Interaction in the Visualization of Multivariate Networks

Michael Wybrow, Niklas Elmqvist, Jean-Daniel Fekete, Tatiana von Landesberger, Jarke van Wijk, Björn Zimmer

Interaction in the Visualization of Multivariate Networks

Michael Wybrow[1], Niklas Elmqvist[2], Jean-Daniel Fekete[3], Tatiana von Landesberger[4], Jarke J. van Wijk[5], and Björn Zimmer[6]

[1] Caulfield School of Information Technology, Monash University
Caulfield, Victoria 3145, Australia
Michael.Wybrow@monash.edu
[2] School of Electrical & Computer Engineering, 465 Northwestern Avenue, Purdue University, West Lafayette, IN 47907-2035, USA
elm@purdue.edu
[3] INRIA, Bât. 660, Université Paris-Sud, 91405 Orsay Cedex, France
Jean-Daniel.Fekete@inria.fr
[4] Technische Univeristät Darmstadt, Fraunhoferstr. 5, 64283 Darmstadt, Germany
tatiana.von-landesberger@gris.tu-darmstadt.de
[5] Department of Mathematics and Computer Science, Eindhoven University of Technology, Den Dolech 2, 5612 AZ Eindhoven, The Netherlands
vanwijk@win.tue.nl
[6] Department of Computer Science, Linnaeus University, 351 95 Växjö, Sweden
Bjorn.Zimmer@lnu.se

Abstract. Interaction is a vital component in the visualization of multivariate networks. It enables greater amounts of information to be seen and explored than is possible with static visualization. Interaction can also help show the information landscape of the data while still allowing users to find and view areas of interest in greater detail and pivot between these. In this chapter we first discuss the design space and requirements for interacting with large multivariate data sets. We describe and classify relevant interaction techniques, and give examples of the interactive aspects of multivariate graph visualization systems. We present recommendations and guidelines for designing novel interaction approaches. Finally, we describe the open challenges within the field of multivariate graph visualization as we see them.

1 Introduction

The overall aim of visualization is to obtain insight into large amounts of data. Detection of patterns as well as outliers are typical examples. For networks, such patterns can be number and position of cliques; for multivariate data this can be the correlation between attributes. The major challenge of multivariate network visualization is to understand the interplay between properties of the network and its associated data, for instance to see if the formation of cliques can be understood from attributes of nodes.

Producing useful and informative visualizations for multivariate networks is a complex and challenging task. Complexity and scalability (see Chapter **??**) are significant issues, both with respect to the graph size as well as to the number and variety of variables. It is very difficult to statically display large, complicated data sets in general, including multivariate data and networks. Occasionally it is possible to nicely encode small multivariate data sets completely in custom static visualizations, such as with Minard's seminal "Napoleon's March to Moscow" visualization [53], but this is rare.

In practice, even moderate-sized networks can be difficult to visualize without overlaps and loss of information, let alone when augmented with additional variables. Moreover, people working with visualizations can usually only comprehend a small subset of the information space at a time. It is therefore important to reduce the relevant information displayed at any point to a manageable amount in order to facilitate understanding of the main data characteristics. Thus, as the data size and complexity (i.e., the combination of dimensions and network complexity) increases, there is a need to efficiently navigate through the data and to enable discovery and communication of the data.

Interaction is a vital component in the visualization of multivariate networks. By allowing people to browse data sets with interactions like panning and zooming, we can enable much more information to be seen and explored than would otherwise be possible with static visualization. Overview-based interactions afford the user the ability to understand a complete picture of the data or information landscape and to decide where to direct her attention. Through search and filtering, interaction can reduce cognitive effort on users by allowing them to locate, focus on and understand subsets of the data in isolation. Pivoting and other navigational interactions at both the view and data level allow people to identify and then to transition between areas of interest.

While there are methods for interacting with graphs and dimensions separately, the combination of both needs special attention. The challenge is to clearly visualize multiple sets of individual dimensions as well as to offer a useful visual overview of data, and allow transitions between these to be easily understood. Moreover, we need to find ways to support users in navigating through the complex data space (graphs $\times$ dimensions) without "getting lost," and without an overburden of interaction actions that may frustrate the user.

In this chapter interaction for the visualization of multivariate networks is considered. After a discussion of the design space for interaction, existing approaches are examined, guidance for designing interactions is offered and open problems in the area are described. It is aimed at readers who are intending to visualize networks with multivariate data. They may be planning to evaluate and select some existing approaches or systems and adapt these to their needs, or they may be thinking about designing a custom visualization tailored to the needs of their data and audience. Rather than just a survey of the field, this chapter should be considered a guide to interaction for networks with multivariate data; explaining what the problems are, what is possible, what has been done before, what might be done in future.

The rest of the chapter is organized in five further sections. The next section discusses the design space and requirements for working with large multivariate data sets, including difficulties in navigating networks and dimensions. Section 3 classifies relevant interaction techniques on the basis of the stages in the standard Information Visualization Reference Model. Section 4 gives examples of the interactive aspects of multivariate graph visualization systems. Section 5 presents recommendations and guidelines for designing novel interaction approaches, including adaptation of existing interaction design principles for use in this setting. Finally, Section 6 puts forward a vision of the challenges and goals as we see them within the field of multivariate graph visualization.

2 Background

Interaction is a vital ingredient of information visualization, and has been heavily studied. In this section, we do not aim to explain in general how interaction works in visualization, as this is very well addressed by excellent books such as [58] and a large number of articles [30,39,72]. Also, we acknowledge that data exploration encompasses much more than just direct interaction with graphical representations, and includes aspects like navigation support, knowledge capture, and collaborative visualization. This area is studied in visual analytics; for an overview see Pike et al. [51].

Furthermore, for this chapter, we mostly consider interaction for standard point-and-click and keyboard interfaces on desktop computers. While multi-touch tablets are commonplace and we are seeing increasing availability of large touch-based tables and displays, there has been relatively little work designing or evaluating interaction techniques for working with large networks or multivariate data on these. This is also the case with other new technology now becoming available to consumers such as 3D displays, contactless input devices, and multi-monitor displays. We discuss this as a key ongoing challenge in Section 6.

Data exploration often involves a top down approach, as strongly summarized in the visual information seeking mantra of Ben Shneiderman [56]: "overview first, zoom and filter, details on demand". Both for network and multivariate visualization, many systems and techniques aim to satisfy this pattern. But in practice, a bottom up approach is used. For instance, in social network visualization a certain person can be the starting point for further exploration [27]; in multivariate visualization one can start from one particular item and explore items which are similar. Since these approaches are valuable, an ideal system should support both.

To describe the multiple kinds of interactions used for the visualization of multivariate networks in more detail, we use the Information Visualization Reference Model [10] (see Figure 1), which breaks down the visualization process into four stages: *raw data* or *source data*, *data tables*, *visual structures* or *visual abstractions*, and *views*. To display the raw data, several transformations have to be applied: the raw data is transformed into data tables through data transformations, the data tables into visual structures through visual mappings,

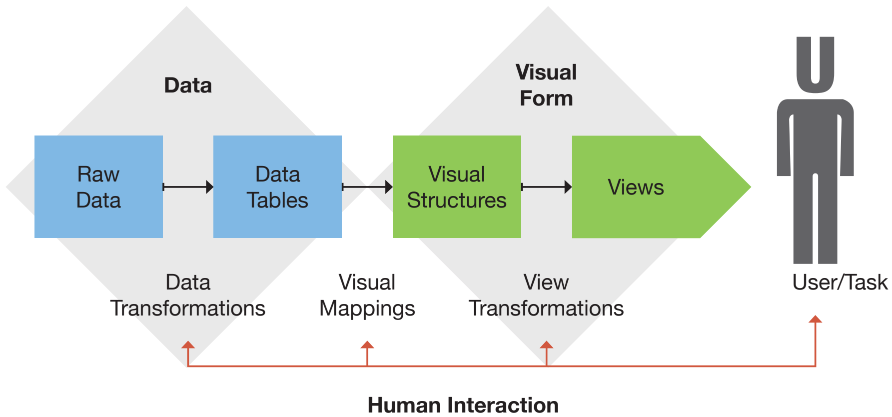

Fig. 1. The information Visualization Reference Model.

and the final rendering transforms the visual structure into an image on a view. All these transformations are performed using a multitude of specific parameters, and interaction can then be defined at the system level as *the change of transformation parameters controlled by the user with immediate feedback to the user*.

This generic model applies both to network and multivariate visualization, and many interaction techniques specifically tailored to the properties of these data types have been developed. In the next section we enumerate the most relevant of these, categorized along the stages of the reference model. However, far less techniques have been developed that specifically aim at interaction with combinations of network and multivariate data. The challenge here is to offer a simple but powerful set of interaction techniques that allows users to explore such combinations with minimal cognitive overload. On the one hand, this should be achievable, since many tasks and operations are similar at a high level; but on the other hand, standard representations of networks and multivariate data do vary largely, and also the more powerful and customized interaction methods for dealing with these data types differ greatly.

These effects can be observed for all stages of the reference model. At first sight, network data and multivariate data seem fundamentally different. However, topological aspects of network data can be nicely captured as multivariate data, simply by calculating topological metrics of nodes and edges. Also, multivariate data can be considered as networks, for instance by introducing edges between nodes that are similar, as pursued by Liu et al. [42]. Having said this, multivariate network visualization usually cannot be reduced to purely multivariate or network visualization. In fact, the combination makes analysis of multivariate networks a real challenge since discovery of an underlying phenomenon in the data can require a detailed understanding of the network topology together with the multivariate attributes, e.g., if variables represent snapshots of a flow dictated by the topology. One consequence for interaction is that users should be enabled to obtain such associated data on request. Filtering of data is a standard operation. For multivariate data this typically involves selection based on ranges of attributes, and for networks the distance from a selected set of nodes can be used.

Concerning the visual representation, network data and multivariate data can be shown separately or be combined. The use of multiple views on data is standard in visualization, and by interacting through linking and brushing, information from different views can be associated. Interaction is crucial here, but also, as both types of data are shown separately, fusion of information is often hard. One way to provide a combined view is to use a network-based approach, where nodes and edges are embellished with iconic representations of values or attributes. This limits the use of standard interaction methods for multivariate data, for instance, to select two ranges for attributes by sweeping out a rectangle in a scatterplot. Another way to combine data in one view is to use a multivariate data-based approach, for instance, by superimposing edges on top of a scatterplot. Now, standard interaction methods for multivariate data can be used, as positions of nodes encode attribute values, but also, some network interaction techniques that imply changes in the layout cannot be used anymore.

The standard approach in the view transformation stage is to provide options for zooming and panning. On the image level, this is straightforward, however, when using multiple views where the spatial dimensions have different meanings, this can be hard to deal with in a natural way.

These examples show there are basically two approaches to interacting with multivariate networks. One approach is to stick to conventional representations and dedicated interaction methods, another, more challenging but also potentially more rewarding approach is to aim for tight integration, both with respect to representation and interaction, to facilitate the understanding of the relation between network and multivariate. In the following sections these approaches are explored in more detail.

3 Classification of Interactions

We use the Information Visualization Reference Model, originally presented in [10], as the basis for our classification of interaction techniques (see Figure 1). We classify interaction techniques based on the level of this pipeline they affect. Note, the match may not be always perfect, as some techniques address multiple levels simultaneously. Where possible, we make use of standard terminology and jargon from the information visualization community in order to simplify access to related work.

Notably, our classification presents the pipeline stages in the reverse order to [10]: we describe interactions at the view-level first for pedagogical clarity, since these are simpler, easier to understand, and are sometimes extended or utilized by interactions in the remaining stages of the pipeline.

Many of the generic interaction techniques are applicable both to standard networks as well as multivariate data, and basic examples are given. As discussed in Section 2, there are many possible graph representations, the choice of which can limit the applicability of interaction techniques since these may be dependent on specific aspects of the chosen graph representation. Examples of complete systems utilizing a mix of interaction techniques to deal simultaneously with a combination of multivariate data and networks are described in Section 4.

This classification is a revised and expanded version of a similar classification of interaction techniques for network visualization appearing in [68]. Note, this is certainly not the only way to define and categorize interaction. For instance, Yi et al. advocate for a taxonomy based on user intent, and they distinguish Select, Explore, Reconfigure, Encode, Abstract/Elaborate, Filter, and Connect as main categories [72]. Similar classification has been recently also presented for cartography [54].

3.1 View-Level Interactions

The view-level interactions are mostly related to visual emphasis of interesting objects, navigation through the data set, and using Magic Lenses to augment the visualized information.

Highlighting

Highlighting transiently changes the visible rendition of items at the view-level, not at the visual encoding level. Although it can be practically implemented with support at the visual structures level, this is not required so we conceptually consider it a view level interaction.

Interactions such as search or mouse hovering may lead to highlighting of objects such as search results or linked content.

Hovering: Hovering is used in multivariate visualizations such as InfoZoom [61] that display large data tables with a smart aggregation mechanism. Rows are items, columns are attributes, and values are in cells. When the mouse passes over a value in a cell, all the cells with the same value for that attribute are highlighted, showing the frequency and distribution of this value. Hovering is even more useful with multiple views to highlight parts linked by some relation. MatrixExplorer [31] uses two linked visual representations for networks, one being a node-link diagram and the other an adjacency matrix. When the mouse hovers over an entity in one visual representation, the same entity is highlighted in the other.

Brushing and Linking: This technique involves the user watching multiple views related to the same dataset. When the pointer is moved over an item in one view, all the related items are highlighted in all the views [4, 9].

For multivariate networks, these views can use the same visual representation or a different one; they can show the same information (e.g., the network topology as a node-link diagram as well as an adjacency matrix [31]), complementary information (e.g., the network topology as a node-link diagram and nominal attributes as lists [28]), or mixed aspects (e.g., the network topology as a node-link diagram and attributes using parallel coordinates [3,60]). These can be used to more easily contrast and compare information or variables in distant places within the network or to see parts of the dataset from different perspectives. The latter is often used when visual encoding does not allow for viewing all the data in one visual representation. This is generally caused by data size (too many data points to show) or data complexity (too many data variables).

Further interaction techniques are often used to augment and enhance the use of multiple views. Some of these will be described in the visual encodings section.

Magic Lenses: Magic lenses [7] are "filters, that modify the presentation of application objects to reveal hidden information, to enhance data of interest, or to suppress distracting information." They have been used extensively in visualization of networks and multivariate data.

Excentric Labeling [20] offers an approach similar to tooltips: labels are interactively displayed over dense visualizations such as scatterplots or node-link diagrams. When enabled, they show a focal region (rectangular or circular) that follows the mouse; all the items inside the region are labeled outside of the region with a line connecting each item to its label. Bertini et al. [5] has extended upon this to give better control of the focal region and visualization of aggregated information on the focal region.

Jusufi et al. [36] describe lenses for multivariate network that display nodes as small multidimensional visualizations when they are within the focal area. They use several visualizations: parallel coordinates, bar charts, and star plots.

Navigation

Panning and zooming: Panning and zooming involve changing the visible viewport over the otherwise unchanged visualized data. These actions are usually accomplished via standard interactions with common controls like scroll bars and sliders, hardware like mouse scroll-wheels and track-pads or using multitouch at touch tables or tables.

Several navigation techniques have been designed to improve panning and zooming over large data sets, which are discussed in detail in Chapter **??**. Suffice it to say that these operations can be very cumbersome, requiring users to drag the cursor for long distances across the screen. The simplest technique to overcome that problem is to use an overview plus details representation, such as a bird's eye view of the visualization in a small window and detailed view in a large one. The viewport of the detailed view is usually displayed as a rectangle on the small window that can be manipulated for fast panning.

In graph visualization, *topology-aware graph navigation* allows automatic panning and zooming in a graph. These actions can be performed directly on the network structure, such as link sliding [46] or bring-and-go [63]. These techniques allow the user to quickly find out-of-viewport nodes that are attached to a particular node, relocate these to be temporarily positioned in their current view and then allow further navigation from them. The bring-and-go technique can also be considered as a magic lens for navigation.

View distortion (single/multiple): View distortion allocates more space to items of the users' interest. In particular, fisheye views generally allow people to see more information at a point of interest. For example, this can reveal detailed information that was initially smaller than one pixel in size.

For graphs, there are specific distortion techniques, such as Balloon Focus in a treemap [64] and a guaranteed visibility technique in dendrograms [47] that allocate more space to the nodes in focus for their detailed inspection. These techniques allow for multiple foci at the same time.

Distortion can be applied also to edges, improving the visibility of items on the screen. For example, Edge Lenses [71] interactively displace edges under the pointer in order to avoid overplotting of edges over nodes or edges over each other. Tominski et al. [63] have proposed two types of lenses to facilitate the exploration of networks: *Local Edge Lens* only show edges with vertices inside the focal region to locally reduce clutter; *Bring Neighbors Lens* transiently moves vertices that are connected to vertices in the focal area but not visible in the viewport at the boundaries of the focal area. Their lenses can also be combined. Note that the latter technique can be seen as an example of magic lenses.

The view distortion is not always geometric: *Semantic zooming* changes the visual representation and level or details according to the zoom level. The interaction technique remains the same as panning & zooming (e.g., using the mouse wheel or a zoom slider) but the visual effect of zooming is changed. Semantic zooming [50] involves changing the visual parameters by altering the amount of detail shown at various levels of zoom. The simpler kind of semantic zooming consists of showing more details when zooming in, and less when zooming out, connecting the zoom level to the data aggregation level [19]. This could involve showing more of a network at the greater zoom depth such as changing graph aggregation level [17].

3.2 Visual Structure Level Interactions

Selection: Selection interactions alter the visual parameters of the visualization. They generally result in the most basic form of encoding change in order to highlight or emphasize areas of the network. Often they modify visual attributes of the graph entities (e.g., color, size, line width, etc). Selection differs from the view-level highlighting in that it implies a state change at least at the visual structure level, sometimes even at the data level. Also, highlighting is transient and changes implicitly as the pointer moves or the search query changes, whereas selection is explicitly set on or off.

There are various ways of selecting. For graphs one can select/brush nodes directly by clicking on them, select nodes according to their network properties [6] or select items according to network attribute values. The latter is specifically suitable when analyzing multivariate networks. Moreover, the network structure can be used for an enhanced highlighting, i.e., not only the selected nodes are highlighted but also their neighbors or parent/child nodes. This can be extended with node or edge properties, where only those adjacent/connected nodes are highlighted that have certain node attribute values. An example is highlighting of controlled companies in a shareholding network [62].

McGuffin et al. [44] have described techniques to select subgraphs interactively. In addition to traditional rectangle and lasso selection of nodes, they introduce a special kind of radial menu to further control and extend the selection of nodes (e.g., extending it by increasing a radius from the current selection: add nodes at distance 2, 3, etc.) They also introduce a special kind of menu box that appears transiently to operate on the current selection for visual structure level or data level operations (e.g., align, change color, change shape, etc.).

Changing mapping of attributes: Interactions that change the visual encoding can also be used to explore and understand various dimensions of the data. An example of this is changing the visual mapping of attributes, i.e., which attributes are assigned to which visual attributes such as size and color. Such interaction should be typically provided in interactive graph visualization systems.

Even considering just classic node-link representations for networks, visual encodings and styles of these may still vary greatly. Different emphasis can be given to visual objects, such as by drawing edges faintly using a high level of transparency or displaying nodes as points without size. These choices can in turn lead to vastly different visual results for the same data. Hence, interactively varying such attributes of the visual encoding can be useful to discover different properties of the data. See [41] for some of the more extreme examples, as well as further discussion of similar techniques in Chapter **??**.

Network layout: Layout-based interactions alter the position of nodes and edges based on properties of the network. The intent is for the layout to reveal additional information about the structure of the network.

Examples of layout-altering interactions include positioning nodes and edges to emphasize similarity, such as using Multidimensional Scaling [40], or by applying existing automated graph layout algorithms. Interactions to apply layout changes are typically triggered by changing a layout setting, however layout can sometimes be adjusted by interacting directly with the network, i.e., dragging nodes or edges.

Network layout can be calculated solely in dependence of network structure [26], only in dependence on node properties [6] or a combination of both network structure and network attributes [37, 57]. The type of layout depends on the user task. If the user wishes to analyze the relationships between nodes in the network, a topology-only layout is sufficient. However, if she wishes to analyze the interconnection of network structure and network attributes (e.g., are people with similar characteristics friends?), a layout that takes both network structure and network attributes into account is preferable.

Moreover, constraint-based network layout approaches can allow interactive control and fine-tuning of the layout [16], and may be used in conjunction with multiple views and semantic zooming to allow interactive browsing and exploration of large multivariate networks [15].

Multiple differing network layouts can be coupled with multiple views and augmented with brushing and other highlighting techniques to understand the relations between them [11]. This allows the user to compare and analyze the network from different perspectives, and detect information which might have been hidden while using a single layout.

Representation: Graphs and multivariate data can be represented visually in various ways (e.g., node-link diagrams vs. adjacency matrices for graphs; scatterplot matrices vs. parallel coordinate plots for multivariate data, etc.). As one representation may not reveal the intended information on the network, the user may wish to change the representation in order to gain a better view of the data. This is done using interactions altering visual encoding of parts of the network or present alternative representations such as matrix views, tables, or even a mixed representation such as in NodeTrix [32].

3.3 Data Level Interactions

Data-based interactions involve selecting which data to show (showing more, less or completely different data) or manipulating data values (deleting, inserting data).

Selecting Data for Visualization

Filtering: For large graphs, the whole graph may not be shown on the screen. The user then can decide either to reduce the size of the displayed data set (filtering) or to expand on demand the currently shown part of the data set (adding undisplayed data). Then, data level filtering interaction enables display of just interesting subsets of the data.

Such interaction can be performed directly in the network visualization (by selecting nodes to hide) or using a query interface. The query interface can range from a simple slider for attribute values, to a histogram-based filter, right through to filtering via brushing in additional views on the data (multiple views).

Dynamic Querying: Sometimes there can be one or more important variables to focus on within the visualization. A prominent example is time. The user may wish to browse through time in the visualization of dynamic graphs. For this it is useful to provide controls allowing the user to directly move through the range of possible values. This is analogous to using sliders and other common controls to provide panning and zooming for the space dimension.

Adding undisplayed data: An alternative way of exploring large graphs is to show a small part of the graph at the beginning of the analysis process (e.g., as a result of a search for interesting nodes) and then expand this selection on demand [27]. The expansion allows the user to add undisplayed data to the network. This can be by navigating through the network topology, such as showing neighboring nodes or connections between nodes on demand [28]. In hierarchic graphs (trees), one can navigate along the hierarchy and show nodes on a lower level of hierarchy, or show only nodes at a certain level [19].

The number of possible expansions of a graph might be very large, and the user may not know which parts of the graph to expand. In such situations, it is useful to show information on which elements to display when there are more candidates than there is room to show. Such decisions are often based on a *degree of interest* function. Such functions can be calculated in many different ways (e.g., [23, 27, 29, 43]).

Search: Search-based interactions at the data level are most useful when not all of the multivariate network data can be shown at once. They allow particular entities of interest to be extracted and displayed or highlighted from the entire data set. Specific examples are to:

– search for nodes/edges with certain attribute values;
– search for nodes/edges with certain topological properties;
– search for subgraphs with specific structural properties (motifs) [67]; and
– search for graphs—interactive user interfaces for defining query graphs and searching for them [66].

Search actions may be performed in various ways. They may involve construction of textual or graphic queries, may be performed by example, or achieved by finding similar items to those in a selection drawn or otherwise specified by the user. Search interactions may result in other data level changes such as filtering and adding undisplayed data.

Pivoting: In the case that different variables are represented by different edge and node types in a heterogeneous network, pivoting is an interaction approach where the user can visualize a couple of variables at once and switch between looking at various slices of the data. Usually this involves keeping some common part of the network visible and as stable as possible during pivot actions, such as in PivotGraph [69] or PivotPaths [13].

Changing Data

In some cases, the user may wish to change the input data such as data attributes or graph structure. This can be done by direct data editing in the user interface or by aggregation.

Editing: Multiple different ways to edit graphs exist:

- Graph structure: the user may wish to edit the graph structure: delete or add nodes or edges. This is usually done directly in the visual interface by selecting nodes/edges to delete or by drawing new nodes or edges. The system can then either show these changes directly or can show the impact of these changes on the network structure [67].
- Attributes: in multivariate networks, the user may additionally change attribute values for certain nodes. Moreover, the user can run a specific algorithm which creates new attribute values that can be explored or used for navigation in the graph. This includes creating new attributes by combining existing attributes (such as sum of two attributes) or by creating attributes describing node or edge topological information (e.g., betweenness centrality).

Aggregation: Large graphs are often simplified by aggregation. Aggregation merges several nodes and/or edges to so-called supernodes or superedges, where a supernode represents several nodes and a superedge represents several edges. The user may choose to see one of the predefined graph levels (pre-defined aggregation) or define the aggregation interactively. Such aggregation can merge user-selected nodes into one node [2] or can automatically merge nodes based on user-defined node attributes [69] or on topologic network properties [67]. The aggregation based on selected network attributes is specially useful for multivariate networks. This allows for variable views on the graph and its structure.

Annotation: Annotation is an interaction where the user can add additional information to objects in the visualization in order to augment their understanding of the data and indicate or signpost points of interest. This is analogous to using notes in order to make sense of complexity, although this is arguably more valuable when it is done in-place by annotating the network directly. In this way the annotations cause changes to the data which subsequently allows the user to search, filter or otherwise interact with the annotations directly.

History and Provenance: Interactive exploration and analysis of large graphs includes many steps—interaction actions—and feedback loops. The performed interactions are then difficult to remember and reproduce. This is facilitated by tracking of user actions. GraphDice [6] records view changes and selection changes and shows them as a set of miniatures. Hovering over a miniature transiently changes the selection to use the one recorded in the history. Clicking on the miniature sets the view and selection to the recorded one. The RelaNet System tracks and automatically aggregates all user actions [65]. It then shows them to the user using a graphical representation: a tree whose nodes are visualization states and edges are actions. The user can click on a node in the tree in order to resume that previous visualization state. The user can then either replay the actions or start a new exploration path (creating a new branch in the tree). All actions can be stored, shared, and reviewed.

Recorded actions can be analyzed algorithmically or shown to the user for their visual inspection. The CZSaw system [38] keeps track of all interactions and allows the user to explore and share them.

The tracking, reproducibility and analysis of user actions is still a large challenge in visual analytics. This problem belongs to the more general issue of analytical provenance[7] addressed by systems such as VisTrails [59].

4 Exemplars

To better illustrate effective interaction techniques and methodologies for multivariate graphs, we here present four exemplars of existing InfoVis systems that include such techniques. These exemplars are the GraphDice system by Bezerianos et al. [6], the GraphTrail system by Dunne et al. [14], Parallel Node-Link Bands by Ghani et al. [22], and the state transition networks by Pretorius and van Wijk [52]. We used the following criteria when selecting these exemplars for inclusion in the chapter:

- **Representative:** Our objective was to select exemplars that capture a wide range of representative interaction techniques.
- **Significant:** The included examples all provide interaction techniques that are among the first of their kind.
- **Best practices:** All exemplars demonstrate best practices in interaction for multivariate graphs.
- **Familiar:** Our selection is by necessity limited by our knowledge, experience, and preconceptions of the general field of multivariate graph interactions.

In no way do we claim that this set of exemplars is exhaustive or optimal. There may exist several other InfoVis tools that we could have selected instead of these four. We only claim that our selection is representative and illustrative.

We use the term “analyst” to refer to a domain specialist performing analysis tasks with the system, rather than a “data analyst”.

4.1 GraphDice

GraphDice [6] is a multivariate graph visualization tool that supports navigation in data space similar to the scatterplot matrix navigation proposed in the ScatterDice [18] tool. The key contribution of GraphDice is the integration of attribute-based layout with interactive data space navigation, where both intrinsic (such as the age, gender, and annual income) as well as derived (layout position, degrees, and centrality) attributes of actors in a social network form the data space. This supports a smooth and fluid visual exploration process where users can seamlessly sculpt their queries across all attributes (see Figure 2).

In terms of specific interaction techniques for multivariate graphs, GraphDice supports the following:

[7] See `http://www.vacommunity.org/AnalyticProvenanceWorkshop` for the first workshop on this issue

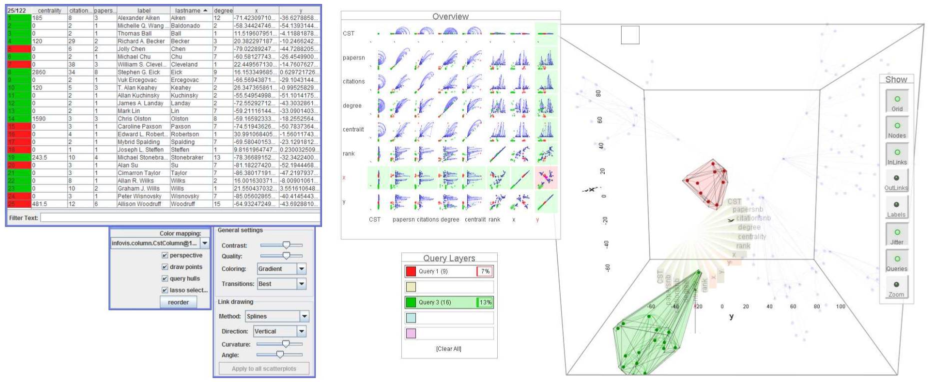

Fig. 2. The GraphDice [6] multivariate visualization tool shown visualizing an IEEE InfoVis co-authorship network consisting of both intrinsic and derived attributes. The analyst is in the process of transitioning between two different node attributes; the transition is shown as a smooth animation.

– **Smoothly changing visual mapping:** The key feature of data space navigation [18] is to smoothly change the mapping of attribute dimensions to positional (X and Y) visual variables using an animated transition. GraphDice does not discriminate between intrinsic and computed attributes, thereby allowing the analyst to transition from a geographic or computed graph layout to other attributes such as degree, centrality, age, gender, income, etc.
– **Pivoting:** Data space navigation in GraphDice also allows for pivoting a multivariate graph to study different slices, or *facets*, of the data. This interaction is inspired by PivotGraphs [69], and also incorporates node and link aggregation to minimize overplotting and to summarize a large number of data points. Similarly, GraphDice also summarizes multiple time points into intervals that are visible during pivoting.
– **Query sculpting:** Query sculpting is a faceted filtering technique that is closely integrated with the data space navigation and pivoting functionality in GraphDice. The analyst can use lasso, bounding box, or interval selection on the main node-link display to create queries in the dataset. These queries are maintained in a query control box, which also summarizes the size, distribution, and name of each query. Analysts can then use data space navigation to pivot the query, allowing them to sculpt it by adding additional constraints on other attribute dimensions.

4.2 GraphTrail

The GraphTrail [14] visual analytics tool by Dunne et al. supports exploration of graph data where the nodes and links are both multivariate—containing multiple attributes, as already prominently discussed in this chapter—as well as multimodal (called *heterogeneous* in the paper)—where the nodes or links are of different types, or *modes*. The work presents two case studies: (1) publication

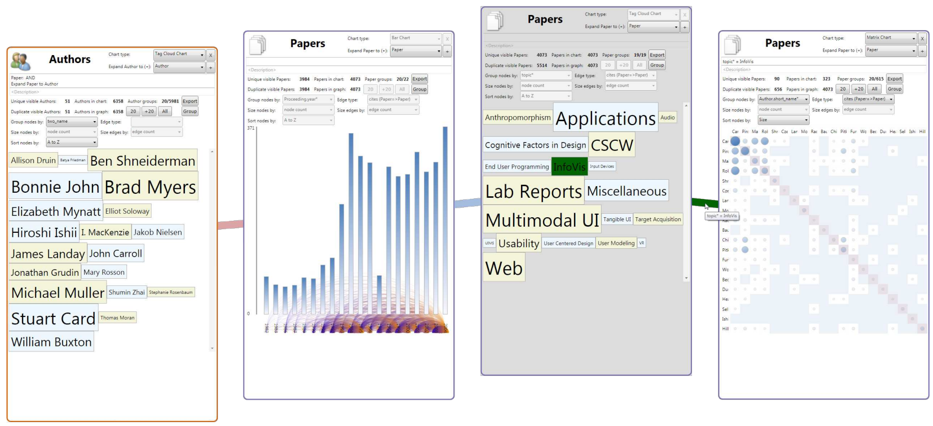

Fig. 3. GraphTrail [14] overview of a multivariate co-authorship dataset for the ACM CHI conference. The screenshot shows examples of the tag cloud, hybrid bar chart, and matrix chart supported by the tool.

data for the ACM CHI conference (Figure 3), where nodes contain attributes such as year, title, name, locations, and date, and the modes are authors, papers, and proceedings; and (2) a large-scale archeological graph of artifacts consisting of 24 different node modes and 35 link modes. The GraphTrail tool supports the following specific interaction techniques for multivariate graphs:

- **Aggregation:** The tool presents aggregated views of graphs in self-contained charts such as bar charts, tag clouds, and tables instead of the raw graph data as a traditional node-link diagram. The purpose is to use familiar and readable visual summaries as opposed to the full graph dataset.
- **Visual history:** While not strictly a multivariate graph interaction technique, GraphTrail provides an innate visual interaction history by maintaining each exploration branch as a chain, or trail, of connected charts. This allows the analyst to refer back to the exploration path, which may potentially be branching, at any time.
- **Exploratory interactions:** The tool supports three specific interaction techniques for multivariate graph exploration:
  - *Filtering and merging:* Selecting subsets of a dataset for drill-down and merging disparate subsets into a single chart using direct manipulation.
  - *Pivoting:* Transitioning between different edge and node types (i.e., modes) to explore multimodal relationship in the graph.
  - *Cloning:* Duplicating subsets and charts with dependencies to avoid having to propagate upstream changes to connected child charts.

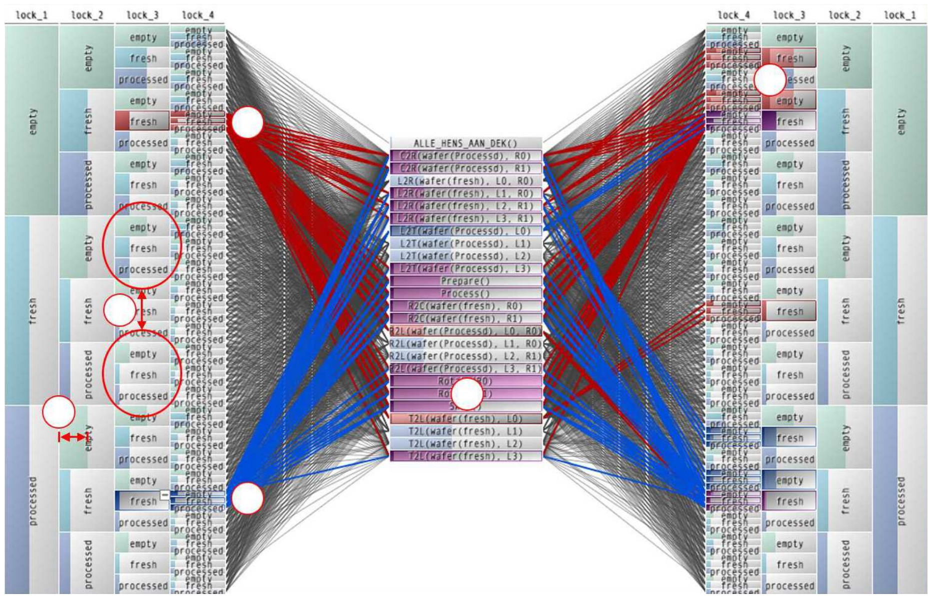

Fig. 4. A multivariate graph visualization designed by Pretorius and Van Wijk [52] of a state transition graph created by a system analyst.

4.3 State Transition Networks

Pretorius and van Wijk [52] present a multivariate graph visualization technique for visual inspection of state transition graphs (Figure 4). Such graphs are common for complex systems and are often used for design, debugging, and evaluation. The visual representation is based on separating the different modes in these state transition graphs and showing two modes at a time, essentially as a bipartite graph layout with nodes on each side of the display and the links (and edge labels) connecting the modes in-between. The implementation allows the user to navigate between which two modes they wish to drill down into. In addition, the system supports several dedicated multivariate graph interaction techniques:

- **Selection and highlighting:** A simple but key interaction technique is the ability to select a node (or a cluster of nodes) in the visual representation, causing all contained or connected nodes and edges to be highlighted in red. This interaction is coupled with appropriate visual representations that also highlight multivariate attributes in the connecting edges.
- **Filtering:** In conjunction with the selection technique, the multivariate graph implementation also supports adding to or subtracting from the selection to further refine the analyst's exploration.
- **Clustering:** To cope with the large scale of the state transition graphs, the prototype implementation supports clustering and aggregating nodes and edges based on attributes and labels.

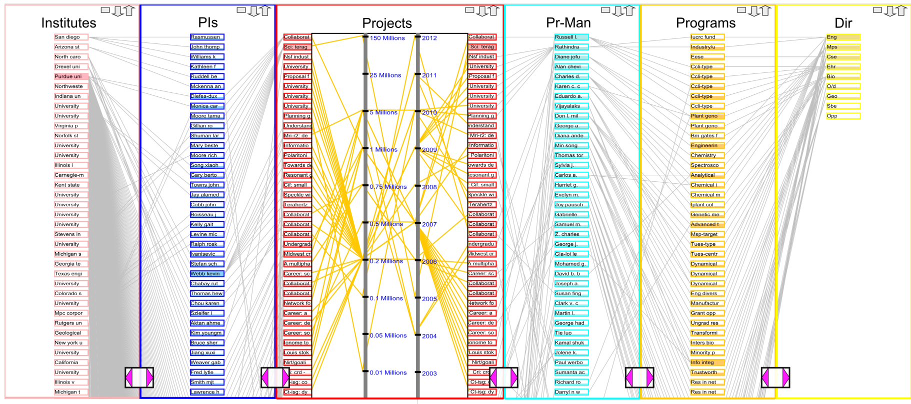

Fig. 5. The parallel node-link bands (PNLBs) [22] technique visualizing a multimodal NSF funding graph. The parallel coordinate inset is a specific interaction technique called "open sesame" for drilling down into one or several scalar attributes of a set of nodes; in this case, it is visualizing the year and amount for funded projects.

4.4 Parallel Node-Link Bands

Similar to GraphTrail, the parallel node-link bands (PNLBs) [22] method is a graph visualization technique for multimodal and multivariate graphs, i.e., graphs where nodes and links not only have multiple attributes, but also belong to two or more different modes, or types. However, instead of focusing on aggregated charts summarizing the network, PNLBs draw on the work by Pretorius and van Wijk [52] (discussed above) to retain the node-link visual metaphor but separates the nodes by their respective mode into specific *bands* organized by slicing the viewport vertically (Figure 5). Unlike Pretorius and van Wijk, PNLBs generalize to any number of bands, although they only show links between adjacent bands, suppressing all other link modes to minimize visual clutter. For this reason, the technique also borrows many ideas from semantic substrates [57], where node modes are organized into spatially disjoint substrates. However, PNLBs were designed specifically for multivariate graph exploration, and provide the following interaction techniques to support this goal:

- **View distortion:** PNLB bands can be zoomed and panned independently of each other; furthermore, they can also be designed to support semantic zooming. One particular use of this is to enable view distortion where a fisheye function around a selection or the user's mouse cursor can smoothly expand and compress the visual marks representing the nodes in the graph.
- **Multivariate drill-down:** The tool allows the analyst to drill down into entire bands or individual nodes to uncover the multivariate attributes "hidden" in the data. For example, tag cloud and details-on-demand popups can show summaries or the full details of a node. Furthermore, a specific interaction technique dubbed "open sesame" (visible in Figure 5) integrates

a parallel coordinate inset within an expanded node band to show quantitative data for those nodes; parallel coordinates were chosen because they closely mimic the overall visual design of PNLBs, but other chart types can be integrated as well.
- **Multimodal drill-down:** Another drill-down option focuses on the topological nature of the graph by exposing the within-network relations within the dataset, i.e., the links that connected nodes of the same mode. This is a necessary mechanism since the PNLBs technique is designed to primarily show between-mode links for adjacent mode bands.

5 Recommendations and Guidelines

When designing or evaluating interactive visualizations for multivariate network data, it is useful to consider potential interaction techniques with regard to their usability. Indeed, this is even more important when testing novel unproven interaction techniques. There exists a large body of experience in the Human-Computer Interaction community regarding the usability of interaction techniques. In this section, we describe some well accepted usability principles and interaction design guidelines and discuss them within the context of multivariate network visualization. The information in this section should provide a useful lens through which to assess the appropriateness of particular interaction techniques.

We group these guidelines into three broad categories—Learnability, Flexibility and Robustness—as suggested by Dix et al. [12]. We also draw from other sources, including Cognitive Dimensions of Notations [8, 25], which offers some useful vocabulary for discussing design and choice of interactions as well as evaluating the impact a design will have on users.

5.1 Learnability

Learnability describes a set of principles that can be used to determine the ease with which a new user can begin productive work with the system [12]. This is especially important when designing interactions for multivariate network visualization since these will often present a large amount of data, and complex interfaces for exploring the dimensions of the data. Hence, anything that can be done to help users quickly learn the system and accompanying interactions is crucial.

One important aspect of Learnability is **Predictability**, which simply states that a given interaction should always behave predictably, i.e., exhibit deterministic behavior. Also, it should exhibit **Consistency** in that an interaction that can be performed on one element of the visualization should be able to be applied to other objects and produce similar results. This means sticking with established conventions for network and multivariate visualization, such as those described earlier in this chapter.

Other aspects of Learnability are **Familiarity** and **Generalizability** which deal with creating interfaces and interactions that map as closely as possible to the real world or similar interfaces the user will already be familiar with. Ideally, this is done to maximize utilization of users' past experience. This includes making use of familiar metaphors where applicable, as discussed at length in Chapter **??**.

Affordance [48] describes the ability of physical or digital objects to suggest how they may be interacted with through their appearance. For example, the handles on a drawer afford the user the ability to pull out the drawer. Similarly, the appearance of standard GUI controls like sliders and buttons suggest how users may interact with them.

When designing novel interfaces for working with multivariate network visualizations it might not always be possible to give interactive controls or elements these obvious affordances due to the complexity of the interface, but the appearance of controls can still sometimes be enhanced in subtle ways to hint at their intended use, or at least at the possibly for interaction. For example, with colored hyperlinks in web documents, it is not always obvious what effect clicking a link with have, but the user knows it will do something and can make an educated guess based on the link text and surrounding context. Affordance is related to the discoverability of interactive capabilities in the interface; with the profusion of graphical entities shown by visualization, it is always tempting to provide interactions contextual to specific entities, but without affordance, the chance that a novice user will discover it by chance is very low. When designing user interface components or assigning contextual interactions to graphical entities, ask yourself if they adequately express their role to the user?

Avoiding **Hidden Dependencies** [8, 25] means that the link between connected items should be visible and obvious to the user. This can be a problem in any system with filtering and search where the visualization may just show a subset of the results. Ideally, if the information linking matching elements is significant the visualization can show the smallest subset of the network that connects these nodes and edges in the search results. This becomes a greater problem when an interaction leads to a surprising result due to such a hidden dependency.

Supporting **Progressive Evaluation** [8, 25] means allowing users to be able to take a break at any time and take stock of their progress so far. This is especially important for exploratory tasks involving novice users.

5.2 Flexibility

Flexibility groups a set of principles that deal with best practices for the avenues for information exchange between the user and visualization system.

Cognitive load [45] is based on the fact that humans can hold relatively little information in short term memory—famously, seven plus/minus two pieces of information. As a result, we must consider the amount of information that needs to be retained in working memory in order to effectively work with a system or interface. As much as possible there is a need to alleviate the user having to commit unnecessary information to memory. In terms of visualizations of multivariate networks, this means relevant data should be highly visible and the interface should make clear what data and attributes the users are looking at, how they reached this point, as well as how they can return to earlier points of exploration. It should be possible to delegate the task of remembering information for complex processing tasks to the system where these details should be presented in a fashion that is easy for the users to understand.

Cognitive load and other limits of visualization are discussed in more detail in the chapter on scalability considerations (Section **??**).

Fitts's law [21] is a model that describes the act of pointing. It says that the time taken to rapidly move a pointing device to a target object is proportional to the size of the target and the distance to it. The implication of this for interaction and interface design is that the most frequently used controls should be the closest and largest. In the case of visualizations for multivariate networks, a prerequisite for answering the question of where to put the controls would be understanding the kinds of tasks that users were going to perform most commonly with the visualization, since context matters [1].

Visualizations should have low **Viscosity** [8, 25], i.e., common tasks should be able to be accomplished with a minimal number of actions or effort on the part of the users.

Abstraction [8,25] involves providing shortcuts to the user in order to facilitate them working efficiently with logical sets of the data at once. This is often vital in multivariate network data, since the aim is to allow the user to manipulate the visualization at the level of a particular attribute or dimensions of the data rather than forcing them to interrogate the properties associated with individual nodes and edges themselves. Abstractions should be used where possible, since these simplify many tasks and help with understanding the network and associated variables.

Terseness and **Diffuseness** [8, 25] state that it is important to dedicate appropriate amounts of display space to the various elements of the visualization, rather than devoting too much or too little space to them. This may seem obvious, but you should think about and question the space that is being used to show various elements of the data set.

An important general quality to strive for in designing interactive visualizations of multivariate networks is providing good **Guidance**, both in terms of dimensions and graph structure. That is, when the user can not currently see some particular dimension of the data or a section of network, we would like to let them know this information exists and also give them some estimation of the importance of the non-visible information.

5.3 Robustness

Robustness principles relate to how the system supports the user in accomplishing their goals

Direct Manipulation [55] describes interaction that is performed directly on objects and provides continuous, fast feedback in response to change. Another way of thinking about this is providing **continuity** and thus avoiding abrupt changes that could potentially confuse the user. When an action is not being performed by the user directly, it can sometimes be smoothly animated to achieve a similar effect. Direct manipulation interactions map more closely to object behavior in the real world and thus has a few important benefits; it allows users to quickly determine or predict the final outcome of their action, it allows them to more easily realize when performing an action would lead to unintended consequences, and to more easily reverse an action. It can also allows users to reach a desired state in a single action that would otherwise require several actions when only seeing incremental effects of the system after each individual action.

Recoverability [12] is an important robustness principle for most user interaction. It suggests that our visualization systems should easily allow the user to undo any action they have made in error. As we mentioned before, **responsive** interactions such as **direct manipulation** approaches also help in this regard.

Premature Commitment [8, 25] means users should not have to make any decision before they have adequate information to base it upon. This can often be solved by providing a flexibility to the system where the user can reach a particular result or view of the data via multiple paths, rather than just a single specific sequence of actions. Also, the user should have the ability to try things out without committing to them (**Provisionality**).

Error-Proneness [8, 25] describes the ability of the system to induce errors from the user and not protect them from these mistakes. When evaluating an interface to a complicated multivariate network visualization we must consider what is being shown to the user. Could they easily mistake or confuse some aspect of the visualization and reach an incorrect inference or conclusion? We want to avoid this.

Finally, it should be noted that creating effective interactive visualizations for multivariate network data is a difficult task that combines the inherent complexity of navigating large graphs with understanding and exploring multiple dimensions. As noted by Pike et al. [51], it is important to remember that the purpose of interaction is to enable an analytic discourse during which users build, test and refine knowledge. Hence, the design of new interaction approaches should carefully focus on the likely aims, intentions and actions of the user. Designs should also be formally evaluated through user studies and have their effectiveness tested with real users.

6 Challenges and Vision

This chapter has discussed and classified various interactive techniques for multivariate and network visualization. It has explored their use in several effective multivariate network visualization systems, and has described guidelines for designing successful interaction approaches. Here we conclude the chapter by outlining what we see as the major unsolved challenges in this space, and offering our thoughts on where research in this field might head in coming years.

We describe several challenges, which we broadly group into data type, data exploration, user interfaces and evaluation categories.

Data characteristics

Scalability: Dealing with scalability is obviously a major challenge. While interaction can help with some of the issues, it is often complicated by scale. For example, can we fully show both dimensions and network data, or must we reduce the complexity of the data and its presentation? Is interaction in real time still possible on large data sets? Scalability considerations for multivariate network visualization are be discussed in detail in Chapter **??**.

Temporal data: Another challenge is in dealing with multivariate networks with a temporal dimension. Given that humans have a particular understanding of time, it can be useful to leverage this familiarity and treat it specially when designing interactive visualizations. This is explored in detail in Chapter **??**.

Data Exploration and Comprehension

Understanding the information landscape: The first big challenge we see is that visualization research and systems often give good individual detailed views of particular facets of the data, but don't necessarily offer a visual interface that allows understanding of the entire information landscape at a high level. The difficulty is in giving enough of a sense of what the data is, conveying its meaning, as well as hinting at the dimensions or facets of the data that could be worth exploring in greater depth.

In connection to this, there needs to be more work on automatic and semi-automatic identification of important nodes and dimensions for directing the user during analysis tasks, as well as subsequent evaluation of such approaches.

Provenance of exploration process: We think a significant unaddressed challenge is supporting provenance in multivariate network visualizations. This involves assisting the user in tracking the exploration process, and the history of their interactions with the system. This is important since these actions form a critical part of the analysis process. While the information visualization community is aware of the importance of recording and showing which interactions were performed and when [24, 49, 51, 65], these methods either do not support multivariate networks or only to a small extent (e.g., selecting variables for visual mapping and search).

Building Interactive User Interfaces

Ad-hoc design of user interfaces: Currently, many multivariate network visualizations are built by extending upon existing systems using traditional interfaces. This is generally not desirable when the interfaces are required to encapsulate so much complexity. Ideally, we would like to see more approaches utilising principled design from the beginning. That is, designing interfaces and interactions specifically to support required application rather than just bolting additional controls and complexity onto existing interfaces. Specifically, this means techniques should be designed based on principles and guidelines like those presented in Section 5, but also be given formal user testing to prove their effectiveness.

In the case where multivariate network visualizations make use of **multiple views** there is a challenge in providing elegant and simple mechanisms to manage them. This is important since a single view will never be adequate to explore large multivariate networks. Users will always spend significant time controlling, comparing and navigating between views. Additionally, particular application domains often require their own unique multiple view configurations that are still a challenge to build interactively [70].

This is not to say that we can't have a set of general purpose visualization components and reuse them, but in order for such objects to be useful they will require standardized, consistent interaction. We need not just a common vocabulary and behavior, but also an understanding of their specificity and efficiency for particular uses.

Emerging hardware: Multivariate network visualization could potentially make use of emerging hardware such as multi-touch tables and tablets, contactless input devices (Microsoft Kinect, Leap Motion Controller), wall-sized displays, and even 3D stereoscopic displays or immersive cave environments. Additionally, there is the possibility of building novel gadgets or devices specifically for interacting with this sort of data and visualization.

Of course, these technologies are exciting to use or witness for the first time but there are a lot of unknowns surrounding their use. Do more pixels help solve the problems we have discussed? Can we benefit from extra dimension in 3D without the typical downsides, such as users becoming disoriented or lost? Can we utilize navigational multi-touch gestures? Can multimodal input provide extra benefits specifically for navigating multidimensional graphs? How can we support collaborative data exploration of multivariate networks between multiple simultaneous local or remote users? Should we build one visualization application for a specific piece of hardware (e.g., a multi-touch tablet) or can we design interactions that will adapt to a range of input devices and output capabilities?

The description of the input side of interactive applications is still at its infancy in HCI, and the management of powerful interaction approaches is relatively new in the visualization field. Adapting visualization interactions to the new setups is therefore a huge challenge [33–35]. This issue is beginning to get research attention. New ideas such as proximity-based interaction on high-walls [34] and new models for interactions across devices [35] are emerging. Research in this area will likely begin by exploring specific hardware configurations and progressively evolve towards some unification for classes of technologies over time.

Evaluation

There is a real need for formal **evaluation** of interaction approaches used for visualization of multivariate networks. There are a range of issues and concerns here. Firstly, there needs to be more consideration of the tasks for which the visualizations are to be used. Do we know what users really need when dealing with multivariate networks? How does this change when they are exploring vs. checking a hypothesis vs. using the visualization to convince another person of some fact? Chapter **??** provides an overview of tasks connected to analysis of multivariate networks, in particular.

Specifically for interaction, we need to evaluate additional factors. For example, when is interaction supportive and when can it become a burden? We need more evaluation of techniques supporting exploration and offering guidance so that the users do not get “lost” in the data space.

Evaluation results and experience partly exist for networks. However, the approaches and studies do not explicitly include and deal with multivariate networks. We need to enhance our understanding around faceted exploration along both multivariate and network data at the same time, since much of what we know applies only to exploration within a single dimension.

Acknowledgements

The genesis and planning of this chapter took place at the Dagstuhl Seminar #13201 “Information Visualization—Towards Multivariate Network Visualization” held in May 2013. We wish to thank Guy Melançon (Université Bordeaux 1, Bordeaux, France) and Robert Kosara (Tableau Software, United States) for their useful contributions during these discussions. Michael Wybrow was supported by the Australian Research Council (ARC) Discovery Project grant DP110101390.

References

1. Appert, C., Beaudouin-Lafon, M., Mackay, W.E.: Context matters: Evaluating interaction techniques with the CIS model. In: Fincher, S., Markopoulos, P., Moore, D., Ruddle, R. (eds.) People and Computers XVIII Design for Life, pp. 279–295. Springer London (2005)
2. Archambault, D., Munzner, T., Auber, D.: GrouseFlocks: Steerable exploration of graph hierarchy space. IEEE Transactions on Visualization and Computer Graphics 14(4), 900–913 (2008)
3. Auber, D., Archambault, D., Bourqui, R., Lambert, A., Mathiaut, M., Mary, P., Delest, M., Dubois, J., Melançon, G.: The Tulip 3 Framework: A scalable software library for information visualization applications based on relational data. Technical Report RR-7860, INRIA (Jan 2012)
4. Becker, R.A., Cleveland, W.S.: Brushing scatterplots. Technometrics 29(2), 127–142 (May 1987)
5. Bertini, E., Rigamonti, M., Lalanne, D.: Extended excentric labeling. In: Proceedings of the 11th Eurographics / IEEE - VGTC conference on Visualization. pp. 927–934. EuroVis’09, Eurographics Association, Aire-la-Ville, Switzerland (2009)

6. Bezerianos, A., Chevalier, F., Dragicevic, P., Elmqvist, N., Fekete, J.D.: GraphDice: A system for exploring multivariate social networks. Computer Graphics Forum 29(3), 863–872 (2010)
7. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., DeRose, T.D.: Toolglass and magic lenses: the see-through interface. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques. pp. 73–80. SIGGRAPH '93, ACM, New York, NY, USA (1993)
8. Blackwell, A.F., Britton, C., Cox, A.L., Green, T.R.G., Gurr, C.A., Kadoda, G.F., Kutar, M., Loomes, M., Nehaniv, C.L., Petre, M., Roast, C., Roe, C., Wong, A., Young, R.M.: Cognitive dimensions of notations: Design tools for cognitive technology. In: CT '01: Proceedings of the 4th International Conference on Cognitive Technology. pp. 325–341. Springer (2001)
9. Buja, A., McDonald, J.A., Michalak, J., Stuetzle, W.: Interactive data visualization using focusing and linking. In: Proceedings of the 2nd Conference on Visualization. pp. 156–163. VIS '91, IEEE Computer Society Press, Los Alamitos, CA, USA (1991)
10. Card, S., Mackinlay, J., Shneiderman, B.: Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann Publishers (1999)
11. Collins, C., Carpendale, S.: VisLink: Revealing relationships amongst visualizations. IEEE Transactions on Visualization and Computer Graphics 13(6), 1192–1199 (2007)
12. Dix, A., Finlay, J., Abowd, G.D., Beale, R.: Human-Computer Interaction (3rd Edition). Prentice-Hall, Inc. (2003)
13. Dörk, M., Riche, N.H., Ramos, G., Dumais, S.: PivotPaths: Strolling through faceted information spaces. IEEE Transactions on Visualization and Computer Graphics 18(12), 2709–2718 (2012)
14. Dunne, C., Riche, N.H., Lee, B., Metoyer, R., Robertson, G.: GraphTrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history. In: Proceedings of the ACM Conference on Human Factors in Computer Systems. pp. 1663–1672 (2012)
15. Dwyer, T., Marriott, K., Schreiber, F., Stuckey, P., Woodward, M., Wybrow, M.: Exploration of networks using overview+detail with constraint-based cooperative layout. IEEE Transactions on Visualization and Computer Graphics 14(6), 1293–1300 (2008)
16. Dwyer, T., Marriott, K., Wybrow, M.: Dunnart: A constraint-based network diagram authoring tool. In: Proceedings of the International Conference on Graph Drawing. pp. 420–431. Springer (2009)
17. Elmqvist, N., Do, T.N., Goodell, H., Henry, N., Fekete, J.D.: ZAME: Interactive large-scale graph visualization. In: Proceedings of the IEEE Pacific Symposium on Visualization. pp. 215–222 (2008)
18. Elmqvist, N., Dragicevic, P., Fekete, J.D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. IEEE Transactions on Visualization and Computer Graphics 14(6), 1141–1148 (2008)
19. Elmqvist, N., Fekete, J.D.: Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. IEEE Transactions on Visualization and Computer Graphics 16(3), 439–454 (2010)
20. Fekete, J.D., Plaisant, C.: Excentric labeling: dynamic neighborhood labeling for data visualization. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems. pp. 512–519. CHI '99, ACM, New York, NY, USA (1999)
21. Fitts, P.M., Peterson, J.R.: Information capacity of discrete motor responses. Journal of Experimental Psychology 67, 103–112 (1964)

22. Ghani, S., Kwon, B.C., Lee, S., Yi, J.S., Elmqvist, N.: Visual analytics for multimodal social network analysis: A design study with social scientists. IEEE Transactions on Visualization and Computer Graphics 19(12), 2032–2041 (2013)
23. Gladisch, S., Schumann, H., Tominski, C.: Navigation recommendations for exploring hierarchical graphs. In: 9th International Symposium on Visual Computing (ISVC'13). p. to appear. Advances in Visual Computing, Springer (2013)
24. Gotz, D., Zhou, M.X.: Characterizing users' visual analytic activity for insight provenance. Information Visualization 8(1), 42–55 (2009)
25. Green, T.R.G.: Cognitive dimensions of notations. In: Sutcliffe, A., Macaulay, L. (eds.) Proceedings of the 5th Conference of the British Computer Society, Human-Computer Interaction Specialist Group - People and Computers V. pp. 443–460. Cambridge University Press, New York, NY, USA (1989)
26. Hachul, S., Jünger, M.: Large-graph layout algorithms at work: An experimental study. Journal of Graph Algorithms and Applications 11(2), 345–369 (2007)
27. van Ham, F., Perer, A.: "Search, Show Context, Expand on Demand": Supporting large graph exploration with degree-of-interest. IEEE Transactions on Visualization and Computer Graphics 15(6), 953–960 (2009)
28. Heer, J., Boyd, D.: Vizster: Visualizing online social networks. In: IEEE Symposium on Information Visualization. pp. 32–39 (2005)
29. Heer, J., Card, S.K.: DOITrees revisited: scalable, space-constrained visualization of hierarchical data. In: Proceedings of the ACM Conference on Advanced Visual Interfaces. pp. 421–424 (2004)
30. Heer, J., Shneiderman, B.: Interactive dynamics for visual analysis. Commun. ACM 55(4), 45–54 (Apr 2012)
31. Henry, N., Fekete, J.D.: MatrixExplorer: a dual-representation system to explore social networks. IEEE Transactions on Visualization and Computer Graphics 12(5), 677–684 (2006)
32. Henry, N., Fekete, J.D., McGuffin, M.J.: NodeTrix: a hybrid visualization of social networks. IEEE Transactions on Visualization and Computer Graphics 13(6), 1302–1309 (2007)
33. Isenberg, P., Carpendale, S., Bezerianos, A., Henry, N., Fekete, J.D.: CoCoNutTrix: collaborative retrofitting for information visualization. IEEE Comput. Graph. Appl. 29(5), 44–57 (Sep 2009)
34. Jakobsen, M.R., Sahlemariam Haile, Y., Knudsen, S., Hornbaek, K.: Information visualization and proxemics: Design opportunities and empirical findings. Visualization and Computer Graphics, IEEE Transactions on 19(12), 2386–2395 (2013)
35. Jansen, Y., Dragicevic, P.: An interaction model for visualizations beyond the desktop. Visualization and Computer Graphics, IEEE Transactions on 19(12), 2396–2405 (2013)
36. Jusufi, I., Dingjie, Y., Kerren, A.: The Network Lens: Interactive exploration of multivariate networks using visual filtering. In: Information Visualisation (IV), 2010 14th International Conference. pp. 35–42. IEEE Computer Society (2010)
37. Jusufi, I., Kerren, A., Zimmer, B.: Multivariate network exploration with JauntyNets. In: Proceedings of 17th International Conference on Information Visualisation (IV '13). pp. 19–27. IEEE Computer Society Press (2013)
38. Kadivar, N., Chen, V., Dunsmuir, D., Lee, E., Qian, C., Dill, J., Shaw, C., Woodbury, R.: Capturing and supporting the analysis process. In: Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on. pp. 131–138 (2009)
39. Kerren, A., Schreiber, F.: Toward the role of interaction in visual analytics. In: Proceedings of the 2012 Winter Simulation Conference (WSC '12). pp. 420:1–420;13. IEEE Computer Society Press (2012)

40. Kruskal, J., Wish, M.: Multidimensional Scaling. No. 11 in 07, SAGE Publications (1978)
41. Lima, M.: Visual Complexity: Mapping Patterns of Information. Princeton Architectural Press (2011)
42. Liu, Z., Navathe, S.B., Stasko, J.T.: Network-based visual analysis of tabular data. In: IEEE VAST. pp. 41–50 (2011)
43. May, T., Steiger, M., Davey, J., Kohlhammer, J.: Using signposts for navigation in large graphs. Computer Graphics Forum 31(3), 985–994 (2012)
44. McGuffin, M.J., Jurisica, I.: Interaction techniques for selecting and manipulating subgraphs in network visualizations. IEEE Transactions on Visualization and Computer Graphics 15(6), 937–944 (Nov 2009)
45. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review 63, 81–97 (1956)
46. Moscovich, T., Chevalier, F., Henry, N., Pietriga, E., Fekete, J.D.: Topology-aware navigation in large networks. In: Proceedings of the ACM Conference on Human Factors in Computing Systems. pp. 2319–2328 (2009)
47. Munzner, T., Guimbretière, F., Tasiran, S., Zhang, L., Zhou, Y.: TreeJuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. In: ACM SIGGRAPH 2003 Papers. pp. 453–462. SIGGRAPH '03, ACM (2003)
48. Norman, D.A.: The design of everyday things. Basic Books (2002)
49. North, C., Chang, R., Endert, A., Dou, W., May, R., Pike, B., Fink, G.: Analytic provenance: process+interaction+insight. In: Extended Abstracts of the ACM Conference on Human Factors in Computing Systems. pp. 33–36 (2011)
50. Perlin, K., Fox, D.: Pad: an alternative approach to the computer interface. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques. pp. 57–64. SIGGRAPH '93, ACM, New York, NY, USA (1993)
51. Pike, W.A., Stasko, J., Chang, R., O'Connell, T.A.: The science of interaction. Information Visualization 8(4), 263–274 (2009)
52. Pretorius, A.J., van Wijk, J.J.: Visual inspection of multivariate graphs. In: Proceedings of the Eurographics / IEEE - VGTC conference on Visualization. pp. 967–974 (2008)
53. Robinson, A.H.: The thematic maps of Charles Joseph Minard. Imago Mundi 21, 95–108 (1967)
54. Roth, R.E.: An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. Visualization and Computer Graphics, IEEE Transactions on 19(12), 2356–2365 (2013)
55. Shneiderman, B.: Direct manipulation: A step beyond programming languages. IEEE Computer 16(8), 57–69 (1983)
56. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proc. Int. Symp. on Visual Languages. pp. 336–343 (1996)
57. Shneiderman, B., Aris, A.: Network visualization by semantic substrates. IEEE Transactions on Visualization and Computer Graphics 12(5), 733–740 (2006)
58. Shneiderman, B., Plaisant, C.: Designing the User Interface — Strategies for Effective Human-Computer Interaction (5. ed.). Addison-Wesley (2010)
59. Silva, C., Freire, J., Callahan, S.: Provenance for visualizations: Reproducibility and beyond. Computing in Science Engineering 9(5), 82–89 (2007)
60. Smoot, M.E., Ono, K., Ruscheinski, J., Wang, P.L., Ideker, T.: Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 27(3), 431–432 (2011)

61. Spenke, M., Beilken, C., Berlage, T.: FOCUS: the interactive table for product comparison and selection. In: Proceedings of the 9th annual ACM symposium on User interface software and technology. pp. 41–50. UIST '96, ACM, New York, NY, USA (1996)
62. Tekušová, T., Kohlhammer, J.: Visual analysis and exploration of complex corporate shareholder networks. In: Proceedings of the SPIE Conference on Visualization and Data Analysis. pp. 68090F–68090F. International Society for Optics and Photonics (2008)
63. Tominski, C., Abello, J., van Ham, F., Schumann, H.: Fisheye tree views and lenses for graph visualization. In: Proceedings of the Internationl Conference on Information Visualization. pp. 17–24 (2006)
64. Tu, Y., Shen, H.W.: Balloon focus: a seamless multi-focus+ context method for treemaps. IEEE Transactions on Visualization and Computer Graphics 14(6), 1157–1164 (2008)
65. von Landesberger, T., Fiebig, S., Bremm, S., Kuijper, A., Fellner, D.: Handbook of human centric visualization. chap. Interaction Taxonomy for Tracking of User Actions in Visual Analytics Applications, pp. 653–670. Springer (2014)
66. von Landesberger, T., Bremm, S., Bernard, J., Schreck, T.: Smart query definition for content-based search in large sets of graphs. In: Proceedings of EuroVAST. pp. 7–12. European Association for Computer Graphics (Eurographics), Eurographics Association, Goslar (2010)
67. von Landesberger, T., Görner, M., Rehner, R., Schreck, T.: A system for interactive visual analysis of large graphs using motifs in graph editing and aggregation. In: Proceedings of the Vision Modeling Visualization Workshop. pp. 331–339 (2009)
68. von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J., Fekete, J.D., Fellner, D.W.: Visual analysis of large graphs: State-of-the-art and future research challenges. Computer Graphics Forum 30(6), 1719–1749 (2011)
69. Wattenberg, M.: Visual exploration of multivariate graphs. In: Proceedings of the ACM Conference on Human Factors in Computing Systems. pp. 811–819 (2006)
70. Weaver, C.: Building highly-coordinated visualizations in Improvise. Information Visualization, IEEE Symposium on 0, 159–166 (2004)
71. Wong, N., Carpendale, S., Greenberg, S.: Edgelens: An interactive method for managing edge congestion in graphs. In: Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on. pp. 51–58. IEEE (2003)
72. Yi, J.S., Kang, Y.a., Stasko, J.T., Jacko, J.A.: Toward a deeper understanding of the role of interaction in information visualization. IEEE Transactions on Visualization and Computer Graphics 13(6), 1224–1231 (2006)