Network Visualization

November 2, 2018

Contents

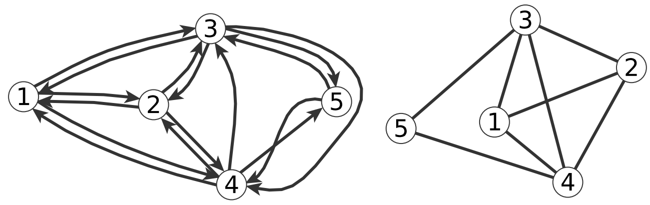
1 Introduction

Data visualization is the art and science of mapping data to graphical variables in a way that facilitates the identification of individual values and aggregate patterns. The main motives are data exploration for analysts and communication of information toward a recipient. One should not underestimate, however, some of the more circumstantial aspects of visualization: decorative appeal, symbolism, and suggestiveness.

Networks pose unique challenges to data visualization because of inherent trade-offs and dependencies among the elements in a graphical mapping. A network diagram of a subway system, for instance, should facilitate travel planning so that finding stations and following lines takes precedence over

y	1	2	3	4	5
1	0	1	1	1	0
2	1	0	1	1	0
3	1	1	0	1	1
4	1	1	1	0	1
5	0	0	1	1	0

(a) $y \in \{0,1\}^{n \times n}$



(b) $E \subseteq V \times V$ (c) $E \subseteq \binom{V}{2}$

Figure 1: A network y and visualizations of the associated directed and undirected graphs $G(y) = (V, E)$ with different layouts and edge styles.

geographical accuracy. Changing the shape of subway lines, however, influences where stations can be placed. Layout problems such as this become even more challenging when networks are large. In addition to visual clutter, for instance, computational complexity of the layout algorithms is a concern. Many other challenges beyond layout need to be addressed, including additional information associated with nodes and links and alternative representations such as adjacency matrices.

Our contribution to this volume is split into two main parts. The first part, in Section 2, provides a high-level overview of the main challenges associated with the visualization of networks and a glimpse at some of the more common techniques to address them. In Section 3, we look more concretely at the visualization of networks from two different applications domains and how it is enabled by interactive software tools. The chapter concludes with opportunities for future work in network visualization.

2 Principles

Adopting the notation used throughout this book, we define a (binary) network with n nodes as a (binary) data matrix $y \in \{0,1\}^{n \times n}$. More general situations are considered briefly in Section 2.5.

Network visualizations are commonly produced using techniques developed for graphs. A *graph representation* $G(y) = (V, E)$ of a network y consists of the set $V = \{1, \ldots, n\}$ of *vertices* and the set $E = \{(i,j) \in V \times V : y_{ij} = 1\}$ of (directed) *edges*. If y is symmetric and non-reflexive, i.e., $y_{ij} = y_{ji}$ and $y_{ii} = 0$ for all $1 \leq i, j \leq n$, the network can also be represented as an *undirected* graph in which the edges are unorderd pairs $\{i, j\} \in E \subseteq \binom{V}{2}$. For convenience we will concentrate on such undirected graphs for the most part.

Multiple forms of visualization have been devised for graphs. Below we

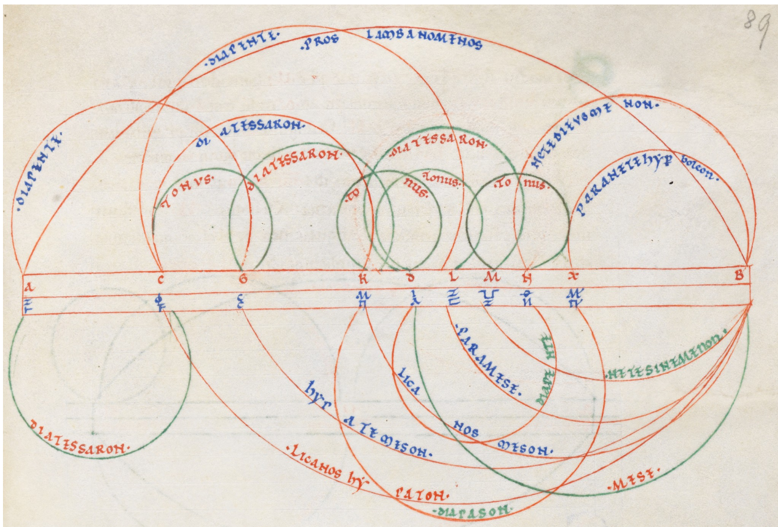


Figure 2: 12th century copy of a diagram showing musical relationships originally due to Boëthius (c. 480–524). Source: St. Gallen, Kantonsbibliothek, Vadianische Sammlung, VadSlg Ms. 296, f. 89r. Boethius, De arithmetica, De institutione musica `https://www.e-codices.ch/en/list/one/vad/0296`

introduce an intuitive design exemplified in Figure 1 that is also the most frequently used, and since its spatial arrangement has more degrees of freedom than a statistical chart, we briefly review the problem of layout in Subsection 2.2. In Subsection 2.3 we give special consideration to large networks. This section ends with a discussion of alternative representations and some ideas for the visualization of multivariate network data.

2.1 Standard representation

The most common representation of graphs are node-link diagrams in which vertices are represented by point-like features and edges by curves or line segments connecting them. Examples are shown in Figure 1 and alternatives are discussed in Subsection 2.4.

Node-link diagrams appear long before Euler's seminal work on the bridges of Königsberg, which is usually considered the beginning of graph theory but does not contain a drawing of a graph. For centuries, point-features and connecting curves had already been used on maps and for non-spatial relations, including ancient board games, astrological and logical diagrams, ancestral relations, and geometric drawings. Kruja et al. [26] give a short history of graph drawing, and one is likely to turn up further stunning examples such as the one shown in Figure 2 when digging deeper.

Although the standard representation is intuitive, it is prone to a number of potential shortcomings. Experiments show that small angles and crossings of edges may hinder reading [34, 42] spatial proximity suggests cohesive groups even when they are connected only loosely [28], and, quite obviously, identification of elements becomes difficult in the presence of overlap.

2.2 Layout

The research field of graph drawing [38] is concerned with geometric representations of and layout algorithms for graphs and hypergraphs whereas visualization design, task appropriateness, and user interaction are more commonly studied in the areas of information visualization, visual analytics, and human-computer interaction.

Spatial arrangement, or layout, is a non-trivial issue with any graphical representation. Unlike statistical charts such as scatterplots, time series, or pie charts, however, node-link diagrams usually do not come with given relative positions and thus exhibit more degrees of freedom. This is a curse and a blessing, because on the one hand, we can adjust the layout to express additional information and/or increase readability, and on the other hand, edges create complex dependencies turning both into rather daunting tasks.

Readability criteria such as density distribution, size of angles, number and angles of crossings, bends, area, symmetry, etc. are sometimes referred to as aesthetic criteria, and their priorities may be influenced by the task at hand. In graph drawing, these are turned into constraints or optimization objectives for layout algorithms.

While specialized algorithms have been proposed for classes of graphs such as trees and planar graphs, and for representation variants such as layered or orthogonal layouts, one group of layout algorithms clearly dominates the practical use of algorithms for general undirected graphs. They are referred to as force-directed algorithms [25, 5] because they are inspired by physical analogies of repelling nodes (for good distribution and little overlap) and edges acting as springs between them (for uniform edge length and visually recognizable group cohesion). As a byproduct, symmetries can often be recognized well, and the simulation of physical forces facilitates modes of interaction that feel natural to users.

On the negative side, the simple, intuitive, and widely available implementations are non-deterministic, sensitive to poor initialization, they often get stuck in local minima from which the iterative improvements strategy is unable to escape, and they have difficulties with graphs of low diameter or large size.

The most robust and reliable variants are instances of multidimensional scaling, with the shortest-path distance as input and coordinates as output. Instead of a force-calculation determining an update step, minimization of a layout objective function is attempted. For a two-dimensional layout $p = (p_i)_{i \in V} \in \mathbb{R}^{n \times 2}$ with $p_i = \langle x_i, y_i \rangle$ the squared relative error of shortest-path distances in the graph represented by Euclidean distances in the layout is defined as

$$stress(p) = \sum_{i,j \in V} \frac{1}{dist(i,j)^2} \left(\|p_i - p_j\| - dist(i,j) \right)^2 \ .$$

Exact minimization is computationally intractable but with good initialization and carefully designed iterative improvement procedures such as majorization [15, 40], low-stress layouts can be obtained reliably and efficiently [9]. Approximate minimization for large graphs is discussed in the next subsection.

Note that the stress objective can be modified by altering the notion of target distances, by varying the relative contribution of dyads i, j, or by building auxiliary graph structures that may include virtual vertices and edges. Other layout requirements may be expressed as constraints that restrict the space of admissible layouts for instance by fixing vertices to certain areas or relative to each other. The approach is thus more flexible and can be adapted to more different application settings than one might initially suspect. This includes dynamic graphs.

The importance of the stress-minimization approach is reinforced by the fact that other important approaches turn out to be special cases. Spectral layout, where coordinates are determined from eigenvectors of the Laplacian matrix of a graph, and barycentric layout, where some vertices are fixed and the others are placed in average position of their neighbors are examples.

A recent development are neighborhood embeddings, where distances are determined only locally and patched together, appear to be especially suited to highlight clustering structure [27, 35].

2.3 Large networks

With increasing size of a network, the problem of visualizing its graph changes until it becomes qualitatively different.

An obvious challenge for layout algorithms is running time. Without special precautions, a single iteration moving every vertex only once requires time linear in the size of the graph. Speed-up techniques attempt to reduce the number of iterations using fast methods that get the larger distances approximately right so that the iterative procedure does local adjustments only.

Multilevel-methods obtain suitable initializations by recursively operating on smaller graphs [13, 17, 22]. Simpler but no less effective is the use of approximate classical scaling [8] a spectral decomposition method that prioritizes larger distances, and requires near-linear running time. Additionally, the time spent in iterations can be reduced by coarsening the stress function and thus eliminating redundancies [32], or by parallelizing algorithms for GPU computation [40].

Beyond runtime, display limitations are another concern. Even with sufficient resolution to display tens of thousands of line segments for edges, it may not be possible for a human viewer to discern the details. Worse, the nature of the stress objective is such that low variance in distances leads to largely uniform vertex distribution and cluttered edges. This is sometimes referred to as the hairball-problem of small-world networks. Compensation techniques include pre- and postprocessing during layout generation and level-of-detail rendering of a given layout.

An example of a preprocessing technique particularly suited for graphs with low variance in distances is the determination of a backbone, i.e., a subgraph induced by edges that are contained in regions of relatively high local density [31]. Absent many shortcut edges, average distances in such backbone structures are generally larger which makes their layout easier.

Edge bundling is a technique that has been used in both, pre- and post-processing. In the most common variants [21, 14], a given layout is modified by bundling the middle segments of edges that would run close to parallel anyway because they start and end in similar layout regions.

Abstractions and simplifications can also be accomplished in graphical space, for instance by adjusting the level of detail at which a graph with a given layout is drawn [44]. A more comprehensive overview is given by von Landesberger et al. [39].

2.4 Other representations

The standard representation in the form of node-link diagrams is not the only way of visualizing networks. Straightforward variants include drawings with orthogonal edges (as are common for circuit schematics) or other restricted slopes (as in metro maps). Implicit representation of edges appears, for instance, in inclusion drawings of trees where vertices are represented as areas and these areas are placed within other areas such that parent-child relationships can be inferred from area inclusion.

Many other representations exist but are often feasible only for graphs that satisfy structural properties such as acyclicity or planarity.

A common alternative that applies to general graphs are matrix representations where rows and columns are indexed by the vertices and edges are represented in matrix cells. A one-dimensional layout problems remains: the (joint) ordering of rows and columns. This ordering is important as it relates to the recognizability of structural features in the form of cell-arrangement patterns such as on- or off-diagonal blocks (density within or between groups) and crosses (high-degree vertices brokering between groups).

Just like graph layout algorithms, many ordering criteria and algorithms have been considered [3]. Most of the objectives are computationally intractable [10] giving rise to interesting computational challenges.

A comparison between standard and matrix representations suggests that they have complementary strengths and weaknesses [16]. Consequently, there are approaches that transition between these representations at different resolution levels [1] or combine them in a single representation based on local density [20] or select paths [37]. Attempts to alleviate some of the weaknesses include modifications adding cues at the boundaries [19] and decomposing the rectangular area into patches [2, 11]. Matrix representations are also particularly suited for certain forms of interaction such as resizing or folding rows and columns [12].

Finally we mention that hypergraphs, where edges are subsets of vertices of any cardinality, can be represented as bipartite graphs in which both the original vertices and edges are represented as vertices, and each vertex-edge incidence is represented by an edge. While this representation allows to use common graph visualization techniques, more specific representations such as Venn diagrams exist as well.

2.5 Multivariate networks

A network as defined above is a single variable representing relationships between entities. In realistic data-analytic scenarios, it is unlikely to be the only variable. Often, there will be additional node-level attributes and multiple types of relations, possibly on changing sets of nodes. Even more dimensions are introduced if one or more of the variables vary over time, which results in a dynamic network. Networks made up of multiple relations are often referred to as multilayer networks [24] and their visualization is discussed in [23].

As an example consider the visual encoding of two node variables in coordinates. Their quantitative, ordinal, or categorical values constrain the spatial layout and in the extreme case node positions are fixed as in a scatterplot [43]. Note that choosing such a graphical mapping favors the nodal attribute data over network structure which was the sole criterion in the layout algorithms above.

Many other ways of visually encoding multivariate networks exist. Often, additional graphical elements are added such as labels, colors, and glyphs for enriching nodes, line thickness, shapes, and gradients for enriching links, or additional separating lines and boundaries to compartmentalize information. Figure 3 shows an example in which the fixed location of each dyad in a matrix representation is leveraged for aligning time series data.

The more a network graphic is visually enriched, the more relevant the importance of general visualization guidelines. When using color to group nodes into categories it is, for instance, important to understand that a viewer will only be able to distinguish reliably six to twelve of such categorical color bins. An overview of work that helps to properly ground designs in human perception is provided by Munzner [29].

3 Application Examples

From a data visualization perspective, the graphical representation of a network should be designed such that relevant information can be perceived with ease and accuracy. Since data, information, and tasks differ across application domains, so does appropriate visualization. Domain traditions and prior knowledge of recipients require further adaptation.

We next discuss network visualization in two very different scenarios to illustrate the breadth and depth of issues arising.

3.1 Social Networks

When social phenomena are described as networks of social relations, the information to be conveyed in their visualizations may be manifold, with different foci, different aspects, and on different levels [18, 4]. Thus, an especially rich set of tasks and visualization techniques has evolved around the concept of social networks [6].

On the macro level, the interest is generally in characteristics of the social network as a whole. Such characterics may include whether the network consists of a dense core and a loosely connected periphery or whether it is polycentric, whether there are many shortcuts that accelerate diffusion or whether there are bottlenecks, and whether subnetworks are organized hierarchically or whether the network is flat. Certain characteristics that are commonly encountered in social networks require adaptation of layout algorithms as described for small-world networks in Section 2.2.

Figure 3: Evolution of a social network of co-habitating students [30] with 15 waves of observation stacked from bottom to top in each cell [7]. Length and tilt of each line indicate how the two students involved ranked each other.

On the micro level, the interest is in individual differences and special configurations such as node centrality and the prevelance of substructures which are sometimes referred to as motifs. Network-analytic techniques focusing on characteristics of nodes or links typically yield additional attribute data and therefore lead to multivariate network visualization problems.

The example in Figure 3 depicts a series of social networks in a single matrix representation by combining all observations pertaining to the same dyad into one stacked glyph that captures extent and asymmetry of mutual preference.

3.2 Overlay Networks for Automotive Engineers

In the second example, we discuss a network anlaysis tool called *RelEx* [36], which was built to support automotive engineers in understanding in-car communication networks. The challenges of analyzing such networks are

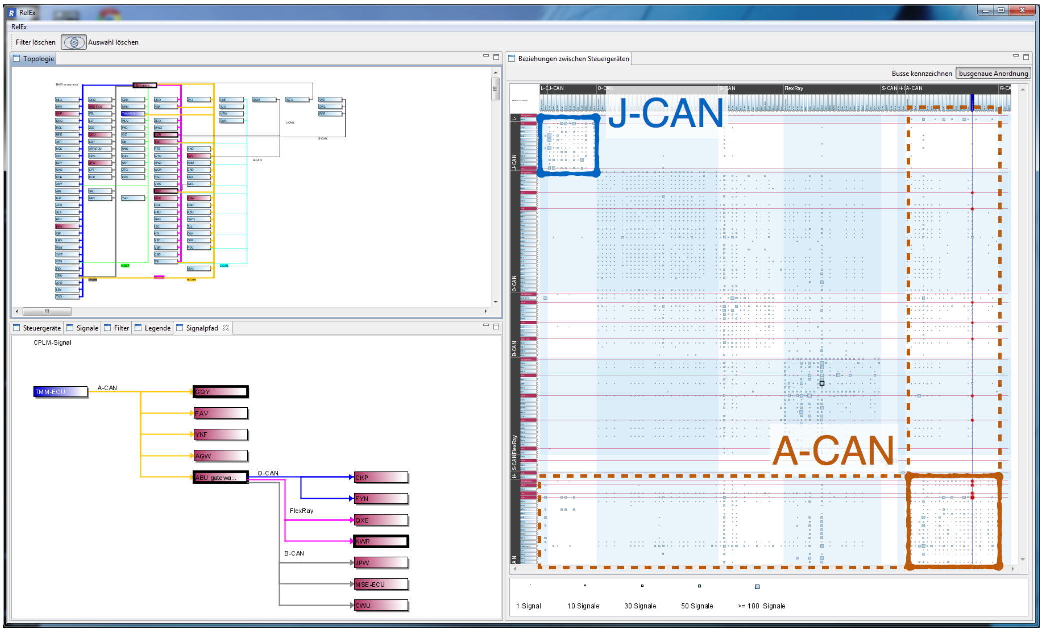


Figure 4: Screenshot of the RelEx (Relation Explorer) tool [36]. The tool comprises three main views: (top-left) the physical network diagram, (bottom-left) a filtered version of the physical network showing a signal path of a selected signal, (right) a matrix view showing an overview of the logical network.

very different as the ones from analyzing social networks. Instead of the node scalability issue in social networks, the main complexity stems from the interplay of different network types that form an overlay network. While there are only a few nodes (up to 100 control units), the network is very dense, which makes a matrix representation a viable design choice.

Abstractly, in-car communication networks can be viewed as an overlay network. The physical network builds the foundation and consist of up to 100 electronic control units (ECUs) as nodes, connected by 10-15 different bus systems as undirected hyperedges[1]. Overlaid on that is a logical network, that specifies how signals are exchanged between ECUs. Abstractly, the logical network forms a multigraph[2]: the nodes are again ECUs, and the signals are directed edges between them. The engineers' primary task is now to map the logical to the physical network. While there are algorithmic ways of mapping, the many ill-defined constraints and dynamic processes in an automobile company make an interactive visual analysis tool a viable choice.

[1] A hyperedge connects multiple nodes

[2] A multigraph can have multiple edges between nodes

Figure 4 shows a screenshot of the RelEx tool. As many visual anlaysis tools, RelEx follow a multiple coordinated views which are connected by linking and brushing[3]. On the top-left, the physical network is shown in an automotive-typical way. The bottom-left shows the specific path a selected signal takes over this network.

The view we want to focus our discussion the most on is the matrix view on the right side. The matrix view provides an overview over the logical network. The ECU nodes are the lines and columns. Each cell in turn is marked with a box in case one or more signals are exchanged between the pair of ECU nodes. The number of signals exchanged is encoded by the size of the box so that important connections that exchange many signals visually pop out (this visual "pop out" effect is further support by adding a black frame for the communication hotspots with more than 100 signals). We see that many signal boxes exist, that is, the logical network is very dense. In a node-link diagram, this characteristic would lead to extreme clutter making it almost impossible to perceive relevant information (the so-called "hairball effect"). Here, the matrix view offers a viable alternative.

In addition, it supports tasks that might be harder to conduct with classical node-link representations. For instance, the ECU nodes in the figure were ordered based on which bus system they are connected to. This is shown as the light blue background stripes in the matrix. Ordering the matrix in this way allows to better understand how much within-bus communication and how much between-bus communication is going on, by observing the intersection between these bus stripes. The within-bus communication is represented by the signal boxes that lie on the diagonal intersections of bus stripes. For instance, the intersection rectangle at the top left, shown as blue highlight in Figure 4, indicates the communication that is going on among ECUs connected to the J-CAN. The orange highlight at the bottom right shows the communication within the A-CAN. While for both CAN busses there is much within-bus communication on the diagonal, the J-CAN also comprises much communication to other bus systems. This can be seen by the signal boxes that are in the stripes but not on the diagonal, as indicated by the dotted lines in Figure 4. Such insights, can be of high importance for automative engineers when making decisions in how to change and optimize the network design.

[3]Selections in one view get also highlighted in all other views.

4 Challenges and Opportunities

As the requirements change with origin, structure, content, representation, and interest, network visualization tasks abound and new challenges arise continuously. New approaches to network analysis and applications of network science in other domains inspire novel forms of network visualization.

Visualization tools therefore often combine tested and generic methods for layout with flexible means for attribute mapping and interactive exploration. Still, as visualization can be seen as the human lense to data, further research is needed to assess which visualization designs are understood by targeted groups of recipients. Display technologies, 3D printing, and augmented reality provide further opportunities to explore networks.

An important challenges for users of network visualization systems is not to fall for images of complexity and decorative beauty but to concentrate on the essential purpose of network visualization, namely to facilitate exploration and hypothesis formation as well as communication and the provision of evidence for conclusions.

References

[1] James Abello and Frank van Ham. Matrix zoom: A visual interface to semi-external graphs. In Ward and Munzner [41], pages 183–190.

[2] Juhee Bae and Benjamin Watson. Developing and evaluating quilts for the depiction of large layered graphs. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2268–2275, 2011.

[3] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. Matrix reordering methods for table and network visualization. *Comput. Graph. Forum*, 35(3):693–716, 2016.

[4] Stephen P. Borgatti, Martin G. Everett, and Jeffrey C. Johnson. *Analyzing Social Networks.* Sage, 2nd edition, 2018.

[5] Ulrik Brandes. Force-directed graph drawing. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*, pages 768–773. Springer, 2016.

[6] Ulrik Brandes, Linton C. Freeman, and Dorothea Wagner. Social networks. In Tamassia [38], pages 805–839.

[7] Ulrik Brandes and Bobo Nick. Asymmetric relations in longitudinal social networks. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2283–2290, 2011.

[8] Ulrik Brandes and Christian Pich. Eigensolver methods for progressive multidimensional scaling of large data. In Michael Kaufmann and Dorothea Wagner, editors, *Graph Drawing, 14th International Symposium, GD 2006, Karlsruhe, Germany, September 18-20, 2006. Revised Papers*, volume 4372 of *Lecture Notes in Computer Science*, pages 42–53. Springer, 2006.

[9] Ulrik Brandes and Christian Pich. An experimental study on distance-based graph drawing. In Ioannis G. Tollis and Maurizio Patrignani, editors, *Graph Drawing, 16th International Symposium, GD 2008, Heraklion, Crete, Greece, September 21-24, 2008. Revised Papers*, volume 5417 of *Lecture Notes in Computer Science*, pages 218–229. Springer, 2008.

[10] Josep Díaz, Jordi Petit, and Maria J. Serna. A survey of graph layout problems. *ACM Comput. Surv.*, 34(3):313–356, 2002.

[11] Kasper Dinkla, Michel A. Westenberg, and Jarke J. van Wijk. Compressed adjacency matrices: Untangling gene regulatory networks. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2457–2466, 2012.

[12] Niklas Elmqvist, Yann Riche, Nathalie Henry Riche, and Jean-Daniel Fekete. Melange: Space folding for visual exploration. *IEEE Trans. Vis. Comput. Graph.*, 16(3):468–483, 2010.

[13] Pawel Gajer and Stephen G. Kobourov. GRIP: graph drawing with intelligent placement. *J. Graph Algorithms Appl.*, 6(3):203–224, 2002.

[14] Emden R. Gansner, Yifan Hu, Stephen C. North, and Carlos Eduardo Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In Giuseppe Di Battista, Jean-Daniel Fekete, and Huamin Qu, editors, *IEEE Pacific Visualization Symposium, PacificVis 2011, Hong Kong, China, 1-4 March, 2011*, pages 187–194. IEEE Computer Society, 2011.

[15] Emden R. Gansner, Yehuda Koren, and Stephen C. North. Graph drawing by stress majorization. In Pach [33], pages 239–250.

[16] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In Ward and Munzner [41], pages 17–24.

[17] Stefan Hachul and Michael Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In Pach [33], pages 285–295.

[18] Marina Hennig, Ulrik Brandes, Jürgen Pfeffer, and Ines Mergel. *Studying Social Networks – A Guide to Empirical Research*. Campus, 2012.

[19] Nathalie Henry and Jean-Daniel Fekete. Matlink: Enhanced matrix visualization for analyzing social networks. In Maria Cecília Calani Baranauskas, Philippe A. Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa, editors, *Human-Computer Interaction - INTERACT 2007, 11th IFIP TC 13 International Conference, Rio de Janeiro, Brazil, September 10-14, 2007, Proceedings, Part II*, volume 4663 of *Lecture Notes in Computer Science*, pages 288–302. Springer, 2007.

[20] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1302–1309, 2007.

[21] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. Vis. Comput. Graph.*, 12(5):741–748, 2006.

[22] Yifan Hu. Efficient, high-quality force-directed graph drawing. *The Mathematica Journal*, 10(1):37–71, 2006.

[23] Andreas Kerren, Helen C. Purchase, and Matthew O. Ward, editors. *Multivariate Network Visualization - Dagstuhl Seminar #13201, Dagstuhl Castle, Germany, May 12-17, 2013, Revised Discussions*, volume 8380 of *Lecture Notes in Computer Science*. Springer, 2014.

[24] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *J. Complex Networks*, 2(3):203–271, 2014.

[25] Stephen G. Kobourov. Force-directed drawing algorithms. In Tamassia [38], pages 383–408.

[26] Eriola Kruja, Joe Marks, Ann Blair, and Richard C. Waters. A short note on the history of graph drawing. In Petra Mutzel, Michael Jünger,

and Sebastian Leipert, editors, *Graph Drawing, 9th International Symposium, GD 2001 Vienna, Austria, September 23-26, 2001, Revised Papers*, volume 2265 of *Lecture Notes in Computer Science*, pages 272–286. Springer, 2001.

[27] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[28] Cathleen McGrath, Jim Blythe, and David Krackhardt. Seeing groups in graph layouts. *Connections*, 19(2):22–29, 1996.

[29] Tamara Munzner. *Visualization analysis and design*. AK Peters/CRC Press, 2014.

[30] Theodore M. Newcomb. *The Acquaintance Process*. Holt, Rinehart, and Winston, New York, NY, 1961.

[31] Arlind Nocaj, Mark Ortmann, and Ulrik Brandes. Untangling the hairballs of multi-centered, small-world online social media networks. *J. Graph Algorithms Appl.*, 19(2):595–618, 2015.

[32] Mark Ortmann, Mirza Klimenta, and Ulrik Brandes. A sparse stress model. *J. Graph Algorithms Appl.*, 21(5):791–821, 2017.

[33] János Pach, editor. *Graph Drawing, 12th International Symposium, GD 2004, New York, NY, USA, September 29 - October 2, 2004, Revised Selected Papers*, volume 3383 of *Lecture Notes in Computer Science*. Springer, 2004.

[34] Helen C. Purchase. Effective information visualisation: a study of graph drawing aesthetics and algorithms. *Interacting with Computers*, 13(2):147–162, 2000.

[35] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[36] Michael Sedlmair, Annika Frank, Tamara Munzner, and Andreas Butz. RelEx: Visualization for actively changing overlay network specifications. *IEEE transactions on visualization and computer graphics*, 18(12):2729–2738, 2012.

[37] Zeqian Shen and Kwan-Liu Ma. Path visualization for adjacency matrices. In Ken Museth, Torsten Möller, and Anders Ynnerman, editors,

EuroVis07: Joint Eurographics - IEEE VGTC Symposium on Visualization, Norrköping, Sweden, 23-25 May 2007, pages 83–90. Eurographics Association, 2007.

[38] Roberto Tamassia, editor. *Handbook on Graph Drawing and Visualization*. Chapman and Hall/CRC, 2013.

[39] Tatiana von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J. van Wijk, Jean-Daniel Fekete, and Dieter W. Fellner. Visual analysis of large graphs. In Helwig Hauser and Erik Reinhard, editors, *Eurographics 2010 - State of the Art Reports, Norrköping, Sweden, May 3-7, 2010*, pages 37–60. Eurographics Association, 2010.

[40] Yunhai Wang, Yanyan Wang, Yinqi Sun, Lifeng Zhu, Kecheng Lu, Chi-Wing Fu, Michael Sedlmair, Oliver Deussen, and Baoquan Chen. Revisiting stress majorization as a unified framework for interactive constrained graph visualization. *IEEE transactions on visualization and computer graphics*, 24(1):489–499, 2018.

[41] Matthew O. Ward and Tamara Munzner, editors. *10th IEEE Symposium on Information Visualization (InfoVis 2004), 10-12 October 2004, Austin, TX, USA*. IEEE Computer Society, 2004.

[42] Colin Ware, Helen C. Purchase, Linda Colpoys, and Matthew McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, 2002.

[43] Martin Wattenberg. Visual exploration of multivariate graphs. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 811–819. ACM, 2006.

[44] Michael Zinsmaier, Ulrik Brandes, Oliver Deussen, and Hendrik Strobelt. Interactive level-of-detail rendering of large graphs. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2486–2495, 2012.