decision making.

These examples all (automatically) support the presentation and explanation of findings to a broader audience. In the small multiples, large singles approach presentation and explanation is supported by automatically keeping a visual history. After exploration each finding can directly be explained by the path leading to it. Also, in the DOSA approach presentation of findings comes for free by creating a high-level overview of findings. In a similar fashion, the overview is interactively constructed by the user during exploration in the work on decision trees. This overview, without distracting details, can directly be used for presentation and visually explains the patterns.

Reflecting on the methods to achieve *automatic support for explanation and presentation* we can extract three techniques from the above implementations:

- **Automatic abstraction** – introduce a linked higher-level abstraction view in addition to the main view. This view should be updated automatically during the exploration and reflect the detail view on a higher aggregation level. Also, details should be kept to a minimum here to simplify presentation and reduce distraction from the main message to be conveyed.
- **Construction during exploration** – make the high-level overview part of the exploration process itself by interactive construction.
- **Automatic visual history** – keep a visual history of all important decisions taken during the exploration. This exploration trail provides a direct abstraction and supports the communication of findings by presenting an explanation.

8.2.3 Consistent interaction metaphors and visual variables

For all views and interaction widgets consistent interaction metaphors should be used to increase usability. Moreover, consistent visual variables should be used for all views to increase visual association.

If multiple views and controls are used in a (prototype) solution, all interaction techniques should be implemented consistently. If items can be highlighted and selected in one view, users expect these to also work for all other views. This means that users should be enabled to highlight and select the items in all other views as well, using the same interaction techniques. This also applies to more subtle interface element interactions. For example, if users are enabled to highlight an element and when clicked on some action is performed, then this is also expected for other elements that can be highlighted. If implemented poorly this leads to frustration and misunderstanding. In addition, consistent interface elements should be used for similar data-types, e.g., assuming the data types for two different variables are integer, we do not want a slider for one variable and a spin-box for the other. Next to consistent use of interaction metaphors, we think interaction controls should be kept to a minimum, *i.e.*, it is better to choose good default values and hide the controls, than to show them all and let the user define values. In order to enable users to change the (carefully chosen) default values, the controls should be shown on demand with for example, an *advanced* or *expert-mode*.

The use of color is a strong indicator things are linked to each other and whenever possible should be used to indicate this. For example, with the DOSA approach in Chapter 4 selections of interest are used as a central element. The selections of interest can be manipulated with many controls and interaction techniques across different views and panels. We use the visual variable color to indicate association. In the detail view the selection of interest is represented with a box that has a uniquely identifying color. The same color is used in the aggregated box in the high-level overview as well as in the selection view which is used to order the selections. Also, the handles of the scented widgets in the attribute view share this color. If a different selection of interest is manipulated the color of the handles is automatically updated to reflect this.

Furthermore, a consistent selection color should be used for all selections and unified across all views, *e.g.*, data items in a visualization, items in a list or table. This color harmony encourages a relaxed and clear well-ordered interface hence, improving the user experience and lowering the barrier for exploration.

The use of consistent interaction metaphors and visual variables seems like a no-brainer but is too often neglected or poorly implemented in practice.

8.3 Future Work

In addition to the suggestions for future work for the specific techniques, presented in earlier chapters, here we present global directions for future work. There are four underexplored areas that we believe deserve more attention: streaming network data; scalability in terms of time; evolving hierarchical node structure; and comparison of multiple dynamic multivariate networks. Below we discuss these and suggest how the methods and techniques presented in this dissertation can contribute.

Streaming network data In this dissertation and with dynamic multivariate network exploration in general, it is assumed that the data is complete and full information is known at the moment of analysis. How to deal with streaming network data is still an open question. What visualization and interaction techniques are most suitable to support the exploration and analysis of such data? How long should data be kept in the visualization, in memory, and how to store this? Should aggregations be shown, and how often should they be updated? These are challenging and as of yet unsolved problems. Related to this is computational efficiency, which becomes important in a streaming network data setting.

The extended massive sequence view can probably be adapted to support streaming data by adding new data at the right side of the visualization and simultaneously fade out old data at the left side. However, many problems need to be solved first for a viable solution, for example, how to deal with sudden bursts of data addition, and how to diminish or prevent the effects of change blindness.

From all techniques discussed in this dissertation the most suited technique is presented in Chapter 7: the reduction of snapshots to points. By using a linear dimensionality reduction technique such as PCA we can correctly position new snapshots in the projection view. However, if many points are added, the projection likely needs to be updated. In addition, all other challenges mentioned before also apply here.

Temporal scalability In contrast to network scalability in terms of nodes and edges, an under-explored area is scalability with respect to the number of timesteps in the dynamic network. It is not clear what visualization is best suited when the number of timesteps is large, *e.g.*, in the order of thousands or more. Clearly, animation does not work, because it needs to run for a long period which intensifies the tracking of changes. In addition, this also renders the comparison of

multiple moments in time impossible. Furthermore, here a timeline does not help due to the sheer amount of positions to check, compare, remember, and replay. Also, small multiples split on the time attribute are no solution here due to the minimal visualization space for each multiple.

The msv tends to get very elongated for a large number of timesteps. A solution here could be to divide time in multiple intervals and analyse and explore these in isolation. However, this introduces new problems such as how to compare the intervals with each other. The *circular* msv would emphasize events that occurred more recently, which might not be desirable. Also, the projection view in the reduction of snapshots to points technique only scales to an amount of points until clutter and heavy overdraw occur. This could potentially be solved by using clustering techniques or density-based rendering, but how to support this with intuitive interaction is non-trivial.

Dynamic hierarchical structure Current approaches for the exploration and analysis of dynamic multivariate networks sometimes take extra node information into account such as hierarchical structure. This hierarchy is generally assumed to be static, *i.e.*, it does not change over time. In real-world dynamic networks, this hierarchy possibly changes over time, for example the organization structure of a company changes as the company grows and goes through different phases (startup, growth, expansion, and mature business). Similar techniques for the exploration of dynamic networks, as presented in this dissertation, can also be applied to dynamic hierarchies, as this is a special type of network (tree). However, how to visualize both the evolution of the dynamic network as well as the changing hierarchical structure to enable simultaneous exploration, and analyse one in the context of the other, is unexplored in literature.

Comparison Traditionally, visualization and interaction techniques are developed for the exploration and analysis of a single dynamic multivariate network. The comparison of two (or multiple) dynamic multivariate networks for the entire timespan is rarely discussed in literature. However, comparison of networks helps in the understanding and identification of different temporal network types such as, only increasing or decreasing the number of edges over time, oscillating between multiple stable temporal states, *et cetera*. Different views on a single network might be extended to multiple views in the reordering techniques of the msv. Also, the *projection* view of the snapshot reduction technique could show multiple networks. The snapshots of the different networks should then be connected with their own connecting line. However, this would not scale to the comparison of a large number of networks. Furthermore, for the comparison of multiple networks it might not be best to show them both, but rather show the differences, *e.g.*, as in Crippa *et al.* [73].

In the introduction we argued that the interactive visualization of dynamic multivariate networks is a challenging problem. In this dissertation we have presented different methods and techniques that provide a solution. However, there are still many challenges left to be solved.

Bibliography

- [1] United States Census Bureau, 2010 Census Data. <https://www.census.gov/2010census/data/>, 2010. Accessed March, 2014. (page 57).
- [2] Crop Prospects and Food Situation. *Food and Agriculture Organization of United Nations*, 1 (March 2012). (page 87).
- [3] Centre for research on the epidemiology of disasters, em-dat, the international disaster database. <http://www.emdat.be/database>, 2013. Accessed on 1/02/2013. (page 87).
- [4] NOAA/FEWSNet. Famine Early Warning System Network. <http://www.cpc.ncep.noaa.gov/products/fews/africa/>, 2013. Accessed on 1/01/2013. (pages 84, 87).
- [5] SynerScope connecting the dots. <http://www.synerscope.com>, 2013. Accessed: 13/02/2013. (page 78).
- [6] Weather Underground. Historical Weather. <http://www.wunderground.com/history/>, 2013. [Online; accessed 2014-12-18]. (pages 83, 87).
- [7] I.R.S. Internal Revenue Service, SOI Tax Stats - County-to-County Migration Data Files. <http://www.irs.gov/uac/SOI-Tax-Stats-County-to-County-Migration-Data-Files>, 2014. Accessed March, 2014. (page 57).
- [8] SocioPatterns. <http://www.sociopatterns.org/datasets/>, 2015. [Online; accessed 2015-03-18]. (page 132).
- [9] ABELLO, J., HADLAK, S., SCHUMANN, H., AND SCHULZ, H.-J. A Modular Degree-of-Interest Specification for the Visual Analysis of Large Dynamic Networks. *IEEE Trans. Vis. Comput. Graphics* 20, 3 (2013), 337–350. (pages 23, 120).
- [10] ABELLO, J., AND VAN HAM, F. Matrix Zoom: A Visual Interface to Semi-External Graphs. In *Proc. IEEE Symp. Information Visualization* (2004), pp. 183–190. (page 16).
- [11] ADRIENKO, G., ADRIENKO, N., MLADENOV, M., MOCK, M., AND POLITZ, C. Identifying Place Histories from Activity Traces with an Eye to Parameter Impact. *IEEE Trans. Vis. Comput. Graphics* 18, 5 (May 2012), 675–688. (page 73).
- [12] ALIAKBARY, S., HABIBI, J., AND MOVAGHAR, A. Feature Extraction from Degree Distribution for Comparison and Analysis of Complex Networks. *The Computer Journal* (2015). (page 126).

- [13] AMAR, R. A., EAGAN, J., AND STASKO, J. T. Low-Level Components of Analytic Activity in Information Visualization. In *Proc. 9th Int. Conf. Information Visualization* (2005), p. 15. (page 38).
- [14] ANDRIENKO, G., ANDRIENKO, N., BREMM, S., SCHRECK, T., VON LANDESBERGER, T., BAK, P., AND KEIM, D. Space-in-time and Time-in-space Self-organizing Maps for Exploring Spatiotemporal Patterns. In *IEEE Eurographics Proc. Conf. Visualization* (2010), pp. 913–922. (page 120).
- [15] ANDRIENKO, G., ANDRIENKO, N., MLADENOV, M., MOCK, M., AND PÖLITZ, C. Discovering bits of place histories from people’s activity traces. In *Proc. IEEE Symp. Visual Analytics Science and Technology* (Oct. 2010), pp. 59–66. (page 73).
- [16] ANGIULLI, F., AND PIZZUTI, C. Fast Outlier Detection in High Dimensional Spaces. In *Principles of Data Mining and Knowledge Discovery*, T. Elomaa, H. Mannila, and H. Toivonen, Eds., vol. 2431 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2002, pp. 15–27. (page 137).
- [17] ARCHAMBAULT, D., ABELLO, J., KENNEDY, J., KOBOUROV, S., MA, K.-L., MIKSCH, S., MUELDER, C., AND TELEA, A. C. Temporal Multivariate Networks. In *Multivariate Network Visualization*, A. Kerren, H. C. Purchase, and M. O. Ward, Eds., vol. 8380 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014, pp. 151–174. (page 120).
- [18] ARCHAMBAULT, D., MUNZNER, T., AND AUBER, D. GrouseFlocks: Steerable Exploration of Graph Hierarchy Space. *IEEE Trans. Vis. Comput. Graphics* 14, 4 (2008), 900–913. (pages 17, 48).
- [19] ARCHAMBAULT, D., AND PURCHASE, H. C. The Mental Map and Memorability in Dynamic Graphs. In *Proc. IEEE Pacific Visualization Symp.* (Feb. 2012), pp. 89–96. (page 18).
- [20] BACH, B., FEKETE, J.-D., AND PIETRIGA, E. GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks. *IEEE Trans. Vis. Comput. Graphics* 20, 5 (2014), 740–754. (pages 18, 23, 119).
- [21] BACH, B., HENRY-RICHE, N., DWYER, T., MADHYASTHA, T., FEKETE, J.-D., AND GRABOWSKI, T. Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks. *Computer Graphics Forum* (2015). (pages 23, 120).
- [22] BACH, B., PIETRIGA, E., AND FEKETE, J.-D. Visualizing Dynamic Networks with Matrix Cubes. In *Proc. SIGCHI Conf. Human Factors in Computing Systems* (New York, NY, USA, 2014), CHI, ACM, pp. 877–886. (pages 19, 20, 23, 120).
- [23] BARUAH, R. D., AND ANGELOV, P. Evolving Social Network Analysis: a Case Study on Mobile Phone Data. In *Evolving and Adaptive Intelligent Systems (EAIS), 2012 IEEE Conference on* (May 2012), pp. 114–120. (page 73).
- [24] BATTISTA, G. D., EADES, P., TAMASSIA, R., AND TOLLIS, I. G. *Graph Drawing: Algorithms for the Visualization of Graphs*. An Alan R. Apt Book. Prentice Hall, 1999. (pages 14, 15, 48).

- [25] BAUR, M., AND SCHANK, T. Dynamic graph drawing in Visone. Tech. rep., Fakultät für Informatik, Universität Karlsruhe, 2008. (page 23).
- [26] BAVOIL, L., CALLAHAN, S. P., CROSSNO, P. J., FREIRE, J., SCHEIDEGGER, C. E., SILVA, C. T., AND VO, H. T. VisTrails: Enabling Interactive Multiple-View Visualizations. *IEEE Visualization* (2005), 135–142. (page 30).
- [27] BECK, F., BURCH, M., DIEHL, S., AND WEISKOPF, D. The State of the Art in Visualizing Dynamic Graphs. In *EuroVis - STARS* (2014), EuroVis, Eurographics Association, pp. 83–103. (pages 21, 22, 23, 119, 120).
- [28] BECK, F., BURCH, M., VEHLow, C., DIEHL, S., AND WEISKOPF, D. Rapid Serial Visual Presentation in Dynamic Graph Visualization. In *IEEE Proc. Symp. Visual Languages and Human-Centric Computing* (2012), pp. 185–192. (page 23).
- [29] BECKER, R. A., CLEVELAND, W. S., AND SHYU, M. J. The Visual Design and Control of Trellis Display. *J. Comp. Graph. Stat.* 5 (1996), 123–155. (page 30).
- [30] BERTIN, J. *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin Press, 1983. (pages 18, 30, 119).
- [31] BEZERIANOS, A., CHEVALIER, F., DRAGICEVIC, P., ELMQVIST, N., AND FEKETE, J.-D. GraphDice: A System for Exploring Multivariate Social Networks. In *Proc. 12th Eurographics IEEE Conf. Visualization* (2010), Eurographics Association, pp. 863–872. (pages 17, 48, 66).
- [32] BIER, E. A., STONE, M. C., PIER, K., BUXTON, W., AND DEROSE, T. D. Toolglass and Magic Lenses: The See-through Interface. In *Proc. 20th Annu. Conf. Computer Graphics and Interactive Techniques* (1993), ACM, pp. 73–80. (page 48).
- [33] BILGIN, C. C., AND YENER, B. Dynamic network evolution: Models, clustering, anomaly detection. *IEEE Networks* (2006). (page 137).
- [34] BLONDEL, V., ESCH, M., AND CHAN, C. Geofast. <http://www.geofast.net>. Accessed on 14/02/2013. (page 73).
- [35] BLONDEL, V. D., ESCH, M., CHAN, C., CLÉROT, F., DEVILLE, P., HUENS, E., MORLOT, F., SMOREDA, Z., AND ZIEMLIICKI, C. Data for Development: the D4D Challenge on Mobile Phone Data. *CoRR abs/1210.0137* (2012). (pages 70, 87).
- [36] BOAS, M., AND HUSER, A. Child labour and cocoa production in West Africa. The case of Côte d’Ivoire and Ghana. *Research Program on Trafficking and Child Labour. Fafo-report 522. Web edition* (2006). (page 83).
- [37] BOITMANIS, K., BRANDES, U., AND PICH, C. Visualizing Internet Evolution on the Autonomous Systems Level. In *Graph Drawing*, S.-H. Hong, T. Nishizeki, and W. Quan, Eds., vol. 4875 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 365–376. (page 18).
- [38] BORISJUK, L., REZA HAJIREZAEI, M., KLUKAS, C., ROLLETSCHEK, H., AND SCHREIBER, F. Integrating Data from Biological Experiments into Metabolic Networks with the DBE Information System. In *Silico Biology 5: 0011* (2005), p. 11. (pages 17, 48).

- [39] BOYANDIN, I., BERTINI, E., AND LALANNE, D. A Qualitative Study on the Exploration of Temporal Changes in Flow Maps with Animation and Small-Multiples. *Comput. Graph. Forum* 31 (2012), 1005–1014. (pages 19, 30).
- [40] BRANDES, U., AND CORMAN, S. R. Visual Unrolling of Network Evolution and the Analysis of Dynamic Discourse. *Information Visualization* 2, 1 (2003), 40–50. (page 23).
- [41] BRANDES, U., FLEISCHER, D., AND PUPPE, T. Dynamic Spectral Layout of Small Worlds. In *Graph Drawing* (2006), Springer, pp. 25–36. (page 23).
- [42] BRANDES, U., INDLEKOFER, N., AND MADER, M. Visualization Methods for Longitudinal Social Networks and Stochastic Actor-oriented Modeling. *Social Networks* 34, 3 (2012), 291–308. (page 18).
- [43] BRANDES, U., AND NICK, B. Asymmetric Relations in Longitudinal Social Networks. *IEEE Trans. Vis. Comput. Graphics* 17, 12 (2011), 2283–2290. (pages 20, 23).
- [44] BRANDES, U., AND WAGNER, D. A Bayesian Paradigm for Dynamic Graph Layout. In *Graph Drawing* (1997), Springer, pp. 236–247. (page 23).
- [45] BREMM, S., VON LANDESBERGER, T., HEß, M., AND FELLNER, D. PCDC - On the Highway to Data - A Tool for the Fast Generation of Large Synthetic Data Sets. In *Proc. Int. Workshop on Visual Analytics* (2012), pp. 7–11. (page 38).
- [46] BRULS, M., HUIZING, K., AND VAN WIJK, J. J. Squarified Treemaps. In *Proc. Joint Eurographics and IEEE TCVG Symp. Visualization* (1999), Press, pp. 33–42. (page 56).
- [47] BUJA, A., McDONALD, J. A., MICHALAK, J., AND STUETZLE, W. Interactive data visualization using focusing and linking. In *Proc. 2nd Conf. Visualization* (Oct. 1991), pp. 156–163. (page 12).
- [48] BURCH, M., BECK, F., AND DIEHL, S. Timeline Trees: Visualizing Sequences of Transactions in Information Hierarchies. In *Proc. Working Conf. Advanced Visual Interfaces* (New York, USA, 2008), ACM, pp. 75–82. (pages 20, 94, 120).
- [49] BURCH, M., BECK, F., AND WEISKOPF, D. Radial Edge Splatting for visualizing dynamic directed graphs. In *Proc. 4th Int. Conf. Information Visualization Theory App.* (2012), SciTePress, pp. 603–612. (page 23).
- [50] BURCH, M., AND DIEHL, S. TimeRadarTrees: Visualizing Dynamic Compound Digraphs. *Comput. Graph. Forum* 27, 3 (2008), 823–830. (pages 20, 23, 94, 120).
- [51] BURCH, M., FRITZ, M., BECK, F., AND DIEHL, S. TimeSpiderTrees: A Novel Visual Metaphor for Dynamic Compound Graphs. In *IEEE Symp. Visual Languages and Human-Centric Computing* (2010), pp. 168–175. (pages 20, 94, 120).
- [52] BURCH, M., HOFERLIN, M., AND WEISKOPF, D. Layered TimeRadarTrees. In *Proc. 15th Int. Conf. Information Visualisation* (2011), pp. 18–25. (pages 20, 23, 94, 120).
- [53] BURCH, M., SCHMIDT, B., AND WEISKOPF, D. A Matrix-Based Visualization for Exploring Dynamic Compound Digraphs. In *IEEE Proc. 17th Int. Conf. Information Visualisation* (2013), pp. 66–73. (pages 20, 23).

- [54] BURCH, M., VEHLow, C., BECK, F., DIEHL, S., AND WEISKOPF, D. Parallel Edge Splatting for Scalable Dynamic Graph Visualization. *IEEE Trans. Vis. Comput. Graphics* 17, 12 (2011), 2344–2353. (pages 18, 20, 23, 94, 120).
- [55] BURCH, M., AND WEISKOPF, D. Visualizing Dynamic Quantitative Data in Hierarchies - TimeEdgeTrees: Attaching Dynamic Weights to Tree Edges. In *Proc. Int. Conf. Information Vis. Theory App.* (2011), pp. 177–186. (pages 20, 94, 96, 120).
- [56] BURCH, M., AND WEISKOPF, D. A Flip-Book of Edge-Splatted Small Multiples for Visualizing Dynamic Graphs. In *Proc. 7th Int. Symp. Visual Information Communication and Interaction* (New York, NY, USA, 2014), ACM, pp. 29:29–29:38. (page 23).
- [57] CALLAHAN, S. P., FREIRE, J., SANTOS, E., SCHEIDEGGER, C. E., SILVA, C. T., AND VO, H. T. VisTrails: Visualization Meets Data Management. In *Proc. ACM Int. Conf. Manage. of data* (2006), SIGMOD, ACM, pp. 745–747. (page 30).
- [58] CARD, S. Information Visualization. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, J. A. Jacko and A. Sears, Eds. Lawrence Erlbaum Associates, 2007, pp. 544–582. (page 11).
- [59] CARD, S. K., MACKINLAY, J. D., AND SHNEIDERMAN, B., Eds. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. (pages 10, 11, 12).
- [60] CARROLL, R. Chocolate war erupts in Ivory Coast, *The Guardian* (14/05/2004). <http://www.guardian.co.uk/world/2004/may/14/rorycarroll>, 2004. Accessed on 1/02/2013. (page 83).
- [61] CERNÝ, V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J. of Optimization Theory and Applications* 45 (1985), 41–51. (page 100).
- [62] CHEN, C. Top 10 unsolved information visualization problems. *IEEE Comput. Graph. Appl.* 25, 4 (July 2005), 12–16. (page 11).
- [63] CHI, E. H.-H. Web Analysis Visualization Spreadsheet. In *WOWS* (1999), pp. 24–31. (page 30).
- [64] CHI, E. H.-H., BARRY, P., RIEDL, J., AND KONSTAN, J. A Spreadsheet Approach to Information Visualization. In *Proc. IEEE Symp. Inform. Visualization* (1997), IEEE Computer Society, pp. 17–24. (page 30).
- [65] CHI, E. H.-H., RIEDL, J., BARRY, P., AND KONSTAN, J. Principles for Information Visualization Spreadsheets. *IEEE Comput. Graph. Appl.* 18, 4 (1998), 30–38. (page 30).
- [66] COCKBURN, A., KARLSON, A., AND BEDERSON, B. B. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Comput. Surv.* 41, 1 (Jan. 2009), 1–31. (page 12).

- [67] COHEN, J. D. Drawing Graphs to Convey Proximity: An Incremental Arrangement Method. *ACM Trans. Comput.-Hum. Interact.* 4, 3 (Sep. 1997), 197–229. (page 14).
- [68] COHEN, R. F., DI BATTISTA, G., TAMASSIA, R., AND TOLLIS, I. G. Dynamic Graph Drawings: Trees, Series-Parallel Digraphs, and st-Digraphs. *SIAM Journal on Computing* 24, 5 (1995), 970–1001. (page 23).
- [69] COHEN, R. F., DI BATTISTA, G., TAMASSIA, R., TOLLIS, I. G., AND BERTOLAZZI, P. A Framework for Dynamic Graph Drawing. In *Proc. 8th Annu. Symp. Computational Geometry* (1992), ACM, pp. 261–270. (page 23).
- [70] COLLIAT, G. OLAP, Relational, and Multidimensional Database Systems. *SIGMOD Rec.* 25, 3 (1996), 64–69. (page 49).
- [71] CORNELISSEN, B., HOLTEN, D., ZAIDMAN, A., MOONEN, L., VAN WIJK, J. J., AND VAN DEURSEN, A. Understanding Execution Traces Using Massive Sequence and Circular Bundle Views. In *Proc. 15th IEEE Int. Conf. Program Comprehension* (2007), pp. 49–58. (pages 92, 94).
- [72] CORREA, C., CRNOVRSANIN, T., MUELDER, C., SHEN, Z., ARMSTRONG, R., SHEARER, J., AND MA, K.-L. Cell Phone Mini Challenge Award: Intuitive Social Network Graphs Visual Analytics of Cell Phone Data using Mobivis and Ontovis. In *Proc. IEEE Symp. Visual Analytics Science and Technology* (Oct. 2008), pp. 211–212. (page 73).
- [73] CRIPPA, A., MAURITS, N. M., LORIST, M. M., AND ROERDINK, J. B. Graph averaging as a means to compare multichannel EEG coherence networks and its application to the study of mental fatigue and neurodegenerative disease. *Comput. & Graphics* 35, 2 (2011), 265–274. (page 150).
- [74] DE PAUW, W., LORENZ, D., VLISSIDES, J., AND WEGMAN, M. Execution Patterns in Object-Oriented Visualization. In *Proc. 4th Conf. Object-Oriented Technologies and Systems* (Berkeley, CA, USA, 1998), USENIX Association, pp. 16–16. (page 95).
- [75] DIALLO, Y., AND MAX-PLANCK-INSTITUT FÜR ETHNOLOGISCHE FORSCHUNG. *Conflict, Cooperation and Integration: A West African Example (Côte D’Ivoire)*. Max Planck Institute for Social Anthropology working papers. Max Planck Inst. for Social Anthropology, 2001. (page 85).
- [76] DIEHL, S., AND GÖRG, C. *Graphs, They Are Changing*, vol. 2528. Springer-Verlag, 2002, pp. 23–30. (pages 18, 23, 94, 120).
- [77] DIEHL, S., GÖRG, C., AND KERREN, A. Preserving the mental map using foresighted layout. In *IEEE Proc. 3rd Conf. Visualization* (2001), Eurographics Association, pp. 175–184. (page 23).
- [78] DIJKSTRA, E. W. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* 1, 1 (1959), 269–271. (page 55).
- [79] DINKLA, K., WESTENBERG, M. A., AND VAN WIJK, J. J. Compressed Adjacency Matrices: Untangling Gene Regulatory Networks. *IEEE Trans. Vis. Comput. Graphics* 18, 12 (Dec. 2012), 2457–2466. (page 16).

- [80] DIX, A. Introduction to Information Visualisation. In *Information Retrieval Meets Information Visualization*, M. Agosti, N. Ferro, P. Forner, H. Müller, and G. Santucci, Eds., vol. 7757 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pp. 1–27. (page 11).
- [81] DUNNE, C., HENRY RICHE, N., LEE, B., METOYER, R., AND ROBERTSON, G. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History. In *Proc. SIGCHI Conf. Human Factors in Computing Systems* (2012), ACM, pp. 1663–1672. (page 49).
- [82] DUNNE, C., AND SHNEIDERMAN, B. Motif Simplification: Improving Network Visualization Readability with Fan, Connector, and Clique Glyphs. In *Proc. SIGCHI Conf. Human Factors in Computing Systems* (2013), ACM, pp. 3247–3256. (page 48).
- [83] DWYER, T., AND EADES, P. Visualising a Fund Manager Flow Graph with Columns and Worms. In *Proc. 6th Int. Conf. Information Vis.* (2002), pp. 147–152. (pages 19, 23, 120).
- [84] EAGLE, N., AND PENTLAND, A. Eigenbehaviors: Identifying Structure in Routine. *Behavioral Ecology and Sociobiology* 63 (May 2009), 1057–1066. (page 73).
- [85] EICK, S. G., AND WARD, A. An Interactive Visualization for Message Sequence Charts. In *Proc. 4th Workshop Program Comprehension* (1996), pp. 2–8. (page 94).
- [86] ELMQVIST, N., DO, T.-N., GOODELL, H., HENRY, N., AND FEKETE, J.-D. ZAME: Interactive Large-Scale Graph Visualization. In *IEEE Proc. Symp. Pacific Visualization* (March 2008), pp. 215–222. (page 16).
- [87] ELMQVIST, N., DRAGICEVIC, P., AND FEKETE, J.-D. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Trans. Vis. Comput. Graphics* 14, 6 (2008), 1141–1148. (page 66).
- [88] ELMQVIST, N., HENRY, N., RICHE, Y., AND FEKETE, J.-D. Mélange: Space Folding for Multi-focus Interaction. In *Proc. SIGCHI Conf. Human Factors in Computing Systems* (New York, NY, USA, 2008), ACM, pp. 1333–1342. (page 16).
- [89] ELMQVIST, N., STASKO, J. T., AND TSIGAS, P. DataMeadow: A Visual Canvas for Analysis of Large-Scale Multivariate Data. In *Proc. IEEE Symp. Visual Analytics Science and Technology* (2007), pp. 187–194. (page 51).
- [90] ERTEN, C., HARDING, P. J., KOBOUROV, S. G., WAMPLER, K., AND YEE, G. GraphAEL: Graph Animations with Evolving Layouts. In *Graph Drawing*, G. Liotta, Ed., vol. 2912 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 98–110. (pages 18, 23).
- [91] ERTEN, C., KOBOUROV, S. G., LE, V., AND NAVABI, A. Simultaneous graph drawing: Layout algorithms and visualization schemes. In *Graph Drawing* (2004), Springer, pp. 437–449. (page 23).
- [92] FALKOWSKI, T., BARTELHEIMER, B., AND SPILIOPOULOU, M. Mining and Visualizing the Evolution of Subgroups in Social Networks. In *IEEE/WIC/ACM Int. Conf. Web Intelligence* (Dec. 2006), pp. 52–58. (page 120).

- [93] FARRUGIA, M., HURLEY, N., AND QUIGLEY, A. Exploring Temporal Ego Networks Using Small Multiples and Tree-ring Layouts. In *Proc. 4th Int. Conf. Advances in Human Computer Interfaces* (2011). (pages 18, 23, 94, 120).
- [94] FEDERICO, P., AIGNER, W., MIKSCH, S., WINDHAGER, F., AND SMUC, M. Vertigo Zoom: Combining Relational and Temporal Perspectives on Dynamic Networks. In *Proc. Work. Conf. Advanced Visual Interfaces* (New York, NY, USA, 2012), ACM, pp. 437–440. (page 19).
- [95] FEDERICO, P., AIGNER, W., MIKSCH, S., WINDHAGER, F., AND ZENK, L. A Visual Analytics Approach to Dynamic Social Networks. In *Proc. 11th Int. Conf. Knowledge Management and Knowledge Technologies* (New York, NY, USA, 2011), ACM, pp. 1–8. (pages 18, 19, 23, 94, 120).
- [96] FEDERICO, P., PFEFFER, J., AIGNER, W., MIKSCH, S., AND ZENK, L. Visual Analysis of Dynamic Networks Using Change Centrality. In *IEEE/ACM Int. Conf. Advances Social Networks Analysis and Mining* (Aug. 2012), pp. 179–183. (pages 19, 20).
- [97] FEKETE, J.-D., WANG, D., DANG, N., AND PLAISANT, C. Overlaying Graph Links on Treemaps. *IEEE Symp. Information Visualization Conf. Compendium (demonstration)*, 2003. (pages 17, 48).
- [98] FENG, K.-C., WANG, C., SHEN, H.-W., AND LEE, T.-Y. Coherent Time-Varying Graph Drawing with Multifocus+Context Interaction. *IEEE Trans. Vis. Comput. Graphics* 18, 8 (2012), 1330–1342. (page 23).
- [99] FORRESTER, D., KOBOUROV, S. G., NAVABI, A., WAMPLER, K., AND YEE, G. V. graphael: A system for generalized force-directed layouts. In *Graph drawing* (2005), Springer, pp. 454–464. (page 23).
- [100] FOURNET, J., AND BARRAT, A. Contact Patterns among High School Students. *PLoS ONE* 9, 9 (Sep. 2014), e107878. (pages 123, 124, 132, 136).
- [101] FRASINCAR, F., TELEA, A. C., AND HOUBEN, G.-J. Adapting Graph Visualization Techniques for the Visualization of RDF Data. In *Visualizing the Semantic Web*, V. Geroimenko and C. Chen, Eds. Springer London, 2006, pp. 154–171. (pages 17, 48).
- [102] FREEDMAN, D., AND DIACONIS, P. On the Histogram as a Density Estimator: L₂ Theory. *Probab. Theory Related Fields* 57, 4 (1981), 453–476. (page 37).
- [103] FRIEDRICH, C., AND EADES, P. The Marey graph animation tool demo. In *Graph Drawing* (2001), Springer, pp. 396–406. (page 23).
- [104] FRIEDRICH, C., AND EADES, P. Graph drawing in motion. *Journal of Graph Algorithms and Applications* 6, 3 (2002), 353–370. (page 23).
- [105] FRIEDRICH, C., AND HOULE, M. E. Graph drawing in motion II. In *Graph Drawing* (2002), Springer, pp. 220–231. (page 23).
- [106] FRISHMAN, Y., AND TAL, A. Dynamic Drawing of Clustered Graphs. In *Proc. IEEE Symp. Information Visualization* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 191–198. (pages 18, 23, 94, 119).

- [107] FRISHMAN, Y., AND TAL, A. Dynamic Drawing of Clustered Graphs. In *IEEE Symp. Information Visualization* (2004), pp. 191–198. (page 18).
- [108] FRISHMAN, Y., AND TAL, A. Online Dynamic Graph Drawing. *IEEE Trans. Vis. Comput. Graphics* 14, 4 (2008), 727–740. (pages 18, 23).
- [109] FRUCHTERMAN, T. M. J., AND REINGOLD, E. M. Graph Drawing by Force-directed Placement. *Softw. Pract. Exper.* 21, 11 (Nov. 1991), 1129–1164. (page 14).
- [110] GAERTLER, M., AND WAGNER, D. A hybrid model for drawing dynamic and evolving graphs. In *Graph Drawing* (2006), Springer, pp. 189–200. (page 23).
- [111] GAJER, P., AND KOBOUROV, S. G. GRIP: Graph dRawing with Intelligent Placement. In *Proc. 8th Int. Symp. Graph Drawing* (London, UK, 2001), Springer-Verlag, pp. 222–228. (page 14).
- [112] GANSNER, E. R., AND NORTH, S. C. An Open Graph Visualization System and its Applications to Software Engineering. *Software - Practice and Experience* 30, 11 (2000), 1203–1233. (pages 57, 128).
- [113] GAREY, M. R., JOHNSON, D. S., AND STOCKMEYER, L. Some Simplified NP-Complete Problems. In *Proc. 6th Annu. ACM Symp. Theory of computing* (New York, NY, USA, 1974), ACM, pp. 47–63. (page 100).
- [114] GHANI, S., KWON, B. C., LEE, S., YI, J. S., AND ELMQVIST, N. Visual Analytics for Multimodal Social Network Analysis: A Design Study with Social Scientists. *IEEE Trans. Vis. Comput. Graphics* 19, 12 (2013), 2032–2041. (page 49).
- [115] GHONIEM, M., FEKETE, J.-D., AND CASTAGLIOLA, P. On the Readability of Graphs Using Node-link and Matrix-based Representations: A Controlled Experiment and Statistical Analysis. *Information Visualization* 4, 2 (Jul. 2005), 114–135. (page 16).
- [116] GOLDSTEIN, E. B. Visual Attention. In *Sensation and Perception*. Wadsworth Cengage Learning, 2009, ch. 9. (page 100).
- [117] GOROCHOWSKI, T. E., DI BERNARDO, M., AND GRIERSON, C. S. Using Aging to Visually Uncover Evolutionary Processes on Networks. *IEEE Trans. Vis. Comput. Graphics* 18, 8 (2012), 1343–1352. (page 23).
- [118] GOVE, R., GRAMSKY, N., KIRBY, R., SEFER, E., SOPAN, A., DUNNE, C., SHNEIDERMAN, B., AND TAIEB-MAIMON, M. NetVisia: Heat Map & Matrix Visualization of Dynamic Social Network Statistics & Content. In *Proc. 3th Int. Conf. Social Computing* (2011), pp. 19–26. (pages 18, 94, 120).
- [119] GREILICH, M., BURCH, M., AND DIEHL, S. Visualizing the Evolution of Compound Digraphs with TimeArcTrees. *Comput. Graph. Forum* 28, 3 (2009), 975–982. (pages 20, 23, 94, 120).
- [120] GROH, G., HANSTEIN, H., AND WÖRNDL, W. Interactively Visualizing Dynamic Social Networks with DySoN. In *Proc. Work. Visual Interfaces to the Social and the Semantic Web* (2009), VISSW. (pages 19, 23, 120).

- [121] GUO, D., CHEN, J., MACEachREN, A. M., AND LIAO, K. A Visualization System for Space-Time and Multivariate Patterns (VIS-STAMP). *IEEE Trans. Vis. Comput. Graphics* 12, 6 (2006), 1461–1474. (page 30).
- [122] GÖRG, C., BIRKE, P., POHL, M., AND DIEHL, S. Dynamic Graph Drawing of Sequences of Orthogonal and Hierarchical Graphs. In *Graph Drawing*, J. Pach, Ed., vol. 3383 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 228–238. (pages 18, 23).
- [123] HACHUL, S. *A potential field based multilevel algorithm for drawing large graphs*. PhD thesis, University of Cologne, 2005. (page 14).
- [124] HADLAK, S., SCHULZ, H.-J., AND SCHUMANN, H. In Situ Exploration of Large Dynamic Networks. *IEEE Trans. Vis. Comput. Graphics* 17, 12 (2011), 2334–2343. (pages 20, 23).
- [125] HADLAK, S., SCHUMANN, H., CAP, C. H., AND WOLLENBERG, T. Supporting the Visual Analysis of Dynamic Networks by Clustering associated Temporal Attributes. *IEEE Trans. Vis. Comput. Graphics* 19, 12 (2013), 2267–2276. (page 120).
- [126] HAIMES, Y. Y., LASON, L. S., AND WISMER, D. A. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE Trans. Syst., Man, Cybern., Syst.* 1, 3 (1971), 296–297. (page 104).
- [127] HALKO, N., MARTINSSON, P.-G., AND TROPP, J. A. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *ArXiv e-prints* (Sep. 2009). (page 125).
- [128] HASTINGS, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57 (1970), 97–109. (page 104).
- [129] HAYASHI, A., MATSUBAYASHI, T., HOSHIDE, T., AND UCHIYAMA, T. Initial Positioning Method for Online and Real-Time Dynamic Graph Drawing of Time Varying Data. In *Proc. 17th Int. Conf. Information Visualisation* (2013), pp. 435–444. (page 23).
- [130] HEER, J., MACKINLAY, J., STOLTE, C., AND AGRAWALA, M. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Trans. Vis. Comput. Graphics* 14, 6 (2008), 1189–1196. (page 33).
- [131] HEER, J., AND SHNEIDERMAN, B. Interactive Dynamics for Visual Analysis. *Commun. ACM* 55, 4 (2012), 45–54. (page 30).
- [132] HENRY, N., AND FEKETE, J.-D. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Trans. Vis. Comput. Graphics* 12, 5 (Sep. 2006), 677–684. (page 16).
- [133] HENRY, N., FEKETE, J.-D., AND MCGUFFIN, M. J. NodeTriX: a Hybrid Visualization of Social Networks. *IEEE Trans. Vis. Comput. Graphics* 13, 6 (Nov. 2007), 1302–1309. (page 16).
- [134] HERMAN, I., MELANCON, G., AND MARSHALL, M. S. Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Trans. Vis. Comput. Graphics* 6, 1 (2000), 24–43. (page 49).

- [135] HLAWATSCH, M., BURCH, M., AND WEISKOPF, D. Visual Adjacency Lists for Dynamic Graphs. *IEEE Trans. Vis. Comput. Graphics* 20, 11 (Nov. 2014), 1590–1603. (pages 16, 18, 20, 22, 23).
- [136] HÖFERLIN, B., HÖFERLIN, M., AND RÄUCHLE, J. Visual Analytics of Mobile Data. In *Proc. Workshop Nokia Mobile Data Challenge* (2012). (page 73).
- [137] HOLTEN, D. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Trans. Vis. Comput. Graphics* 12, 5 (2006), 741–748. (pages 16, 51, 136).
- [138] HOLTEN, D., CORNELISSEN, B., AND VAN WIJK, J. J. Trace Visualization Using Hierarchical Edge Bundles and Massive Sequence Views. In *Proc. 4th IEEE Int. Workshop Visualizing Software for Understanding and Analysis* (2007), pp. 47–54. (pages 23, 92, 93, 95, 99, 107, 111).
- [139] HOLTEN, D., AND VAN WIJK, J. J. Visual Comparison of Hierarchically Organized Data. *Comput. Graph. Forum* 27, 3 (2008), 759–766. (page 107).
- [140] HOLTEN, D., AND VAN WIJK, J. J. Force-directed Edge Bundling for Graph Visualization. In *IEEE Proc. 11th Conf. Visualization* (Chichester, UK, 2009), The Eurographs Association & John Wiley & Sons, Ltd., pp. 983–998. (pages 15, 16).
- [141] HOTELLING, H. Analysis of a Complex of Statistical Variables into Principal Components. *J. Educ. Psych.* 24 (1933). (page 125).
- [142] HU, Y., KOBouroV, S. G., AND VEERAMONI, S. Embedding, Clustering and Coloring for Dynamic Maps. In *IEEE Proc. Symp. Pacific Visualization* (2012), IEEE Computer Society, pp. 33–40. (page 23).
- [143] HUANG, M. L., EADES, P., AND WANG, J. On-line animated visualization of huge graphs using a modified spring algorithm. *Journal of Visual Languages & Computing* 9, 6 (1998), 623–645. (page 23).
- [144] HURTER, C., ERSOY, O., FABRIKANT, S. I., KLEIN, T., AND TELEA, A. Bundled Visualization of Dynamic Graph and Trail Data. *IEEE Trans. Vis. Comput. Graphics* (2013). (page 23).
- [145] HURTER, C., TISSOIRES, B., AND CONVERSY, S. FromDaDy: Spreading Aircraft Trajectories Across Views to Support Iterative Queries. *IEEE Trans. Vis. Comput. Graphics* 15, 6 (2009), 1017–1024. (page 48).
- [146] INSELBERG, A. The Plane with Parallel Coordinates. *The Visual Computer* 1, 2 (1985), 69–91. (page 56).
- [147] INTERNATIONAL COCOA ORGANIZATION, I. C. C. O. Annual Report 2010/2011. *International Cocoa Organization* (2012). (page 83).
- [148] INTERNATIONAL CRISIS GROUP, I. C. G. A Critical Period for Ensuring Stability in Côte d’Ivoire. *Africa Report* 176 (1 August 2011). (page 79).
- [149] INTERNATIONAL CRISIS GROUP, I. C. G. Côte d’Ivoire: Is war the only option? *Africa Report* 171 (3 March 2011). (page 79).

- [150] INTERNATIONAL CRISIS GROUP, I. C. G. Côte d'Ivoire: Defusing Tensions. *Africa Report* 193 (26 November 2012). (page 79).
- [151] ITOH, M., TOYODA, M., AND KITSUREGAWA, M. An Interactive Visualization Framework for Time-Series of Web Graphs in a 3D Environment. In *IEEE Proc. 14th Int. Conf. Information Visualisation* (2010), pp. 54–60. (page 23).
- [152] ITU-T. *Recommendation Z.120: Message Sequence Chart (MSC)*. ITU-T, Geneva, 1993. (pages 92, 95).
- [153] IZENMAN, A. J. Recent Developments in Nonparametric Density Estimation. *J. Amer. Statist. Assoc.* 86, 413 (1991), 205–224. (page 37).
- [154] JACCARD, P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles* 37 (1901), 547–579. (page 75).
- [155] JANKUN-KELLY, T. J., KREYLOS, O., MA, K.-L., HAMANN, B., JOY, K. I., SHALF, J., AND BETHEL, E. W. Deploying Web-Based Visual Exploration Tools on the Grid. *IEEE Comput. Graph. Appl.* 23, 2 (2003), 40–50. (page 30).
- [156] JANKUN-KELLY, T. J., AND MA, K.-L. A Spreadsheet Interface for Visualization Exploration. In *IEEE Visualization* (2000), pp. 69–76. (page 30).
- [157] JANKUN-KELLY, T. J., AND MA, K.-L. Visualization Exploration and Encapsulation via a Spreadsheet-Like Interface. *IEEE Trans. Vis. Comput. Graphics* 7, 3 (2001), 275–287. (page 30).
- [158] JERDING, D. F., AND STASKO, J. T. The information mural: a technique for displaying and navigating large information spaces. *IEEE Symp. Information Visualization* 0 (1995), 43–50. (pages 92, 93).
- [159] JERDING, D. F., AND STASKO, J. T. The Information Mural: a technique for displaying and navigating large information spaces. *IEEE Trans. Vis. Comput. Graphics* 4, 3 (1998), 257–271. (pages 92, 93, 94).
- [160] JERDING, D. F., STASKO, J. T., AND BALL, T. Visualizing Message Patterns in Object-Oriented Program Executions. Tech. rep., Georgia Institute of Technology, 1996. GIT-GVU-96-15. (page 93).
- [161] JERDING, D. F., STASKO, J. T., AND BALL, T. Visualizing Interactions in Program Executions. In *Proc. 19th Int. Conf. Software Engineering* (New York, NY, USA, 1997), ACM, pp. 360–370. (page 94).
- [162] JUSUFI, I., DINGJIE, Y., AND KERREN, A. The Network Lens: Interactive Exploration of Multivariate Networks Using Visual Filtering. In *Proc. 14th Int. Conf. Information Visualisation* (2010), pp. 35–42. (page 48).
- [163] JUSUFI, I., KERREN, A., AND ZIMMER, B. Multivariate Network Exploration with JauntyNets. In *Proc. 17th Int. Conf. Information Visualisation* (2013), pp. 19–27. (pages 17, 48).
- [164] KAMADA, T., AND KAWAI, S. An Algorithm for Drawing General Undirected Graphs. *Inf. Process. Lett.* 31, 1 (Apr. 1989), 7–15. (page 14).

- [165] KEIM, D. A. Information Visualization and Visual Data Mining. *IEEE Trans. Vis. Comput. Graphics* 8, 1 (Jan. 2002), 1–8. (page 12).
- [166] KEIM, D. A., KOHLHAMMER, J., ELLIS, G., AND MANSMANN, F., Eds. *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, Nov. 2010. (page 13).
- [167] KEIM, D. A., MANSMANN, F., SCHNEIDEWIND, J., THOMAS, J., AND ZIEGLER, H. Visual Analytics: Scope and Challenges. In *Visual Data Mining*, S. J. Simoff, M. H. Böhlen, and A. Mazeika, Eds. Springer-Verlag, Berlin, Heidelberg, 2008, pp. 76–90. (pages 14, 71).
- [168] KEIM, D. A., MANSMANN, F., SCHNEIDEWIND, J., AND ZIEGLER, H. Challenges in Visual Data Analysis. In *Proc. Conf. Information Visualization* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 9–16. (pages 11, 71).
- [169] KEIM, D. A., SCHNEIDEWIND, J., AND SIPS, M. CircleView: a New Approach for Visualizing Time-Related Multidimensional Data Sets. In *Proc. Work. Conf. Advanced Visual Interfaces* (2004), ACM, pp. 179–182. (pages 108, 114, 142).
- [170] KERRACHER, N., KENNEDY, J., AND CHALMERS, K. The Design Space of Temporal Graph Visualisation. In *EuroVis - Short Papers* (2014), N. Elmqvist, M. Hlawitschka, and J. Kennedy, Eds., The Eurographics Association. (page 120).
- [171] KERREN, A., PURCHASE, H. C., AND WARD, M. O., Eds. *Multivariate Network Visualization*. No. 8380 in Lecture Notes in Computer Science. Springer International Publishing, 2014. (page 49).
- [172] KIRKPATRICK, S., GELATT, C. D., AND VECCHI, M. P. Optimization by Simulated Annealing. *Science* 220, 4598 (1983), 671–680. (page 100).
- [173] KOBOUROV, S. G. Spring Embedders and Force Directed Graph Drawing Algorithms. *CoRR abs/1201.3011* (2012). (page 14).
- [174] KOSSINETIS, G., AND WATTS, D. J. Empirical Analysis of an Evolving Social Network. *Science* 311, 5757 (2006), 88–90. (page 126).
- [175] KRINGS, G., CALABRESE, F., RATTI, C., AND BLONDEL, V. D. Urban Gravity: a Model for Inter-City Telecommunication Flows. *Journal of Statistical Mechanics: Theory and Experiment* 2009, 7 (2009), Lo7003. (page 73).
- [176] KRUSKAL, J. B. Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis. *Psychometrika* 29, 1 (1964), 1–27. (page 125).
- [177] KUMAR, G., AND GARLAND, M. Visual Exploration of Complex Time-Varying Graphs. *IEEE Trans. Vis. Comput. Graphics* 12, 5 (2006), 805–812. (pages 18, 23, 94, 119).
- [178] KWAN, M.-P., AND LEE, J. Geovisualization of Human Activity Patterns Using 3D GIS : A Time-Geographic Approach In Michael F . Goodchild and Donald G . Janelle . Eds . 2003 . *Spatially integrated social science* 27 (2003), 48–66. (page 73).
- [179] LEE, Y.-Y., LIN, C.-C., AND YEN, H.-C. Mental map preserving graph drawing using simulated annealing. In *Proc. Asia-Pacific Symp. Information Visualisation* (2006), Australian Computer Society, Inc., pp. 179–188. (page 23).

- [180] LIIV, I. Seriation and Matrix Reordering Methods: An Historical Overview. *Statistical Analysis and Data Mining* 3, 2 (2010), 70–91. (pages 93, 119, 136).
- [181] LIN, Y.-R., SUN, J., CAO, N., AND LIU, S. ContexTour: Contextual contour visual analysis on dynamic multi-relational clustering. In *Proc. Int. Conf. Data Mining* (2010), pp. 418–429. (page 23).
- [182] LOUBIER, E., AND DOUSSET, B. Temporal and Relational Data Representation by Graph Morphing. In *Safety and Reliability for managing Risk* (Feb. 2008), vol. 14, European Safety and Reliability Conference. (pages 20, 23).
- [183] LU, L. F., HUANG, M. L., AND HUANG, T.-H. A New Axes Re-ordering Method in Parallel Coordinates Visualization. In *Proc. 11th Int. Conf. Machine Learning and Applications* (2012), vol. 2, pp. 252–257. (page 93).
- [184] LUNZER, A., BELLEMAN, R., MELIS, P., AND STAMATAKOS, G. Preparing, Exploring and Comparing Cancer Simulation Results within a Large Parameter Space. In *14th Int. Conf. Inform. Visualisation* (2010), pp. 258–264. (page 30).
- [185] LYONS, K. A. Cluster Busting in Anchored Graph Drawing. In *Proc. Conf. Centre Advanced Studies on Collaborative Research* (1992), IBM Press, pp. 7–17. (page 18).
- [186] MACEACHREN, A., DAI, X., HARDISTY, F., GUO, D., AND LENGERICH, G. Exploring High-D Spaces with Multiform Matrices and Small Multiples. In *Proc. 9th Annu. IEEE Conf. Inform. Visualization* (2003), pp. 31–38. (page 30).
- [187] MACKINLAY, J. D., HANRAHAN, P., AND STOLTE, C. Show Me: Automatic Presentation for Visual Analysis. *IEEE Trans. Vis. Comput. Graphics* 13, 6 (2007), 1137–1144. (page 30).
- [188] MADAN, A., CEBRIÁN, M., MOTURU, S. T., FARRAHI, K., AND PENTLAND, A. Sensing the “Health State” of a Community. *IEEE Pervasive Computing* 11, 4 (2012), 36–45. (page 108).
- [189] MÄKINEN, E., AND SIIRTOLA, H. Reordering the Reorderable Matrix as an Algorithmic Problem. In *Proc. 1st Int. Conf. Theory and Application of Diagrams* (2000), pp. 453–467. (page 99).
- [190] MARKS, J., ANDALMAN, B., BEARDSLEY, P. A., FREEMAN, W., GIBSON, S., HODGINS, J., KANG, T., MIRTICH, B., PFISTER, H., RUMI, W., RYALL, K., SEIMS, J., AND SHIEBER, S. Design Galleries: a General Approach to Setting Parameters for Computer Graphics and Animation. In *Proc. 24th Annu. Conf. Comput. Graph. and Interactive Techniques* (1997), SIGGRAPH, ACM Press/Addison-Wesley Publishing Co., pp. 389–400. (page 30).
- [191] MARTIN, A. R., AND WARD, M. O. High Dimensional Brushing for Interactive Exploration of Multivariate Data. In *Proc. IEEE Conf. Visualization* (1995), p. 271. (page 51).
- [192] MASHIMA, D., KOBOUROV, S. G., AND HU, Y. Visualizing Dynamic Data with Maps. *IEEE Trans. Vis. Comput. Graphics* 18, 9 (2012), 1424–1437. (page 23).

- [193] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., AND TELLER, E. Equation of State Calculations by Fast Computing Machines. *J. of Chemical Physics* 21 (1953), 1087–1092. (page 104).
- [194] MISUE, K., EADES, P., LAI, W., AND SUGIYAMA, K. Layout Adjustment and the Mental Map. *Journal of Visual Languages & Computing* 6, 2 (1995), 183–210. (page 23).
- [195] MUNZNER, T. *Visualization Analysis and Design*. A.K. Peters visualization series. A K Peters, 2014. (page 10).
- [196] NESBITT, K. V., AND FRIEDRICH, C. Applying Gestalt Principles to Animated Visualizations of Network Data. In *IEEE Proc. 6th Int. Conf. Information Visualisation* (2002), pp. 737–743. (page 23).
- [197] NEUMANN, P., SCHLECHTWEG, S., AND CARPENDALE, M. S. T. ArcTrees: Visualizing Relations in Hierarchical Data. In *EuroVis* (2005), Eurographics Association, pp. 53–60. (page 93).
- [198] NEWMAN, M. E. J., AND WATTS, D. J. Renormalization Group Analysis of the Small-World Network Model. *Physics Letters A* 263, 4–6 (1999), 341–346. (page 129).
- [199] NOACK, A. An Energy Model for Visual Graph Clustering. In *Graph Drawing*, G. Liotta, Ed., vol. 2912 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 425–436. (page 14).
- [200] NORMAN, D. A. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA, 2002. (page 12).
- [201] NORTH, S. C. Incremental layout in DynaDAG. In *Graph Drawing*, F. Brandenburg, Ed., vol. 1027 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1996, pp. 409–418. (pages 18, 23).
- [202] NOWELL, L., HETZLER, E., AND TANASSE, T. Change blindness in information visualization: a case study. In *Proc. IEEE Symp. Information Visualization* (2001), pp. 15–22. (page 19).
- [203] PEARSON, K. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine Series* 6 2, 11 (1901), 559–572. (page 125).
- [204] PERER, A., AND SHNEIDERMAN, B. Balancing Systematic and Flexible Exploration of Social Networks. *IEEE Trans. Vis. Comput. Graphics* 12, 5 (2006), 693–700. (pages 17, 48).
- [205] PERLIN, K., AND FOX, D. Pad: An Alternative Approach to the Computer Interface. In *Proc. 20th Annu. Conf. Computer Graphics and Interactive Techniques* (New York, NY, USA, 1993), ACM, pp. 57–64. (page 12).
- [206] POHL, M., AND BIRKE, P. Interactive Exploration of Large Dynamic Networks. In *Proc. 10th Int. Conf. Visual Information Systems* (2008), Springer-Verlag, pp. 56–67. (page 23).
- [207] POHL, M., REITZ, F., AND BIRKE, P. As Time Goes By: Integrated Visualization and Analysis of Dynamic Networks. In *Proc. Working Conf. Advanced Vis. Interfaces* (2008), ACM, pp. 372–375. (pages 94, 120).

- [208] PRETORIUS, A. J., AND VAN WIJK, J. J. Visual Inspection of Multivariate Graphs. *Comput. Graph. Forum* 27, 3 (2008), 967–974. (page 49).
- [209] PURCHASE, H., ANDRIENKO, N., JANKUN-KELLY, T., AND WARD, M. Theoretical Foundations of Information Visualization. In *Information Visualization*, A. Kerren, J. Stasko, J.-D. Fekete, and C. North, Eds., vol. 4950 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 46–64. (page 11).
- [210] PURCHASE, H., AND SAMRA, A. Extremes Are Better: Investigating Mental Map Preservation in Dynamic Graphs. In *Diagrammatic Representation and Inference*, G. Stapleton, J. Howse, and J. Lee, Eds., vol. 5223 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 60–73. (page 18).
- [211] RAM, D. J., SREENIVAS, T. H., AND SUBRAMANIAM, K. G. Parallel Simulated Annealing Algorithms. *J. Parallel Distrib. Comput.* 37, 2 (1996), 207–212. (page 113).
- [212] RAO, R., AND CARD, S. K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. In *Proc. SIGCHI Conf. Human factors computing systems: celebrating interdependence* (1994), CHI, ACM, pp. 318–322. (page 30).
- [213] REITZ, F. A Framework for an Ego-Centered and Time-Aware Visualization of Relations in Arbitrary Data Repositories. *arXiv preprint arXiv:1009.5183* (2010). (page 23).
- [214] REITZ, F., POHL, M., AND DIEHL, S. Focused Animation of Dynamic Compound Graphs. In *Proc. 13th Int. Conf. Information Visualisation* (2009), pp. 679–684. (pages 18, 23, 94, 119).
- [215] RENIERIS, M., AND REISS, S. P. Almost: Exploring Program Traces. In *Proc. 8th Int. Conf. Information and Knowledge Management* (New York, NY, USA, 1999), ACM, pp. 70–77. (page 95).
- [216] ROBERTSON, G., FERNANDEZ, R., FISHER, D., LEE, B., AND STASKO, J. T. Effectiveness of Animation in Trend Visualization. *IEEE Trans. Vis. Comput. Graphics* 14, 6 (2008), 1325–1332. (pages 18, 94, 120).
- [217] ROKHLIN, V., SZLAM, A., AND TYGERT, M. A Randomized Algorithm for Principal Component Analysis. *SIAM J. Matrix Anal. Appl.* 31, 3 (Aug. 2009), 1100–1124. (page 125).
- [218] ROSVALL, M., AND BERGSTROM, C. T. Mapping Change in Large Networks. *PLoS ONE* 5, 1 (2010), e8694. (pages 18, 20, 94, 120).
- [219] RUFIANGE, S., AND MCGUFFIN, M. J. DiffAni: Visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Trans. Vis. Comput. Graphics* 19, 12 (2013), 2556–2565. (page 23).
- [220] RUFIANGE, S., AND MELANÇON, G. AniMatrix: A Matrix-Based Visualization of Software Evolution. In *Proc. 2nd IEEE Work. Conf. Software Vis.* (2014), pp. 137–146. (pages 18, 19, 23, 119).

- [221] SAFFREY, P., AND PURCHASE, H. The “Mental Map” Versus “Static Aesthetic” Compromise in Dynamic Graphs: A User Study. In *Proc. 9th Conf. Australasian User Interface* (Darlinghurst, Australia, 2008), vol. 76, Australian Computer Society, Inc., pp. 85–93. (page 18).
- [222] SAGL, G., LOIDL, M., AND BEINAT, E. A Visual Analytics Approach for Extracting Spatio-Temporal Urban Mobility Information from Mobile Network Traffic. *ISPRS International Journal of Geo-Information* 1, 3 (2012), 256–271. (page 73).
- [223] SALLABERRY, A., MUELDER, C., AND MA, K.-L. Clustering, Visualizing, and Navigating for Large Dynamic Graphs. In *Graph Drawing* (2012), Springer, pp. 487–498. (page 23).
- [224] SARNI, S., MACIEL, A., AND THALMANN, D. A Spreadsheet Framework for Visual Exploration of Biomedical Datasets. In *Proc. 18th IEEE Symp. Comput.-Based Medical Syst.* (2005), CBMS, pp. 159–164. (page 30).
- [225] SCHEIDEGGER, C., KOOP, D., SANTOS, E., VO, H., CALLAHAN, S., FREIRE, J., AND SILVA, C. Tackling the Provenance Challenge one layer at a time. *Concurr. Comput. : Pract. Exper.* 20, 5 (2008), 473–483. (page 30).
- [226] SCHÖLKOPF, B., SMOLA, A. J., AND MÜLLER, K.-R. Advances in Kernel Methods. MIT Press, Cambridge, MA, USA, 1999, ch. Kernel Principal Component Analysis, pp. 327–352. (page 125).
- [227] SECURITY COUNCIL, U. N. Twenty-eighth report of the Secretary-General on the United Nations Operation in Côte d’Ivoire. *S/2011/387* (24 June 2011). (page 79).
- [228] SECURITY COUNCIL, U. N. Twenty-ninth progress report of the Secretary-General on the United Nations Operation in Côte d’Ivoire. *S/2011/807** (30 December 2011). (page 79).
- [229] SECURITY COUNCIL, U. N. Twenty-seventh progress report of the Secretary-General on the United Nations Operation in Côte d’Ivoire. *S/2011/211* (30 March 2011). (page 79).
- [230] SECURITY COUNCIL, U. N. Special report of the Secretary-General on the United Nations Operation in Côte d’Ivoire. *S/2012/186* (28 March 2012). (pages 79, 80).
- [231] SECURITY COUNCIL, U. N. Thirtieth progress report of the Secretary-General on the United Nations Operation in Côte d’Ivoire. *S/2012/506* (29 June 2012). (page 79).
- [232] SEO, J., AND SHNEIDERMAN, B. A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data. *Information Visualization* 4, 2 (2005), 96–113. (page 66).
- [233] SHAMIR, A., AND STOLPNIK, A. Interactive Visual Queries for Multivariate Graphs Exploration. *Computers & Graphics* 36, 4 (2012), 257–264. Applications of Geometry Processing. (pages 17, 47, 48).
- [234] SHANNON, P., MARKIEL, A., OZIER, O., BALIGA, N. S., WANG, J. T., RAMAGE, D., AMIN, N., SCHWIKOWSKI, B., AND IDEKER, T. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research* 13, 11 (2003), 2498–2504. (page 15).

- [235] SHAW-ETAYLOR, J., AND CRISTIANINI, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. (page 125).
- [236] SHEN, Z., AND MA, K.-L. MobiVis: A Visualization System for Exploring Mobile Data. In *Proc. IEEE Pacific Visualization Symp.* (March 2008), pp. 175–182. (page 73).
- [237] SHEN, Z., MA, K.-L., AND ELIASSI-RAD, T. Visual Analysis of Large Heterogeneous Social Networks by Semantic and Structural Abstraction. *IEEE Trans. Vis. Comput. Graphics* 12, 6 (2006), 1427–1439. (pages 17, 48).
- [238] SHEN-ORR, S. S., MILO, R., MANGAN, S., AND ALON, U. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nature Genetics* 31 (2002), 1061–4036. (page 16).
- [239] SHETTY, J., AND ADIBI, J. The Enron Email Dataset Database Schema and Brief Statistical Report, 2004. (page 63).
- [240] SHI, L., WANG, C., AND WEN, Z. Dynamic Network Visualization in 1.5D. In *Proc. IEEE Pacific Visualization Symp.* (2011), pp. 179–186. (page 23).
- [241] SHNEIDERMAN, B. Tree Visualization with Tree-maps: 2-d Space-filling Approach. *ACM Transactions on Graphics* 11, 1 (1992), 92–99. (page 56).
- [242] SHNEIDERMAN, B. Dynamic queries for visual information seeking. *IEEE Software* 11, 6 (Nov. 1994), 70–77. (page 12).
- [243] SHNEIDERMAN, B. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc. IEEE Symp. Visual Languages* (1996), pp. 336–343. (pages 11, 49).
- [244] SHNEIDERMAN, B., AND ARIS, A. Network Visualization by Semantic Substrates. *IEEE Trans. Vis. Comput. Graphics* 12, 5 (2006), 733–740. (pages 17, 47, 48).
- [245] SHRINIVASAN, Y. B., AND VAN WIJK, J. J. Supporting the analytical reasoning process in information visualization. In *Proc. 26th Annu. SIGCHI Conf. Human factors in computing syst.* (2008), CHI, ACM, pp. 1237–1246. (page 33).
- [246] SIIRTOLA, H., AND MÄKINEN, E. Constructing and Reconstructing the Reorderable Matrix. *Information Visualization* 4, 1 (2005), 32–48. (page 16).
- [247] SILVA, C., FREIRE, J., SANTOS, E., AND ANDERSON, E. Provenance-Enabled Data Exploration and Visualization with VisTrails. In *Conf. 23rd SIBGRAPI Graph., Patterns and Images Tutorials* (2010), pp. 1–9. (page 30).
- [248] SIMONS, D. J. Current Approaches to Change Blindness. *Visual Cognition* 7, 1–3 (2000), 1–15. (page 19).
- [249] SPENCE, R. *Information Visualization: Design for Interaction*, 2 ed. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2007. (page 11).
- [250] STEIGER, M., BERNARD, J., MITTELSTÄDT, S., LÜCKE-TIEKE, H., KEIM, D., MAY, T., AND KOHLHAMMER, J. Visual Analysis of Time-Series Similarities for Anomaly Detection in Sensor Networks. *Comput. Graph. Forum* 33, 3 (2014), 401–410. (page 120).

- [251] STEIN, K., WEGENER, R., AND SCHLIEDER, C. Pixel-Oriented Visualization of Change in Social Networks. In *IEEE Proc. Int. Conf. Advances in Social Networks Analysis and Mining* (2010), pp. 233–240. (page 23).
- [252] STERNBERG, R. J., MIO, J., AND MIO, J. S. *Cognitive Psychology*. Cengage Learning/Wadsworth, 2008. (page 96).
- [253] STOLPER, C. D., KAHNG, M., LIN, Z., FOERSTER, F., GOEL, A., STASKO, J. T., AND CHAU, D. H. GLO-STIX: Graph-Level Operations for Specifying Techniques and Interactive eXploration. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (Dec. 2014), 2320–2328. (page 17).
- [254] STOLTE, C., TANG, D., AND HANRAHAN, P. Multiscale Visualization using Data Cubes. In *Proc. IEEE Symp. Inform. Visualization* (2002), pp. 7–14. (page 30).
- [255] STOLTE, C., TANG, D., AND HANRAHAN, P. Polaris: a System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Trans. Vis. Comput. Graphics* 8, 1 (2002), 52–65. (page 30).
- [256] STOLTE, C., TANG, D., AND HANRAHAN, P. Multiscale Visualization using Data Cubes. *IEEE Trans. Vis. Comput. Graphics* 9, 2 (2003), 176–187. (page 30).
- [257] STREHL, A., AND GHOSH, J. Value-based customer grouping from large retail data sets. vol. 4057, pp. 33–42. (page 75).
- [258] SUMAN, B., AND KUMAR, P. A Survey of Simulated Annealing as a Tool for Single and Multiobjective Optimization. *J. of the Operational Research Society* 57, 10 (2006), 1143–1160. (page 104).
- [259] THOMAS, J. J., AND COOK, K. A. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005. (pages 4, 13, 71).
- [260] THOMAS, J. J., AND COOK, K. A. A Visual Analytics Agenda. *IEEE Comput. Graph. and App.* 26, 1 (2006), 10–13. (page 4).
- [261] TOMINSKI, C., ABELLO, J., AND SCHUMANN, H. CGV - An Interactive Graph Visualization System. *Computers & Graphics* 33, 6 (2009), 660–678. (pages 17, 47, 48).
- [262] TOMINSKI, C., ABELLO, J., VAN HAM, F., AND SCHUMANN, H. Fisheye Tree Views and Lenses for Graph Visualization. In *Proc. Int. Conf. Information Visualisation* (2006), pp. 17–24. (page 48).
- [263] TORY, M., AND MOLLER, T. Human factors in visualization research. *IEEE Trans. Vis. Comput. Graphics* 10 (Jan. 2004), 72–84. (page 11).
- [264] TRAAG, V. A., BROWET, A., CALABRESE, F., AND MORLOT, F. Social Event Detection in Massive Mobile Phone Data Using Probabilistic Location Inference. In *Proc. IEEE 3rd Int. Conf. Social Computing and Privacy, Security, Risk and Trust* (Oct. 2011), pp. 625–628. (page 73).
- [265] TUFTE, E. R. *The Visual Display of Quantitative Information*, second ed. Graphics Press, 2001. (pages 18, 30, 119).

- [266] TURKAY, C., FILZMOSER, P., AND HAUSER, H. Brushing Dimensions – A Dual Visual Analysis Model for High-Dimensional Data. *IEEE Trans. Vis. Comput. Graphics* 17, 12 (Dec. 2011), 2591–2599. (page 137).
- [267] VAN DEN ELZEN, S., BLAAS, J., HOLTEN, D., BUENEN, J.-K., VAN WIJK, J. J., SPOUSTA, R., MIAO, A., SALA, S., AND CHAN, S. Exploration and Analysis of Massive Mobile Phone Data: A Layered Visual Analytics approach. In *Proc. 3rd Int. Conf. Analysis of Mobile Phone Datasets* (Cambridge, MA, May 2013). (pages 23, 69, 185).
- [268] VAN DEN ELZEN, S., HOLTEN, D., BLAAS, J., AND VAN WIJK, J. J. Reordering Massive Sequence Views: Enabling Temporal and Structural Analysis of Dynamic Networks. In *Proc. IEEE PacificVis* (Feb. 2013), pp. 33–40. (pages 23, 91, 93, 113, 122, 185).
- [269] VAN DEN ELZEN, S., HOLTEN, D., BLAAS, J., AND VAN WIJK, J. J. Dynamic Network Visualization with Extended Massive Sequence Views. *IEEE Trans. Vis. Comput. Graphics* 20, 8 (Aug. 2014), 1087–1099. (pages 23, 91, 120, 122).
- [270] VAN DEN ELZEN, S., HOLTEN, D., BLAAS, J., AND VAN WIJK, J. J. Reducing Snapshots to Points: A Visual Analytics Approach to Dynamic Network Exploration. *IEEE Trans. Vis. Comput. Graphics* xx, xx (Dec. 2015), xxxx–xxxx. to appear. (pages 23, 117, 185).
- [271] VAN DEN ELZEN, S., VAN DORTMONT, M., BLAAS, J., HOLTEN, D., VAN HAGE, W., BUENEN, J.-K., VAN WIJK, J. J., SPOUSTA, R., SALA, S., CHAN, S., AND KUZMICKAS, A. Data for Development Reloaded: Visual Matrix Techniques for the Exploration and Analysis of Massive Mobile Phone Data. In *Proc. 4th Int. Conf. Analysis of Mobile Phone Datasets* (Cambridge, MA, April 2015). (page 185).
- [272] VAN DEN ELZEN, S., AND VAN WIJK, J. J. BaobabView: Interactive construction and analysis of decision trees. In *IEEE Conf. Visual Analytics Science and Technology* (Oct. 2011), pp. 151–160. (page 147).
- [273] VAN DEN ELZEN, S., AND VAN WIJK, J. J. Small Multiples, Large Singles: A New Approach for Visual Data Exploration. *Comput. Graph. Forum* 32, 3pt2 (2013), 191–200. (pages 23, 27, 56, 104, 120).
- [274] VAN DEN ELZEN, S., AND VAN WIJK, J. J. Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (Dec. 2014), 2310–2319. (pages 23, 45, 185).
- [275] VAN DER MAATEN, L. Barnes-Hut-SNE. In *Proc. Int. Conf. Learning Representations* (2013). (pages 125, 126).
- [276] VAN DER MAATEN, L., AND HINTON, G. E. Visualizing High-Dimensional Data Using t-SNE. *J. Machine Learning Research* 9 (2008), 2579–2605. (page 125).
- [277] VAN DER MAATEN, L. J. P., POSTMA, E. O., AND VAN DEN HERIK, H. J. Dimensionality Reduction: A Comparative Review. *Technical Report TiCC TR 2009-005* (2009). (page 125).
- [278] VAN HAM, F., AND PERER, A. Search, Show Context, Expand on Demand: Supporting Large Graph Exploration with Degree-of-Interest. *IEEE Trans. Vis. Comput. Graphics* 15, 6 (2009), 953–960. (pages 11, 49).

- [279] VAN WIJK, J. J. Information Visualization: Challenges and Opportunities. Capstone presented at the IEEE 24th VIS Conf., Atlanta, Georgia, USA. <https://vimeo.com/80334651>. (page 4).
- [280] VAN WIJK, J. J., AND NUIJ, W. A. A. A Model for Smooth Viewing and Navigation of Large 2D Information Spaces. *IEEE Trans. Vis. Comput. Graphics* 10, 4 (2004), 447–458. (page 52).
- [281] VAN WIJK, J. J., AND VAN SELOW, E. R. Cluster and calendar based visualization of time series data. In *IEEE Proc. Symp. Information Visualization* (1999), pp. 4–9, 140. (page 120).
- [282] VEHLow, C., BECK, F., AUWÄRTER, P., AND WEISKOPF, D. Visualizing the Evolution of Communities in Dynamic Graphs. *Computer Graphics Forum* 34, 1 (2015), 277–288. (page 23).
- [283] VEHLow, C., BURCH, M., SCHMAUDER, H., AND WEISKOPF, D. Radial Layered Matrix Visualization of Dynamic Graphs. In *IEEE 17th Int. Conf. Information Visualisation* (July 2013), pp. 51–58. (pages 20, 23).
- [284] VOGEL, H. A Better Way to Construct the Sunflower Head. *Mathematical Biosciences* 44, 3–4 (1979), 179–189. (page 128).
- [285] VON LANDESBERGER, T., BREMM, S., ANDRIENKO, N., ANDRIENKO, G., AND TEKUSOVA, M. Visual Analytics Methods for Categorical Spatio-Temporal Data. In *Proc. IEEE Symp. Visual Analytics Science and Technology* (Oct. 2012), pp. 183–192. (page 73).
- [286] VON LANDESBERGER, T., DIEHL, S., BREMM, S., AND FELLNER, D. W. Visual analysis of contagion in networks. *Information Visualization* 14, 2 (2015), 93–110. (page 120).
- [287] VON LANDESBERGER, T., KUIJPER, A., SCHRECK, T., KOHLHAMMER, J., VAN WIJK, J. J., FEKETE, J.-D., AND FELLNER, D. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Comput. Graph. Forum* 30, 6 (2011), 1719–1749. (pages 49, 120).
- [288] WANG, Q., AND BOYER, K. L. Feature Learning by Multidimensional Scaling and Its Applications in Object Recognition. In *Proc. 26th Conf. Graph., Patterns and Images* (2013), pp. 8–15. (page 126).
- [289] WANG BALDONADO, M. Q., WOODRUFF, A., AND KUCHINSKY, A. Guidelines for Using Multiple Views in Information Visualization. In *Proc. Working Conf. Advanced Vis. Interfaces* (New York, NY, USA, 2000), ACM, pp. 110–119. (page 12).
- [290] WARD, M. O. XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proc. IEEE Conf. Visualization* (1994), pp. 326–333. (page 51).
- [291] WARD, M. O. A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization. *Information Visualization* 1, 3–4 (2002), 194–210. (pages 17, 48).
- [292] WARE, C. *Information Visualization: Perception for Design*, 3 ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2012. (page 10).

- [293] WATTENBERG, M. Visual Exploration of Multivariate Graphs. In *Proc. SIGCHI Conf. Human Factors in Computing Systems* (2006), ACM, pp. 811–819. (pages 17, 49).
- [294] WATTS, D. J. *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company, Feb. 2004. (page 15).
- [295] WATTS, D. J., AND STROGATZ, S. H. Collective Dynamics of ‘Small-World’ Networks. *Nature* 393, 6684 (1998), 440–442. (page 129).
- [296] WEHREND, S., AND LEWIS, C. A Problem-Oriented Classification of Visualization Techniques. In *Proc. 1st Conf. Visualization* (1990), VIS, IEEE Computer Society Press, pp. 139–143. (page 38).
- [297] WILLEMS, N., VAN HAGE, W. R., DE VRIES, G., JANSSENS, J. H. M., AND MALAISE, V. An Integrated Approach for Visual Analysis of a Multisource Moving Objects Knowledge Base. *Int. J. Geogr. Inf. Sci.* 24, 10 (2010), 1543–1558. (page 30).
- [298] WILLETT, W., HEER, J., AND AGRAWALA, M. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Trans. Vis. Comput. Graphics* 13, 6 (2007), 1129–1136. (pages 12, 52, 53).
- [299] WONG, N., CARPENDALE, M. S. T., AND GREENBERG, S. EdgeLens: An Interactive Method for Managing Edge Congestion in Graphs. In *Proc. 9th Annu. IEEE Conf. Information Visualization* (2003), pp. 51–58. (pages 47, 48).
- [300] WOODS, D. The tragedy of the cocoa pod: rent-seeking, land and ethnic conflict in Ivory Coast. *The Journal of Modern African Studies* 41 (Nov. 2003), 641–655. (page 83).
- [301] WU, H.-M., TZENG, S., AND CHEN, C.-H. Matrix Visualization. In *Handbook of Data Visualization*, 1 ed., Springer Handbooks of Comp. Statistics. Springer Berlin Heidelberg, 2008, pp. 681–708. (page 93).
- [302] WU, Y., AND TAKATSUKA, M. Visualizing Multivariate Network on the Surface of a Sphere. In *Proc. Asia-Pacific Symp. Information Visualisation* (2006), Australian Computer Society, Inc., pp. 77–83. (pages 17, 48).
- [303] YE, Q., WU, B., HU, D., AND WANG, B. Exploring Temporal Egocentric Networks in Mobile Call Graphs. In *Proc. 6th Int. Conf. Fuzzy Sys. Knowledge Discovery* (Aug. 2009), vol. 2, pp. 413–417. (page 73).
- [304] YE, Q., ZHU, T., HU, D., WU, B., DU, N., AND WANG, B. Cell Phone Mini Challenge Award: Social Network Accuracy; Exploring Temporal Communication in Mobile Call Graphs. In *Proc. IEEE Symp. Visual Analytics Science and Technology* (Oct. 2008), pp. 207–208. (page 73).
- [305] YI, J. S., ELMQVIST, N., AND LEE, S. TimeMatrix: Analyzing temporal social networks using interactive matrix-based visualizations. *Int. J. Human-Computer Interaction* 26, 11-12 (2010), 1031–1051. (pages 20, 23).
- [306] YI, J. S., KANG, Y. A., STASKO, J. T., AND JACKO, J. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Trans. Vis. Comput. Graphics* 13, 6 (2007), 1224–1231. (page 38).

List of Figures

1.1	Network model.	2
1.2	Real-world networks.	3
1.3	Chapter outline overview.	5
2.1	Visualization pipeline.	11
2.2	Visual analytics process.	13
2.3	Node-link diagram and visual adjacency matrix	14
2.4	Hairball Network Visualization	15
2.5	Hybrid node-link matrix visualization	16
2.6	Multivariate network exploration examples.	17
2.7	Example node-link animation.	18
2.8	Small multiples, 2.5D and matrix cube.	19
2.9	Dynamic multivariate network taxonomy.	23
3.1	Novel visual data exploration method using small multiples.	28
3.2	Visual data exploration methods.	32
3.3	Generation of small multiples by splitting a large single.	35
3.4	Graphical user interface.	36
3.5	Four different tested interaction methods.	39
3.6	Interaction method user preference.	42
4.1	Multivariate network exploration using selections of interest.	46
4.2	The DOSA exploration process with selections of interest as central element.	50
4.3	Graphical user interface of the implemented prototype.	54
4.4	Different types of edges involved in a node selection.	55
4.5	A selection of interest containing many edges that clutter the view.	56
4.6	U.S.A. migration data exploration.	58
4.7	Testing for predominantly inbound or outbound regions.	59
4.8	Migration of highly populated low and high crime regions.	60
4.9	Testing for correlation between age and migration.	60
4.10	Path exploration, finding indirect routes from New York to Washington.	62
4.11	Enron e-mail communication exploration using two selections of interest.	65
4.12	Typical C-level communication.	65

5.1	Different coherent components of the visual analytics solution.	70
5.2	Graphical user interface.	74
5.3	Measure overview simultaneously rendering multiple measures.	76
5.4	Measure contribution and matrix views.	76
5.5	Peak in the number of calls and call change due to new year.	80
5.6	Local communication and higher level communication patterns.	80
5.7	General events and local increase identification.	81
5.8	Clashes between communities in Arrah.	82
5.9	Attack on the village of Zriglo.	82
5.10	Increased phone calls correlated with rainfall anomalies.	82
5.11	Rainfall activities in Bas-Sassandra region.	84
5.12	Rainfall activities in Dix-Huit Montagnes region.	84
5.13	Local event decreased call correlation patterns.	85
5.14	Cluster of towers identified in western region with unusual low activity.	85
5.15	Local event decreased call correlation patterns.	86
5.16	Bad weather conditions influencing local call activity around Agboville.	86
5.17	Day before supposedly electric failure in the western region.	86
5.18	Highlight of rainfall pattern in Denguele region in February 2012.	87
6.1	Different visualizations of dynamic network data.	93
6.2	Gestalt principles applied to msv.	96
6.3	Temporal and structural properties of a dynamic network.	97
6.4	Different color encodings for the MSV edges.	98
6.5	<i>Massive Sequence Views</i> as part of a multiple coordinated view application.	98
6.6	Overlapping blocks diagram.	103
6.7	Pareto frontier with search space for both optimization criteria.	105
6.8	Edge length standard deviation μ minimized.	105
6.9	Dynamic network visualized using standard msv and circular msv.	107
6.10	Time-series node data visualized using heatmap approach.	109
6.11	Standard msv (a) and combined reordering strategy applied to msv (b).	109
6.12	Minimizing edge length node reordering strategy applied to circular msv.	110
6.13	Standard hierarchy-constrained msv and combined reordering strategy.	110
6.14	Time-series data associated with each node.	112
6.15	Average edge length versus number of iterations in simulated annealing.	113
7.1	Reducing snapshots of the dynamic network to points.	118
7.2	Visual analytics approach for the exploration of dynamic networks.	121
7.3	Effect of different overlap values α for the snapshots.	123
7.4	Linear dimensionality reduction technique applied to snapshots.	124
7.5	Non-linear dimensionality reduction.	126
7.6	Graphical user interface of the prototype.	127
7.7	Simplifying the identification of states in the network.	128
7.8	Linear reduction reveals four stable network states.	130
7.9	Alternative projections of the dynamic network snapshots.	131
7.10	Linear reduction PCA reveals four network states.	131
7.11	Alternative projections providing more insight.	132

7.12	Linear reduction PCA without and with z-normalization.	133
7.13	Alternative projections with snapshots colored by hour of day.	134
7.14	Non-linear dimensionality reduction of snapshots colored globally. . . .	135
8.1	Chapter relevance overview.	144
8.2	Mock-up illustration of integrated techniques.	145

List of Tables

3.1	Results for the ANOVA analysis on efficiency.	41
3.2	Usability questionnaire results.	41
5.1	Requirements to support users in the exploration and analysis.	72

Summary

Interactive Visualization of Dynamic Multivariate Networks

NETWORKS are ubiquitous in society, some examples are communication networks, social networks, financial networks, and transportation networks. The understanding of networks can help to take better decisions. This dissertation explores and presents interaction and visualization techniques for the exploration and analysis of networks. State-of-the-art as well as current industry standards are mainly based on purely automated methods, such as rule based retrieval. However, these automated methods fall short due to aggregation and loss of context. Also, purely visual methods fall short, for instance due to limited screen resolution. This doctoral dissertation therefore examines the following central research question: *“How to enable people to obtain insight in dynamic multivariate networks using a combination of automated and interactive visual methods?”* We show that a combination of interaction techniques, automated methods, and visualization supports users in the analytical reasoning process to detect anomalies, test hypotheses, gather insights, and obtain new knowledge. In practice, networks are not only large, but also dynamic and multivariate: they have a structural as well as a temporal aspect and besides the topological structure of the network, multivariate data on the nodes and links is available. We present novel techniques that address one or a combination of these aspects.

In Chapter 3 we present a novel visual exploration method based on small multiples and large singles for effective and efficient data analysis that is not restricted to multivariate networks. Users are enabled to explore the state space by offering multiple alternatives from the current state and can then select the alternative of choice and continue the analysis. The intermediate steps in the exploration process are preserved and can be revisited and adapted using an intuitive navigation mechanism based on the well-known undo-redo stack and a filmstrip metaphor.

Exploration and analysis methods are often focused on a single aspect; the network topology or the multivariate data. In addition, tools and techniques are highly domain specific and require expert knowledge. We focus on the non-expert user and propose a novel solution for multivariate network exploration and analysis that tightly couples structural and multivariate analysis in Chapter 4. In short, we go from Detail to Overview via Selections and Aggregations (DOSA): users are enabled to gain insights

through the creation of selections of interest (manually or automatically), and producing high-level, infographic-style overviews simultaneously.

Visual analytics techniques for the exploration and analysis of real-world massive mobile phone data are investigated and implemented in Chapter 5. First, we identify user tasks and develop a system following a visual analytics approach by tightly integrating visualization, interaction and algorithmic support. In addition, we provide different views to explore both space, time, and structure in one unified framework. The system is then evaluated by exploring a massive mobile phone dataset containing billions of calls and SMS exchanges between millions of users located in Ivory Coast in context of the Data for Development challenge.

In Chapter 6 we introduce a technique that extends the Massive Sequence View (MSV) for the analysis of temporal and structural aspects of dynamic networks. Using features in the data as well as Gestalt principles in the visualization, such as closure, proximity, and similarity, we developed node reordering strategies for the MSV to make these features stand out that optionally take the hierarchical node structure into account. This enables users to find temporal properties, such as trends, counter trends, periodicity, temporal shifts, and anomalies in the network as well as structural properties, such as communities and stars.

In Chapter 7 we present a visual analytics model for dynamic network exploration based on discretization, vectorization, dimensionality reduction and visualization in which each time-window of the dynamic network is considered as a point in high-dimensional space. This enables abstraction, visualization and exploration. Two juxtaposed views enable the discovery of stable states, outlier states and the evolution of the network in general.

Problems or domains that are not modeled or presented as networks can often naturally be approached as a network of objects with relations between them. This makes the presented techniques in this dissertation applicable to a broad range of domains.

Curriculum Vitæ

STEFANO Johannes (Stef) van den Elzen was born on 17 December 1985 in Nijmegen and raised in Schaijk, the Netherlands. He completed his pre-university secondary education at the Mondriaan College in Oss. From 2004, he studied Computer Science and Engineering at the department of Mathematics and Computer Science at Eindhoven University of Technology. In 2011 he obtained his Master of Science degree with honors. His graduation work on the interactive construction, analysis and visualization of decision trees was performed in the visualization group under the guidance of prof.dr.ir. Jarke J. van Wijk.

In July 2011, he started as a developer with SynerScope B.V. on a PhD project at the Eindhoven University of Technology. This research was conducted under the supervision of prof.dr.ir. Jarke J. van Wijk (TU/e), dr.ir. Danny H.R. Holten (SynerScope B.V.), and dr.ir. Jorik Blaas (SynerScope B.V.). For his research, he developed tools and techniques for the exploration of dynamic multivariate networks. The results of his PhD research, completed in 2015, were published in a number of articles in international conference proceedings and journals.

He received three best paper awards: at IEEE PacificVis 2013 for the work on Extended Massive Sequence Views [268], at IEEE InfoVis 2014 for the work on Multivariate Network Exploration and Presentation [274], and at IEEE VAST 2015 for the work on Dynamic Network Exploration [270]. Furthermore, he received two best visualization awards for his work on the Exploration and Analysis of Massive Mobile Data at the Data for Development Challenges 2013 [267] and 2015 [271]. The work on Reordering Massive Sequence Views [268] led to a patent application.

Since July 2015 he continues at SynerScope B.V. as a visualization research scientist.

Index

Symbols

ϵ -constraint 104

A

activity log 122
additive blending 56, 77
aggregations 46
analysis 71
animation 30, 50, 119
anomalies 92, 97

B

betweenness 51
bin ranges 37
block 96
boxes 51
branching 43
brushing 51, 113

C

call 70
 behavior 71, 88
 change 71
carry-over effects 38
CDR 70
cell tower 70
circular msv 92, 106
closeness 51
closure 92, 96
clustering 34
collections *see* small multiples
communities 92, 129
community 97
 comparison 29, 56
complex correlation 108
complex patterns 88
computational
 scalability 111
contribution view 77
convolution 103
coordinated view 114
counter trend 96
counterbalance 38

D

d4d challenge 79
data rhythm 103
DDQC 126
degree distribution 129
detail 47
dimensionality reduction 118,
 125
direct manipulation 51
discretization 121
divide-and-conquer 74
domain independence ... 66
DOSA 47, 66
drag handles 52
dual view 37, 42
dynamic
 attribute 55
 network 3, 92, 118

E

edge
 attribute 99
 background 55
 between 55
 directionality 51
 within 55
edge length
 minimization 100
efficiency 37, 40
embedding 48
evaluation 37
event detection 71
events 70
evolution 118
exploration 2, 71
 path 32
 tasks 38

F

fatigue *see* carry-over effects
filmstrip 29
filter 31
 split 33
focus+context 48

Freedman-Diaconis 37

G

geospatial view 77
gestalt 92, 95
GUI 36, 74
guidance 29, 56

H

hierarchical clustering ... 84
hierarchy-constrained msv
 111
high-dimensional space . 121,
 125
high-level
 comparison 51
 overview 47, 76
histogram 56
history trail 29

I

info-graphic 46
 overview 56
interaction 4
 methods 38
intuitive interaction 64

J

juxtaposition 47, 64, 118, 121

K

kernel size 103

L

large single 28, 31
layers 52
layout stability 94
learning *see* carry-over effects
level-of-detail zoom 57
lexicographic order 99
linked views 72
load-on-demand 74, 75
local events 88

- M
 - mapping split 34
 - matrix view 78
 - MDS 126
 - mental map 94
 - metaphors 72
 - migration 57
 - missing data 87
 - mobile data 70
 - MSV 92
 - multivariate
 - data 28, 46
 - exploration 64
 - network 2, 46, 66
- N
 - navigation trail 32
 - network 2
 - exploration 47
 - structure 46
 - networks 92
 - node
 - hierarchy 92, 107
 - ordering 94
 - node-link 47
 - normal behavior 101
 - normalization 125
- O
 - outlier state 118
 - overlap 101, 123
 - overview 46, 47
- P
 - padding 103
 - pagination 37
 - painting 51
 - parallel-coordinate plot .. 56
 - parameter
 - effect 32
 - reset 37
 - pareto frontier 105
 - pareto optimal 104
 - path exploration 62
- PCA 125
- periodicity 92, 96
- pivotgraph 48
- pre-computation 74
- presentation 46, 71
- projection 123, 127
- proximity 92, 96
- Q
 - querying 51
- R
 - rainfall anomalies 83
 - readability 48
 - real-time exploration 74
 - recurring state 118
 - relation 2
 - reordering 92
 - strategies 92
- S
 - scalability 42, 49, 64, 111, 136
 - scatter-plot 56
 - scented widget 52, 53
 - select 31
 - selections 46
 - selections of interest 47
 - self-loops 81
 - semantic substrates 48
 - shift 92, 96
 - similarity 92, 96
 - simulated annealing 100, 104
 - sliding window 123
 - small multiples ... 28, 31, 56, 119, 120
 - generation 35
 - small-world 129
 - snapshots 118, 119
 - sociology 92
 - splitting 33
 - stable state 118
 - standard deviation 104
 - star pattern 97
 - state 31, 118
 - states 118
 - structural 92
 - structure 47
- T
 - t-SNE 126
 - temporal 92
 - behavior 100
 - time-series data 92, 108
 - time-windows 123
 - timeline 129
 - topology 111
 - tower-to-tower 70
 - treemap 56
 - trellis display *see* small multiples
 - trend 92, 96
- U
 - usability 40
 - user
 - experience 71
 - satisfaction 33, 37
 - study 37
 - tasks 71
- V
 - vectorization 121, 125
 - visual
 - analytics 4, 34, 70, 118
 - clutter 92
 - exploration .. 28, 36, 92
 - history 33
 - linking 34
 - navigation 31
 - parameter space ... 30
 - parameters 31
 - patterns 93
 - querying 51, 64
 - scalability 111
 - variables ... 47, 48, 56
- Z
 - zoom-pan 32, 50, 77, 127