Towards Visualizing and Exploring Multivariate Networks on Mobile Devices

Tom Horak
Interactive Media Lab Dresden
Technische Universität Dresden
Dresden, Germany
horakt@acm.org

Ricardo Langner
Interactive Media Lab Dresden
Technische Universität Dresden
Dresden, Germany
langner@acm.org

Raimund Dachselt
Interactive Media Lab Dresden
Centre for Tactile Internet (CeTI)
Cluster of Excellence Physics of Life
Technische Universität Dresden
Dresden, Germany
dachselt@acm.org

Figure 1: For exploring multivariate networks on mobiles, we envision an interface consisting of multiple views. An analyst can switch between these on one device or offload them to multiple devices.

ABSTRACT

Mobile devices are getting more and more powerful, however, interfaces for exploring multivariate network data on-the-go remain rare and challenging to realize. In this work, we present considerations as well as concepts on how to enable such exploration on mobile devices. Specifically, we discuss that an interface with force-directed node-link representation alongside additional views for investigating multivariate aspects and filter functionalities seems promising. Further, such an interface could be advanced by incorporating multiple mobile devices in parallel. As part of our ongoing investigations, we have realized an early prototype indicating the general feasibility of our concepts.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing**; **Visualization**.

KEYWORDS

Mobile visualization, multivariate graph visualization, node-link graphs, cross-device interaction.

ISS '20 Companion, November 8–11, 2020, Virtual Event, Portugal

ACM ISBN 978-1-4503-7526-9/20/11...$15.00
https://doi.org/10.1145/3380867.3426201

ACM Reference Format:
Tom Horak, Ricardo Langner, and Raimund Dachselt. 2020. Towards Visualizing and Exploring Multivariate Networks on Mobile Devices. In *Companion Proceedings of the 2020 Conference on Interactive Surfaces and Spaces (ISS '20 Companion), November 8–11, 2020, Virtual Event, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3380867.3426201

1 INTRODUCTION & BACKGROUND

The powerful capabilities of today's mobile devices allow to conduct more and more professional work on them, including visual data analysis. Research has already provided concepts for such mobile data explorations [3] by, e.g., enabling touch input, discussing aspects of responsiveness [1, 6], or leveraging cross-device concepts [2, 8]. However, these are mostly focused on multivariate data exploration, while multivariate *networks* (or graphs) remain underexplored. These networks are relevant in professional domains (e.g., person networks, energy grids, ontologies) and are particular complex and challenging to visualize: analyst have to consider structural aspects in form of relations between objects (i.e., edges between nodes), alongside multivariate aspects of both objects and relations (i.e., attributes of nodes and/or edges) [12]. Further, it can be of interest to compare network snapshots of different points of time. Recently, Eichmann et al. [4] started considering mobile multivariate networks, adopting a faceted search-like exploration style, but the concept is less suited for general graph tasks [12].

In this work, we discuss strategies for bringing established concepts for visually exploring multivariate networks to mobile devices (Figure 1). More specifically, this involves concepts for an interface allowing to explore both structural and multivariate aspects on one mobile device, before discussing advanced exploration possibilities when using multiple co-located mobile devices in combination.

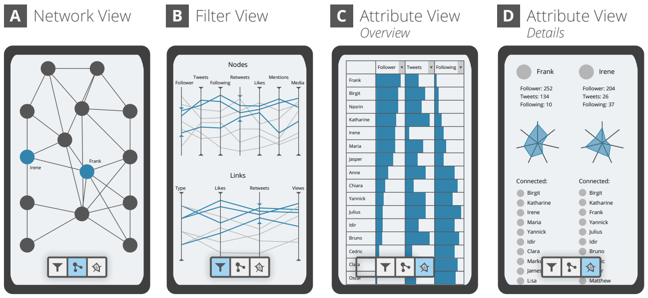

Figure 2: Proposed view concepts for a mobile device: The (a) network view hosts a force-directed node-link graph visualizing the structural aspects; the (b) filter view allows to filter entities via interactive brushing in parallel coordinate plots; and the (c+d) attribute view visualizes multivariate aspects, either by providing an overview on all entities (c) or by showing the details for selected ones (d).

2 ENABLING NETWORK EXPLORATION ON A MOBILE DEVICE

Most existing visualization approaches for multivariate networks either apply layouts with side-by-side representations of structural and multivariate aspects or embed multivariate aspects within the network visualization itself [12]. However, these approaches are quickly becoming too dense to work on small screens. Consequently, we consider it to be more suitable to use *basic representations* split into *separated views*. Specifically, this means that we propose to have separate views for representing the structural aspects, the multivariate aspects as well as filter functionalities (Figure 2). In a mobile interface with such views, an analyst can then quickly switch between them and focus on the aspects that are currently of most interest. In the following, we will discuss considerations for these different views as well as how to apply touch-based interactions.

2.1 Representing Structural Aspects

In order to provide a suitable representation of the structural aspects on a mobile device, it is important to consider the general layout approach, the specific algorithm, as well as the applied encoding.

2.1.1 Layout Approach. Nobre et al. [12] distinguish between two main categories of layouts, node-link layouts and tabular layouts. In general, node-link diagrams are the more common technique for representing networks, while tabular layouts (e.g., adjacency matrix) may imply a higher cognitive load from the user [13]. Both approaches face scalability issues and can quickly become cluttered, which is particularly relevant in the context of small screen spaces. However, for node-link layouts, these can partly be tackled with improved encoding and layout algorithm (e.g., edge bundling, clustering, aggregations). Further, maintaining a reasonable size of elements (i.e., node size) for touch input is easier. In contrast, in tabular layouts, the cell size is defined by the overall size of the matrix and is often a few pixels, requiring focus+context approaches for facilitating interaction or showing further details [7]. In conclusion, we believe that for most explorations, node-link representations are more suited for mobile usage (Figure 2a).

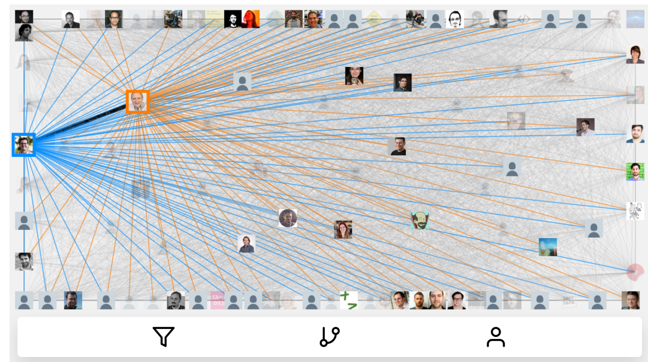

Figure 3: By using a bounded force-directed layout, the resulting graph fits into the devices viewport, independently of it's aspect ratio or orientation. Nodes are mostly placed without overlap, making it easier to select them via touch. As links can be extremely dense and hard to follow, selections should also lead to highlighting the connected links.

2.1.2 Layout Algorithm. Within a node-link representation, the positioning of the nodes is crucial for being able to recognize structural aspects. Two types of algorithms are common: force-directed ones and attribute-based ones. For the latter one, node positions are calculated by mapping attributes on an axis. While the node position then already indicates one or two node attributes, it can also lead to overlap and the actual structural aspects are hard to recognize [12]. In contrast, force-direct layout algorithms simulate forces where nodes attract and repel each other based on the existing links. As a result, strongly connected nodes are placed closer to each other while overlaps of nodes can be minimized.

These characteristics are particularly beneficial for mobile devices. On the one hand, less overlap means that nodes remain easy to select. On the other hand, the algorithm is agnostic to display size, orientation, and aspect ratio, thus, no further adaptions are required for a responsive behavior [1]. Further, force-directed algorithms can be used in a bounded fashion, guaranteeing that the elements remain within a certain area (Figure 3). This way, the network is placed in the viewport and panning operations can be avoided. Notably, one disadvantage of force-directed algorithms is that they are computational expensive, especially when user-driven changes can restart the algorithm (e.g., when manually re-positioning one node). One way to minimize this effect is applying the algorithm statically, that is, calculating the layout only once when loading the application and not during run time. This approach is used in our prototype (Figure 3; for implementation details see section 4).

2.1.3 Encoding Strategies. Attributes can be encoded on both nodes and links, for example, by adapting size, color, or shape/style. However, we believe that these should be used only in a reduced way as the network visualization is already likely to be very dense and prone to clutter. Also, particularly mapping the size to an attribute can lead to overlap or to elements being too small to be touched. As the most promising encoding, we consider color usage to indicate different types of links or nodes, opacity levels for links to reduce visual clutter, as well as icons for visually representing a node id (if applicable). For example, when comparing two snapshots of the

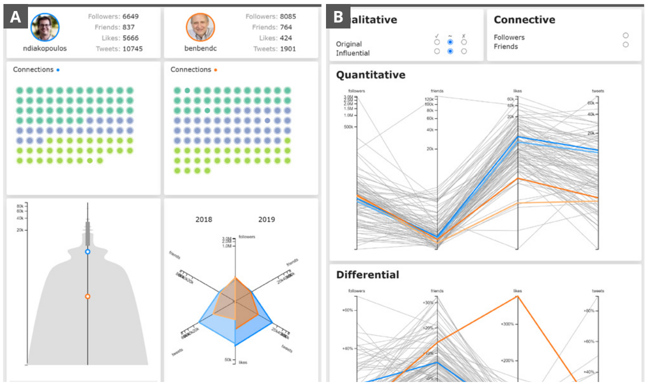

Figure 4: (a) The attribute view can show details for two selected nodes: user name and key attributes, a dot matrix indicating connected nodes (colors represent link types), a violin chart showing the two nodes in context of the overall attribute distribution, and a split star plot comparing the two nodes for two different years (left half 2018, right half 2019). (b) The filter view offers multiple parallel coordinated plots, where on each axis a range can be selected and applied as filter; further, selected nodes are indicated via color.

network (e.g., two different years), color could be used to indicate if a node or link was only present in one year or the other or in both. All in all, this rather basic encoding emphasizes that the main goal of the node-link representation is to communicate the structure, while the attribute-specific aspects are left for a separate view.

2.2 Representing Multivariate Aspects

For the multivariate aspects, it must be possible to get an overview onto them as well as investigating the details of one specific entity (which is either a node or an edge) or comparing multiple ones. We consider these to be two different modes, dependent on if a user has selected entities. If yes, details are shown, otherwise the overview.

2.2.1 Overview. The goal for the overview is to display how attributes are distributed across all nodes or edges, including possible outliers. For this, multiple options exist, e.g., histograms, TableLens-like representations, or parallel coordinates plots. Histograms are limited to showing one attribute, while for parallel coordinates plots it can be hard to mentally map the specific entities to the displayed polylines. Therefore, a table with direct encodings seems very suitable (Figure 2c). In order to consider attributes of both nodes and edges, it must be supported to switch between these.

2.2.2 Node and Edge Details. For a selected node or edge, all its respective attributes should become visible. Among others, star plots allow to visualize multiple heterogeneous attributes at the same time, since for each a separate axis is provided. While star plots face limitations when showing many attributes in parallel, we still consider them to be beneficial for mobile usage, as their fixed aspect ratio makes them easy to display as well as to use them for comparison (see below). In addition, labels, categories, or core attributes can be placed separately as text elements. Also, a violin plot could be used to relate one attribute value to the overall distribution. Although not related to the multivariate aspects, it can also be helpful to indicate all connected nodes and allow for navigating to these (Figure 2d + 4a).

2.2.3 Comparison. In order to foster comparison, the detail view can be facilitated as well. In general, comparison can be made between multiple selected entities or for one entity to an earlier snapshot of the network. For the comparison of multiple entities, overlaying or placing them side-by-side allows to quickly scan their differences and similarities [5, 9]. The star plot in particular is well suited for this, as it can allow for placing them in a grid as small multiples, overlaying the entities in one visualization, or splitting them in one half (Figure 4a, bottom right). The comparison across timesteps can be done similarly: the node or link attributes of one year can be interpreted as one entity, which is then juxta-posed or super-imposed with another year.

2.3 Interface Components & Interaction

To complete the interface, we consider filter facilities as crucial for exploring larger networks. Also, touch input must be supported in all views, as it is the common modality on mobiles.

2.3.1 Filter Functionalities. No matter how good the chosen layout and encoding strategies are, networks can become quickly overwhelming with too many nodes and edges. To drill down and focus on a subset, it should be possible to filter nodes and edges based on their attributes. Instead of offering range sliders for each attribute, we propose to enable filtering in a visual way in a parallel coordinates plot. Here, multiple attributes and their distribution can be visualized at the same time (since recognizing specific entities is not in focus), while brushing on the axes allows to define a filter range. Such a plot can be offered for both node and edge attributes as well as calculated ones (e.g., difference between snapshots). We propose to provide such filter functionalities as a separate view (Figure 2b + 4b).

2.3.2 Interaction. Based on the considerations before, the interface would consist of three main views: network view, attribute view, and filter view. In order to allow for quickly switching between these views, a small floating menu can be provided (Figure 2). Alternatively, views can be switched via swipe. For the touch interaction within the views, it is important to guarantee a certain size of the visual elements, such as nodes, rows in the table, or handles on axes, so that these can easily be selected (cf. fat-finger problem). For the network view, it should also be supported to manually re-position nodes via drag-and-drop to resolve layout issues. For larger networks, zoom and pan becomes relevant as well.

3 ADVANCED EXPLORATION WITH MULTIPLE MOBILES

While the considerations discussed so far can already help to foster the network exploration on mobile devices, the constrained screen estate puts limits on the scalability. Also, switching between different views can increase the mental demand for an analyst, e.g., when configuring filter criteria while also observing their effects. In this context, facilitating multiple mobile devices in parallel (e.g., tablet plus smartphone), can provide promising concepts to overcome the remaining limitations.

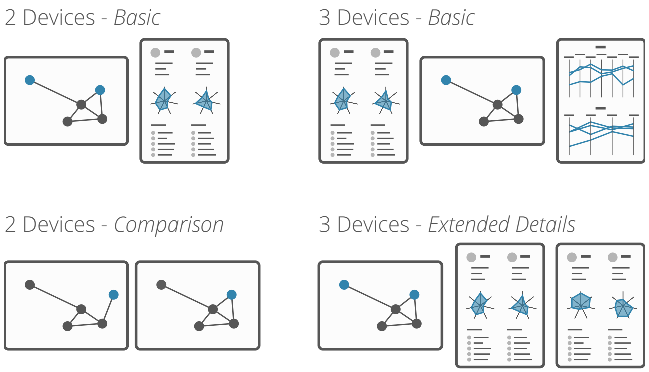

Figure 5: Example device combinations: By offloading the views to multiple devices, they can be viewed in parallel. The same view can also be shown multiple times (bottom row); then it can allow for comparing different versions or for using the additional space for extended details.

Distributed Interfaces. Particularly interesting is distributing the interface across multiple devices [8, 11]. Each device can then show one view, and their combination results in a multi-view setup (Figure 1). Here, each device can show a different view, for example, network view and attribute view (Figure 5). Selections made in the network view would then instantly reveal the attribute details on the other device. Devices can also show the same view type, e.g., the network at two different years, supporting visual comparison [5].

Spatial-awareness of Mobile Devices. As a promising extension of co-located distributed interfaces, the spatial arrangement of devices could be used to activate certain functionalities, e.g., combining displays into one screen space, adapting encodings to foster comparison, or indicating overview+detail constellations [11]. Such concepts have been presented in the general context of visualizations, but have not been investigated in detail for networks yet. Further, ad-hoc combinations with large displays that might be available in the current environment can be an interesting addition, too (see, e.g., [10]).

4 EARLY PROTOTYPE

We implemented a web-based prototype using D3 and SVG for drawing the views, Hammer.js for handling touch gestures, and socket.io for cross-device communication. Currently, we have realized the network view, attribute detail view (showing selected nodes), and filter view (Figure 1 + 4). We incorporated the force-direct layout algorithm offered by D3, used in a static and bounded fashion. The calculation runs in a WebWorker process on load. The devices can synchronize their states (e.g., loaded view, current selection) via Websockets and adapt to updates accordingly.

5 DISCUSSION & CONCLUSION

Enabling network exploration on mobile devices is a challenging task, which is worth to be explored in more detail. With our proposed strategies, concept ideas, and an early prototype, we aim to provide a starting point for such research. More specifically, we plan to develop our prototype further and fully implement the described aspects. This can then allow for properly evaluating our approach. Beyond that, it is important to explore and compare alternatives, e.g., incorporating adjacency matrices, other layout algorithms, or different attribute encodings, as well as concepts for supporting other high-level network tasks such as editing. For now, we hope that our considerations outlined here can spark further research and discussion on this topic.

ACKNOWLEDGMENTS

We thank Vincent Thiele, Sam Toorchi Roodsa, and Jonas Leupold for implementing the current version of the prototype. This work was supported by the DFG grant DA 1319/3-3 (GEMS 2.0) as well as DFG grant 389792660 as part of TRR 248 (see https://perspicuous-computing.science).

REFERENCES

[1] Keith Andrews and Aleš Smrdel. 2017. Responsive Data Visualisation. In *EuroVis 2017 - Posters*. Eurographics Association, Aire-la-Ville, Switzerland, 113–115. https://doi.org/10.2312/eurp.20171182

[2] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Nylandsted Klokmose, and Nicolai Marquardt. 2019. Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 562:1–562:28. https://doi.org/10.1145/3290605.3300792

[3] Eun Kyoung Choe, Raimund Dachselt, Petra Isenberg, and Bongshin Lee. 2019. Mobile Data Visualization (Dagstuhl Seminar 19292). *Dagstuhl Reports* 9, 7 (2019), 78–93. https://doi.org/10.4230/DAGREP.9.7.78

[4] Philipp Eichmann, Darren Edge, Nathan Evans, Bongshin Lee, Matthew Brehmer, and Christopher White. 2020. Orchard: Exploring Multivariate Heterogeneous Networks on Mobile Phones. *Computer Graphics Forum* 39, 3 (June 2020), 115–126. https://doi.org/10.1111/cgf.13967

[5] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D. Hansen, and Jonathan C. Roberts. 2011. Visual Comparison for Information Visualization. *Information Visualization* 10, 4 (Sept. 2011), 289–309. https://doi.org/10.1177/1473871611416549

[6] Jane Hoffswell, Wilmot Li, and Zhicheng Liu. 2020. Techniques for Flexible Responsive Visualization Design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1—-13. https://doi.org/10.1145/3313831.3376777

[7] Tom Horak, Philip Berger, Heidrun Schumann, Raimund Dachselt, and Christian Tominski. 2021. Responsive Matrix Cells: A Focus+Context Approach for Exploring and Editing Multivariate Graphs. *IEEE Transactions on Visualization and Computer Graphics* (Feb. 2021). arXiv:2009.03385

[8] Tom Horak, Andreas Mathisen, Clemens N. Klokmose, Raimund Dachselt, and Niklas Elmqvist. 2019. Vistribute: Distributing Interactive Visualizations in Dynamic Multi-Device Setups. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 616:1–616:13. https://doi.org/10.1145/3290605.3300846

[9] Waqas Javed and Niklas Elmqvist. 2012. Exploring the Design Space of Composite Visualization. In *Proceedings of the IEEE Pacific Symposium on Visualization*. IEEE, Piscataway, NJ, USA, 1–8. https://doi.org/10.1109/pacificvis.2012.6183556

[10] Ulrike Kister, Konstantin Klamka, Christian Tominski, and Raimund Dachselt. 2017. GraSp: Combining Spatially-aware Mobile Devices and a Display Wall for Graph Visualization and Interaction. *Computer Graphics Forum* 36 (June 2017), 503–514. https://doi.org/10.1111/cgf.13206

[11] Ricardo Langner, Tom Horak, and Raimund Dachselt. 2018. VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan. 2018), 626–636. https://doi.org/10.1109/tvcg.2017.2744019

[12] Carolina Nobre, Miriah Meyer, Marc Streit, and Alexander Lex. 2019. The State of the Art in Visualizing Multivariate Networks. *Computer Graphics Forum* 38, 3 (June 2019), 807–832. https://doi.org/10.1111/cgf.13728

[13] Carolina Nobre, Dylan Wootton, Lane Harrison, and Alexander Lex. 2020. Evaluating Multivariate Network Visualization Techniques Using a Validated Design and Crowdsourcing Approach. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376381