1/2-Day Tutorial Proposal: Grooming the Hairball – How to Tidy up Network Visualizations?

Hans-Jörg Schulz*
University of Rostock, Germany

Christophe Hurter†
ENAC/University of Toulouse, France

ABSTRACT

Every visualization researcher and practitioner knows the painful experience of a beautifully designed network layout breaking down once the input graph scales up to realistic node and edge counts. The resulting "hairball" suffers from cluttering and overplotting to an extreme that renders it unusable for any practical purposes. Since researchers have had this experience for decades, various approaches have been developed on all stages of the visualization pipeline to alleviate this problem. They range from filtering and clustering techniques on the data level to modern GPU-based techniques on the image level. This tutorial gives an overview of these techniques and discusses their applicability and interplay in different application scenarios. By doing so, it provides a unique problem-oriented perspective on the field of scalable network visualization, which is an area of active research today more than ever. The tutorial serves mainly to further the understanding of network visualization beyond the point of creating an initial layout. It thus caters to an intermediate level audience with some basic knowledge on graph layout and visualization, but it will certainly present an interesting cross-section through the larger domains of network visualization and graph drawing for established researchers as well.

Index Terms: I.3.3 [Computing Methodologies]: Computer Graphics—Picture/Image Generation; I.3.6 [Computing Methodologies]: Computer Graphics—Methodology and Techniques

1 TUTORIAL CONTENTS

Since network-structured data has become a mainstream concept that every user of online social networks is familiar with, the interest in the visualization of such data has grown – and so has the realization that this is a challenging and computationally complex task in particular for larger networks. The proposed **1/2 day tutorial for intermediate audiences** discusses the different existing approaches that address this challenge of producing or refining network visualizations to scale to realistically sized data sets while maintaining readability. It covers the topic from the two perspectives of node-set-based methods (Part I) and edge-set-based methods (Part II), which are each further subdivided into methods working on the three levels of the visualization pipeline: the data level, the geometry level, and the image level. A third part then brings the former two parts together by discussing their interplay and application in various domain-specific scenarios (Part III). The proposed three parts of this tutorial are outlined in the following.

1.1 PART I: Methods Working on the Node Set

Methods that aim to reduce or refine the node set are probably the most commonly used approaches to get a grip on large networks. The reason is that they automatically serve the reduction of the edge set as well, as all reduction on the nodes is reflected on their incident edges. In this part, a selected number of such approaches is covered that are deemed characteristic for the particular level and widely used in practice.

Data Level Methods. On data level, two main directions can be taken that aim to reduce the input graph before it is laid out and thus generates the problems mentioned above. These directions are abstraction (e.g., clustering or partitioning) and selection (e.g., filtering or contraction). Abstraction basically coarsens the level of detail, but keeps the full extent of the network, whereas selection cuts down on the extent, but keeps the remainder of the network at fullest detail. This part highlights some fundamental graph clustering approaches and graph partitioning approaches, and then it shows how partitioning can be used in conjunction with clustering, as it is done in [1, 37]. The latter is a means to alleviate some of the problematic aspects of graph clustering, such as its long runtime and the poor interpretability of its results. The selection part briefly covers how it is actually performed and then focuses on the underlying question of how to distinguish between salient nodes and those of lesser importance to be filtered out. To this end, different importance metrics [15, 22] and different Degree of Interest methods [38, 27] are introduced and discussed.

Geometry Level Methods. On geometry level, one has to distinguish between methods that merely reflect a prior reduction on data level (e.g., layouts for clustered graphs) and those that perform a genuine reduction on the geometry level of a previously unreduced graph. As examples for layout methods reflecting a reduction, techniques are presented that tie in with a hierarchical clustering by using force-directed layouts [6, 28], as well as with navigation techniques that support switching between different clustering layers [8]. Methods that perform the desired reduction of the number of nodes on a purely geometric basis are rare. But approaches, such as GraphDice [2], which position nodes with the same numerical attributes on the same position of a scatterplot-like layout, are a good example of how this can nevertheless be achieved.

Image Level Methods. On image level, the methods described in the literature aim to render nodes, which are placed close together as one larger blob or splat. This does not only reduce the clutter, but also indicates where many nodes have been aggregated in this way, thus giving visual cues for zooming in to investigate these dense regions in more detail. Existing techniques range from those that operate globally on the entire given node-link diagram, such as Graph Splatting [41], to those that operate more locally only in those regions where it is necessary. Example for the latter are the density-based node aggregation [43] and the adaptive coloring by content-aware scaling [34]. These examples will be covered in this last part of the node-related reduction techniques, before switching the perspective to the edge-related methods in the following part.

1.2 PART II: Methods Working on the Edge Set

In this part of the tutorial, we will discuss algorithms and interactive techniques to reduce edge clutter in visualizations. These techniques mainly rely on edge-bundling algorithms which have been subject to an increased interest in active research and a number of improvements and enhancement in recent years. In the same way as the first part of the tutorial, this part is divided into the three

*e-mail: hjschulz@informatik.uni-rostock.de
†e-mail: christophe.hurter@enac.fr

levels on which edge simplification can occur: the data level, the geometry level, and the image level.

Data Level Methods. On data level, edges can be filtered or aggregated. Filtering techniques remove edges with given criteria, whereas aggregation techniques merge edges having similar semantics.

Aggregation and edge clustering methods are given in [7]. Interactive systems, such as Node Trix, compact dense subgraphs into matrix representations [14, 13]. In Ploceus [26], one can display networks from different perspectives, at different levels of abstraction, and with different edge semantics.

Regarding filtering techniques, direct queries can filter out edges with multivariate criteria. Edges can be filtered with Centrality Based Visualization of Small World Graphs [40], or explored with spanning tree as used in TreePlus [25].

Geometry Level Methods. On geometry level, dense edge visualizations can be uncluttered by using edge bundling techniques. Edge bundling techniques trade clutter for overdraw by routing geometrically and semantically related edges along similar paths. This improves readability in terms of finding groups of nodes related to each other by tracing groups of edges (the bundles) which are separated by whitespace [11]. Dickerson et al. merge edges by reducing non-planar graphs to planar ones [4]. The first edge bundling technique was the flow map visualizations which produce a binary clustering of nodes in a directed graph representing flows to route curved edges along [29]. Flow maps' control meshes are used by several authors to route curved edges, e.g., [30, 42].

These techniques were later generalized into edge bundling approaches that use a graph structure to route curved edges. Holten pioneered edge bundling for compound graphs by routing edges along the hierarchy layout using B-splines [16]. Gansner and Koren bundle edges in a circular node layout similar to [16] by area optimization metrics [12]. Control meshes can also be used for edge clustering in graphs, e.g., [30, 42]; a Delaunay-based extension called geometric-based edge bundling (GBEB) [3]; and "winding roads" (WR) that use Voronoi diagrams for 2D and 3D layouts [24, 23].

The most popular technique is the force-directed edge layout technique which use curved edges to minimize crossings, and implicitly creates bundle-like shapes [5]. Force-directed edge bundling (FDEB) creates bundles by attracting control points on edges close to each other [17], and was adapted to separate bundles running in opposite directions [35]. The MINGLE method uses multilevel clustering to significantly accelerate the bundling process [11].

Computation times for larger graph struggle with the algorithmic complexity of the edge bundling problem. This makes scalability the major issue when using edge bundling techniques. The latest techniques use therefore the image level to bundle edges.

Image Level Methods. Thanks to the recent improvements regarding graphic hardware and its flexible usage, image level methods are nowadays very popular. Graphic cards can be used to improve rendering aesthetics and to address scalability issues.

Several techniques exist for rendering and exploring bundled layouts, e.g., edge color interpolation for edge directions [16, 3]; transparency or hue for local edge density, i.e., the importance of a bundle, or for edge lengths [24]. Bundles can be drawn as compact shapes whose structure is emphasized by shaded cushions [36, 31]. Graph splatting visualizes node-link diagrams as continuous scalar fields using color and/or height maps [41, 21].

Several techniques exists to improve scalability based on image level. Skeleton-based edge bundling (SBEB) use the skeletons or medial axes of the graph drawing's thresholded distance transform as bundling cues to produce strongly ramified bundles [9]. To explore crowded areas where several bundles overlap, bundled layouts can be interactively deformed using semantic lenses [18]. Hurter et al. use a pixel based bundling method to explore dynamic graphs [20].

1.3 PART III: Application Scenarios and Summary

In this final part, we will bring the methods of the first two parts together by discussing which of them to apply and how to go about it for a few selected application cases. In particular, this part will highlight how to combine edge and node simplification under the different constraints imposed by the application domains. For this part, we rely mostly on our own experience in applying these techniques to trajectory exploration, software revision analysis, and stream graph exploration [21, 18, 20], as well as to social networking [39].

At the end of this part, we will briefly summarize the key points of our tutorial and also emphasize on the open challenges in this area, so that starting graduate students will have some pointers in which direction to look to make a contribution. Among other points, these challenges include scalability issues w.r.t. screen space, computation time, but also interaction, as well as compatibility issues among the individual approaches, which currently limit their concerted use.

2 Tutorial Organization and Course Material

The half-day tutorial will be held as a presentation using pictures, videos, and live demos from literature as well as the presenters' own work. We are going to spend about 75 minutes for each of Part I and Part II, and the remaining 1 hour for Part III and answering further questions. The participants will be provided with tutorial notes including an extensive literature list on the subject.

After completing this tutorial, participants can expect to have gained a comprehensive overview of existing approaches for making visualizations more expressive and more effective for communicating larger networks. Through the selected application scenarios, the participants will also get a grasp on when and how to utilize these techniques in practice. In addition, concrete suggestions on issues of realizing and implementing the discussed algorithms from the experience of the presenters will be given.

3 Instructor Information

Hans-Jörg Schulz received his PhD from the University of Rostock in 2010, where he is now a post-doctoral researcher. With his expertise in graph visualization, his perspective on showing relationships in data stems from a graph drawing point of view. His main interest lies in visualizing hierarchical relationships in data [32, 33]. This naturally includes compact tree visualizations for reducing clutter and overplotting [34] and hierarchically clustered graphs [39], but also extends to the recursive refinement of dynamic network visualization [13]. More on his work can be found at `http://hjschulz.net`

Christophe Hurter received his PhD from the University of Toulouse in 2010. He is associate professor at the Interactive computing laboratory at the French national aviation university (ENAC, Ecole Nationale de l'Aviation Civile) in Toulouse France. His research interests lie in the research areas of information visualization and Human-Computer Interaction, particularly including the visualization of multivariate data in space and time, the design of scalable visual interfaces and the development of pixel based rendering techniques. He published several papers on new scalable edge bundling techniques [19, 10], and real-time interaction techniques with edge bundling [18, 20]. More on his work can be found at `http://www.recherche.enac.fr/~hurter/`

Acknowledgements

Hans-Jörg Schulz's research is funded through an individual grant by the German Research Foundation (DFG).

References

[1] J. Abello, F. van Ham, and N. Krishnan. ASK-GraphView: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, 2006.

[2] A. Bezerianos, F. Chevalier, P. Dragicevic, NiklasElmqvist, and J.-D. Fekete. Graphdice: A system for exploring multivariate social networks. *Computer Graphics Forum*, 29(3):863–872, 2010.

[3] W. Cui, H. Zhou, H. Qu, P. Wong, and X. Li. Geometry-based edge clustering for graph visualization. *IEEE TVCG*, 14(6):1277–1284, 2008.

[4] M. Dickerson, D. Eppstein, M. Goodrich, and J. Meng. Confluent drawings: Visualizing non-planar diagrams in a planar way. *Journal of Graph Algorithms and Applications*, 9(1):31–52, 2005.

[5] T. Dwyer, K. Marriott, and M. Wybrow. Integrating edge routing into force-directed layout. In *Proc. Graph Drawing*, pages 8–19, 2007.

[6] P. Eades and M. L. Huang. Navigating clustered graphs using force-directed methods. *Journal of Graph Algorithms and Applications*, 4(3):157–181, 2000.

[7] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE TVCG*, 13(6):1216–1223, 2007.

[8] N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2010.

[9] O. Ersoy, C. Hurter, F. Paulovich, G. Cantareira, and A. Telea. Skeleton-based edge bundles for graph visualization. *IEEE TVCG*, 17(2):2364 – 2373, 2011.

[10] O. Ersoy, C. Hurter, F. Paulovich, G. Cantareiro, and A. Telea. Skeleton-based edge bundling for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2364–2373, 2011.

[11] E. Gansner, Y. Hu, S. North, and C. Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *Proc. PacificVis*, pages 187–194, 2011.

[12] E. Gansner and Y. Koren. Improved circular layouts. In *Proc. Graph Drawing*, pages 386–398, 2006.

[13] S. Hadlak, H.-J. Schulz, and H. Schumann. In situ exploration of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2334–2343, 2011.

[14] N. Henry, J.-D. Fekete, and M. J. McGuffin. NodeTrix: A hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.

[15] I. Herman, S. Marshall, G. Melançon, D. J. Duke, M. Delest, and J.-P. Domenger. Skeletal images as visual cues in graph visualization. In *Proceedings of VisSym'99*, pages 13–22, 1999.

[16] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE TVCG*, 12(5):741–748, 2006.

[17] D. Holten and J. J. van Wijk. Force-directed edge bundling for graph visualization. *Comp. Graph. Forum*, 28(3):670–677, 2009.

[18] C. Hurter, O. Ersoy, and A. Telea. MoleView: An attribute and structure-based semantic lens for large element-based plots. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2600–2609, 2011.

[19] C. Hurter, O. Ersoy, and A. Telea. Graph bundling by kernel density estimation. *Computer Graphics Forum*, 31(3pt1):865–874, 2012.

[20] C. Hurter, O. Ersoy, and A. Telea. Smooth bundling of large streaming and sequence graphs. In *Proceedings of the PacificVis'13*, 2013.

[21] C. Hurter, B. Tissoires, and S. Conversy. FromDaDy: Spreading data across views to support iterative exploration of aircraft trajectories. *IEEE TVCG*, 15(6):1017–1024, 2009.

[22] H. Karloff and K. E. Shirley. Maximum entropy summary trees. *Computer Graphics Forum*, 32, 2013. to appear.

[23] A. Lambert, R. Bourqui, and D. Auber. 3D edge bundling for geographical data visualization. In *Proc. Information Visualisation*, pages 329–335, 2010.

[24] A. Lambert, R. Bourqui, and D. Auber. Winding roads: Routing edges into bundles. *Comp. Graph. Forum*, 29(3):432–439, 2010.

[25] B. Lee, C. S. Parr, C. Plaisant, B. B. Bederson, V. D. Veksler, W. D. Gray, and C. Kotfila. TreePlus: Interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414–1426, 2006.

[26] Z. Liu, S. B. Navathe, and J. Stasko. Network-based visual analysis of tabular data. In *Proceedings of the VAST 2011*, pages 41–50, 2011.

[27] T. May, M. Steiger, J. Davey, and J. Kohlhammer. Using signposts for navigation in large graphs. *Computer Graphics Forum*, 31(3pt2):985–994, 2012.

[28] C. Muelder and K.-L. Ma. Rapid graph layout using space filling curves. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1301–1308, 2008.

[29] D. Phan, L. Xiao, R. Yeh, P. Hanrahan, and T. Winograd. Flow map layout. In *Proc. InfoVis*, pages 219–224, 2005.

[30] H. Qu, H. Zhou, and Y. Wu. Controllable and progressive edge clustering for large networks. In *Proc. Graph Drawing*, pages 399–404, 2006.

[31] R. Scheepens, N. Willems, H. van de Wetering, G. Andrienko, N. Andrienko, and J. J. van Wijk. Composite density maps for multivariate trajectories. *IEEE TVCG*, 17(12):2518–2527, 2011.

[32] H.-J. Schulz. Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications*, 31(6):11–15, 2011.

[33] H.-J. Schulz, Z. Akbar, and F. Maurer. A generative layout approach for rooted tree drawings. In *Proceedings of the PacificVis'13*, 2013.

[34] H.-J. Schulz, S. Hadlak, and H. Schumann. Point-based visualization for large hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):598–611, 2011.

[35] D. Selassie, B. Heller, and J. Heer. Divided edge bundling for directional network data. *IEEE TVCG*, 19(12):754–763, 2011.

[36] A. Telea and O. Ersoy. Image-based edge bundles: Simplified visualization of large graphs. *Comp. Graph. Forum*, 29(3):543–551, 2010.

[37] C. Tominski, J. Abello, and H. Schumann. CGV – an interactive graph visualization system. *Computers and Graphics*, 33(6):660–678, 2009.

[38] F. van Ham and A. Perer. 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.

[39] F. van Ham, H.-J. Schulz, and J. M. Dimicco. HoneyComb: Visual analysis of large scale social networks. In *Proceedings of the INTERACT'09*, pages 429–442, 2009.

[40] F. van Ham and M. Wattenberg. Centrality based visualization of small world graphs. *Computers and Graphics*, 27(3):975–982, 2008.

[41] R. van Liere and W. de Leeuw. GraphSplatting: Visualizing graphs as continuous fields. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):206–212, 2003.

[42] H. Zhou, X. Yuan, W. Cui, H. Qu, and B. Chen. Energy-based hierarchical edge clustering of graphs. In *Proc. PacificVis*, pages 55–62, 2008.

[43] M. Zinsmaier, U. Brandes, O. Deussen, and H. Strobelt. Interactive level-of-detail rendering of large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2486–2495, 2012.