EuroVis Workshop on Visual Analytics (2012)
K. Matkovic and G. Santucci (Editors)

Graph Visualization Using Hierarchical Edge Routing and Bundling

W. Kienreich[1], R. Wozelka[1], V. Sabol[1] and C. Seifert[1,2]

[1]Know-Center Graz, Austria
[2]Knowledge Management Institute, TU Graz, Austria

Abstract

Driven by rapidly growing application areas such as semantic knowledge bases and social networks, visualization of large graphs has been gaining importance recently. A large amount of nodes and intersecting edges presents a major challenge for usability and aesthetics, and may also pose a scalability problem. Edge routing and bundling methods proved useful for reducing clutter, while hierarchical techniques, besides providing the basis for level of detail rendering, also address scalability. We present work in progress which combines hierarchical techniques with edge routing and bundling methods, and utilizes their respective advantages. The proposed graph visualization method employs hierarchical aggregation of graph nodes and edges, and applies edge routing and bundling along the hierarchy to reduce clutter and improve the clarity of the representation.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Visualization techniques offer means for exploratory navigation and analysis of relationships present in graph data. Visual approaches are particularly useful when the user, possibly without having clearly defined objectives, seeks to gain insight into an unfamiliar graph data set. As a consequence, tools and methods for visual analysis of large graphs are becoming increasingly common in various application domains, prominent examples being semantic knowledge bases (see [ASJC11]) and social networks (see [SMER06]). Different authors, such as Bennet et al. [BRSG07] and Beck et al. [BBD09], support the position that consideration of aesthetic properties of graph layouts promises to improve the readability of the visual representation. Large number of nodes and edges are a source of usability and aesthetic problems which are caused by clutter arising from intersecting and overlapping links. Another issue related to large graphs is the scalability of the visualization, which is limited by the number of visual items that can be simultaneously displayed - a problem affecting mobile devices with small screens and Web-based clients with limited computing power.

In this paper we present work in progress which primarily addresses the problem of clutter, but also provides means for dealing with scalability issues. We propose a novel graph visualization method which utilizes hierarchical aggregation of graph data. Nodes and edges are aggregated to meta-nodes and meta-edges, providing a level of detail capable rendering and navigation. For reduction of clutter we introduce two hierarchical approaches. The first method extends the “flat” Voronoi-based edge routing technique introduced by Lambert et al. [LBA10] with a strategy for routing edges along a hierarchy of nested Voronoi areas. The second method builds upon the idea of force-directed edge bundling by Holten et al. [Hv09] and applies it on the hierarchically organized graph data set. Respective advantages of the hierarchical routing and bundling methods are compared and contrasted with flat, non-hierarchical representations.

2. Related Work

Graph visualization is a broad area of research where a large number of different approaches, each addressing particular requirements, have been developed (cf. [HMM00]). Visualizations of graphs consisting of a larger number of nodes and edges often employ clutter reduction techniques to address issues with *usability and visual appearance*. Edge bundling and edge routing methods have been successfully applied as clutter reduction techniques. Cui et al. [CZQ*08] proposes a geometry-based method for clustering edges into bundles, which employs control meshes to guide the bundling process. Holten [Hol06] introduces an edge bundling method utilizing hierarchical graph organization, and later describes a "self-organised" force-directed edge bundling method which does not require a control mesh or a hierarchy et al. [Hv09]. Kienreich et al. [KS10] presents a force-directed edge bundling algorithm, which accounts for semantic properties of edges. An edge routing algorithm recently introduced by Lambert et al. [LBA10] utilizes quad trees and Voronoi diagrams to avoid node-edge overlaps.

Hierarchical aggregation of graph nodes and edges using graph clustering methods (see [Sch07, AW10]) to address both scalability as well as usability and aesthetic aspects is not a new idea (cf. [Fen97]). Eades et al. [EH00] and Abello et al. [AvHK06] create a hierarchy of supernodes using graph clustering. The subsequent layout is calculated by traversing the hierarchy and positioning childnodes with a force-directed placement (FDP) algorithm et al. [FR91], which scales poorly but is known to produce visually pleasing layouts. As FDP is each time applied on only a small number of child-nodes, very large graphs can be visualized in an effective manner. The graph is navigated by expanding only branches which are of interest to the user, which aids clarity and reduces rendering costs. Bourquoi et al. [BAM07] take this idea a step further by assigning dedicated polygonal areas to clusters using Voronoi area subdivision, with the goal of reducing overlap.

Existing approaches employ either hierarchical aggregation or edge routing and bundling, while our method combines the advantages of both.

3. Hierarchical Approach

The steps of the overall hierarchical edge routing and bundling approach are depicted in figure 1. Our method requires that graph nodes are structured in a hierarchy (tree), which aggregates nodes into meta-nodes (also called clusters). The hierarchy serves three purposes: (i) fast recursive layout using FDP and Voronoi area subdivision, (ii) aggregating edges over the hierarchy, and (iii) level of detail rendering. Usually, the hierarchy is not given a-priori and needs to be generated. In principle, there are two different methods for constructing the hierarchy: graph clustering and, for semantic data, hierarchy extraction along semantic relations.

Graph nodes are laid out by traversing the hierarchy from top to bottom and positioning children on each hierarchy level. In the same step a Voronoi area subdivision is computed using layout positions as control points. Then, edges are aggregated into meta-edges by traversing the hierarchy bottom-up and summarizing edges from children in their parent nodes. For edge routing, a grid is calculated based on the hierarchically nested Voronoi areas computed during the node layout. The edges are then routed along the shortest paths in the grid. For edge bundling, the node positions and the edge information are sufficient (i.e. the grid is not used). The steps of our method are described into detail in sections 3.1 to 3.5.

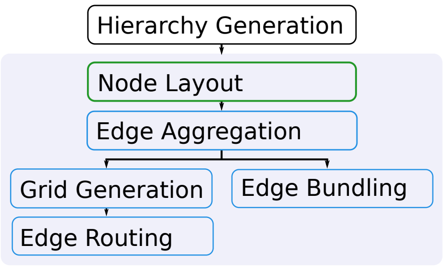

Figure 1: *Process overview*

3.1. Node Layout

Node and meta-node positions are computed by a recursive algorithm [MSG10] which traverses the hierarchy in a top-down manner: First, the top-level meta-nodes are positioned using the LinLog layout algorithm [Noa07] producing a layout where strongly interconnected meta-nodes are placed close to each other. Then, each meta-node is assigned a polygonal area using Voronoi area subdivision [Aur91]. Meta-nodes are repositioned to the centre of mass of the Voronoi region, so that very close points are moved away from each other and more space for routed edges is created. The method proceeds recursively by placing children within their parent's area and assigning each child its own Voronoi polygon. Recursion stops when the bottom of the hierarchy is reached and the original nodes are laid out and assigned Voronoi regions.

3.2. Edge Aggregation

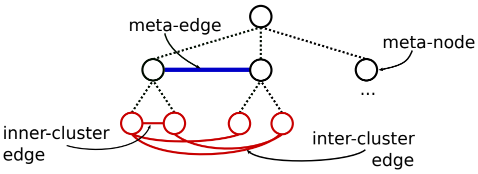

Figure 2: *Edge aggregation: Inter-cluster edges (red) are aggregated to meta-edges (blue) on the parent level.*

For the hierarchical edge bundling and routing, information on inter-cluster edges (i.e. edges connecting nodes which are not direct siblings in the hierarchy), is propagated from the leaf-nodes to the parent meta-node. The meta-edge propagates upwards until it connects two sibling meta-nodes (i.e. until it becomes an inner-cluster edge). Figure 2 shows this principle of edge aggregation. The red (leaf) nodes and the red edges represent the original graph. Black nodes and edges represent the imposed hierarchy. Inter-cluster edges are aggregated and represented by a meta-edge (blue) on the parent level of the hierarchy. The weight of the aggregated meta-edge is either set to the number of aggregated edges or to the sum of their weights (in case of a weighted graph).

3.3. Grid Generation

Basically, the Voronoi polygons constructed in the node layout step (see section 3.1) are used for routing the edges. Additionally to the nodes of the Voronoi polygons, we add a point for each Voronoi edge which divides the edge in half. These points connect to the node (or meta-node) within the corresponding Voronoi area, forming the node's "local grid". The local grid is only considered for routing edges between a node and the boundary of its Voronoi polygon (which is part of the Voronoi grid of its parent). Figure 3 (left) shows a simple example grid, where the nodes are positioned in a regular pattern and thus the Voronoi subdivision consists of rectangular areas. The edges inside the area of node 1 and node 2 are only visible for routing to node 1 and node 2 respectively (local grid).

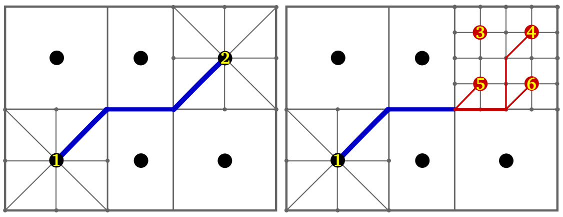

Figure 3: *Edge routing. Left: aggregated edge from cluster 1 to cluster 2. Right: expanded cluster 2, meta-edge is drawn from cluster 1 to the nearest routing point of cluster 2, inside cluster 2 original edges are drawn.*

3.4. Edge Routing

The edge routing step uses the meta-edges and the grid constructed in the previous steps and proceeds along the hierarchy in a top-down manner. The edges or meta-edges on each hierarchy level are routed along the grid edges using Dijkstra's shortest path algorithm, similar to Lambert et al. [LBA10]. A simple example, with start and end nodes of the edge being on the same hierarchy level, is shown in figure 3 (left). The routing is done in three steps. First, the edge is routed from the start node to the parent's Voronoi boundary using the local grid. Second, the edge is routed towards the Voronoi boundary of the end node along the Voronoi boundaries of the sibling nodes (i.e. using the grid at the hierarchy level). Third, starting from the Voronoi boundary of the end node the edge is routed along the local grid. An example demonstrating the resulting edge route over multiple hierarchy levels is shown in figure 3 (right) where cluster 2 is expanded. The meta-edge from cluster 1 (blue) is routed only up to the Voronoi boundary of cluster 2, and then it fans out to original edges (red), which are routed on the local grid of cluster 2. Note, that the edge to node 4 is again routed along the Voronoi boundary of node 5 and 6 and only at the boundary of its own region it is allowed to enter the local grid.

3.5. Edge Bundling

Edges are bundled using force-directed edge bundling introduced by Holten et al. [Hv09]. However, the large number of inter-cluster edges will not be bundled and rendered individually, because they have been hierarchically aggregated to a smaller number of inner-cluster meta-edges. In our approach, on each hierarchy level only these inner-cluster (meta-)edges (i.e. those between nodes having the same parent) are bundled together and rendered reducing the amount of clutter on screen. In other words, edge bundling is always performed locally within the parent cluster.

4. Discussion

Figure 4 shows our edge routing and edge bundling approach on a subset of approximately 3000 documents of the Reuters RCV document collection [LYRL04]. The hierarchy was created using the hierarchical clustering algorithm described by Muhr et al. [MSG10] yielding a cluster tree of depth 4. We generated graph data by treating documents as nodes and document similarities as weights of graph edges. Edges with lowest weights were pruned so that a) every node is connected by a least 1 edge, b) no node has more than 10 edges. The resulting graph is composed of approximately 3000 nodes and 20000 edges.

In figure 4 on left, the non-hierarchical edge routing (top) and edge bundling (bottom) are shown. Obviously, the large number of displayed nodes and edges causes clutter, whereby the problem is far more pronounced with edge bundling. In the centre of figure 4, top-level meta-nodes (cloud-icons) and aggregated meta-edges (blue) connecting them can be seen. In our opinion, edge bundling appears more suitable for providing an overview, because it makes it easier to follow the propagation of edges compared to edge routing, which conveys meta-edge weight through colour intensity but exhibits high edge overlap.

The user can navigate the hierarchy from top to bottom by expanding the cluster hierarchy and unveiling more detailed structures of the graph, which is demonstrated by screenshots on the right-hand side of figure 4. Note that, in contrast to non-hierarchical layouts, only the part of the graph which is of interest to the user is shown in detail, thus increasing the overall clarity. Another advantage of the hierarchical approach is that far less edges and nodes are rendered, making it more suitable for devices with small screens and limited computing power.

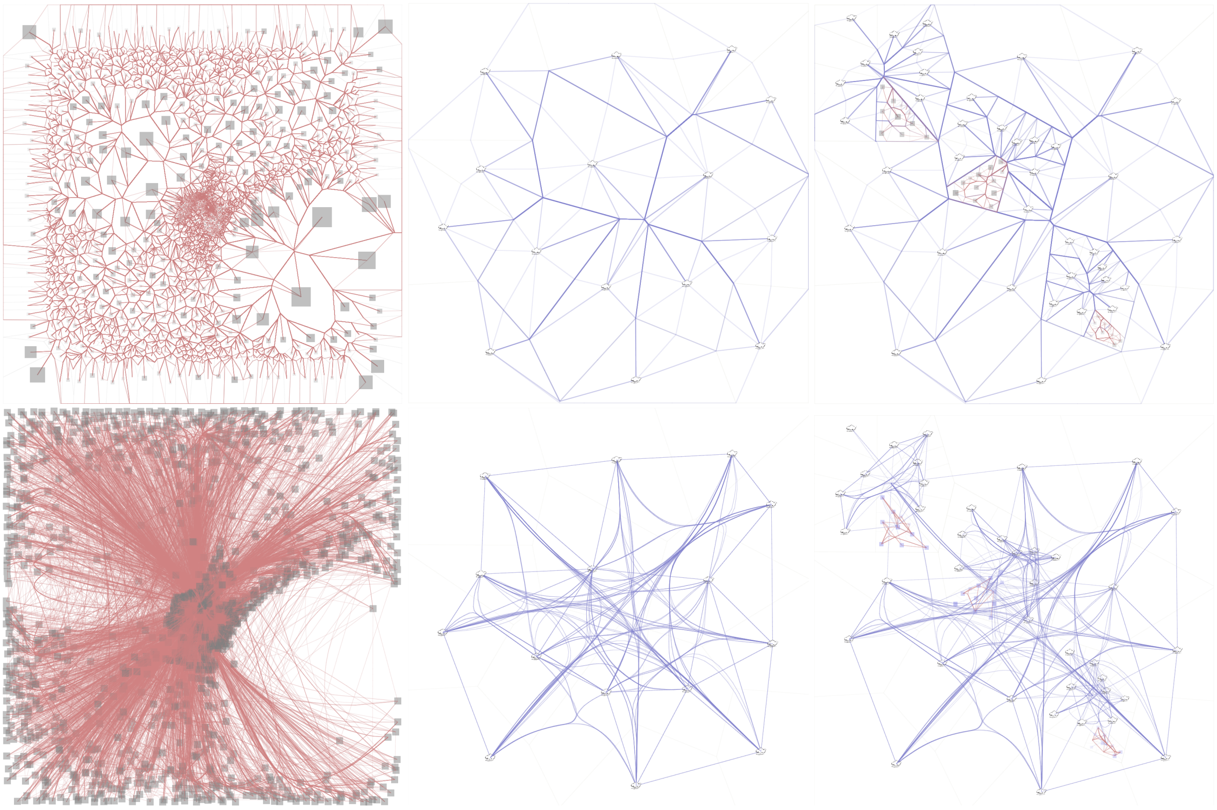

Figure 4: *Edge routing (top row) and bundling (bottom row) examples for a graph consisting of approximately 3000 nodes and 20000 edges. Red edges represent original edges of the graph, blue edges are meta-edges. First column: no hierarchy. Second column: routing and bundling using the hierarchy. Third column: greater level of detail for selected hierarchy branches.*

Comparing edge routing with edge bundling with more details shown, we conclude that routing is superior in terms of clarity. Comparing regions with original nodes (rectangles) and edges (in red) one sees that edge bundling introduces significantly more clutter than edge routing.

5. Conclusion and Future Work

We presented a graph visualization method which extends existing methods for edge bundling and routing to hierarchically aggregated graphs, resulting in reduced clutter, improved clarity of the representation and potentially better scalability on small (mobile) devices.

In the future we will focus on supporting interactive analytical tasks, such as in Wong et al. [WMC*09], and perform user tests to evaluate the effectiveness of our approach and compare it to other methods. To address the issue of ambiguity of aggregated meta-edges we plan to introduce and evaluate user interaction features, such as mouse-over highlighting of individual edges within the hierarchy of meta-edges. Further, we will introduce spline-based edge rendering, which will solve the problem of overlapping edges belonging to adjacent clusters routed along a common Voronoi boundary. As another approach to tackle edge overlap we plan to reserve dedicated space along cluster Voronoi boundaries to prevent cluster-local edges from obscuring meta-edges passing through.

In principle, our current design (implemented in Java) supports visualizations of large graphs on thin clients. Thus, we will implement a client-server solution, where the layouting is performed on the server, while the HTML5-based thin client is only responsible for dynamically loading and rendering parts of the hierarchically structured graph layout. Furthermore, we are currently evaluating the scalability of graph clustering methods for the hierarchy generation step, in order to scale the layouting algorithm to very large data sets.

Acknowledgement: The Know-Center is funded within the Austrian COMET Program under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

[ASJC11] AL-SAFFAR S., JOSLYN C., CHAPPELL A.: Structure discovery in large semantic graphs using extant ontological scaling and descriptive semantics. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on* (aug. 2011), vol. 1, pp. 211 –218. 1

[Aur91] AURENHAMMER F.: Voronoi diagrams – survey of a fundamental geometric data structure. *ACM Comput. Surv. 23* (September 1991), 345–405. 2

[AvHK06] ABELLO J., VAN HAM F., KRISHNAN N.: Askgraphview: A large scale graph visualization system. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5 (sept.-oct. 2006), 669 –676. 2

[AW10] AGGARWAL C. C., WANG H.: A survey of clustering algorithms for graph data. In *Managing and Mining Graph Data*, Aggarwal C. C., Wang H., Elmagarmid A. K., (Eds.), vol. 40 of *The Kluwer International Series on Advances in Database Systems*. Springer US, 2010, pp. 275–301. 2

[BAM07] BOURQUI R., AUBER D., MARY P.: How to draw clustered weighted graphs using a multilevel force-directed graph drawing algorithm. In *Information Visualization, 2007. IV '07. 11th International Conference* (july 2007), pp. 757 –764. 2

[BBD09] BECK F., BURCH M., DIEHL S.: Towards an aesthetic dimensions framework for dynamic graph visualisations. In *Information Visualisation, 2009 13th International Conference* (july 2009), pp. 592 –597. 1

[BRSG07] BENNETT C., RYALL J., SPALTEHOLZ L., GOOCH1 A.: The Aesthetics of Graph Visualization. In *Computational Aesthetics in Graphics, Visualization, and Imaging*, Cunningham D. W., Meyer G., Neumann L., (Eds.). 2007, pp. 1–8. 1

[CZQ*08] CUI W., ZHOU H., QU H., WONG P. C., LI X.: Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov.-Dec. 2008), 1277–1284. 2

[EH00] EADES P., HUANG M. L.: Navigating clustered graphs using force-directed methods. *Journal of Graph Algorithms and Applications 4* (2000), 157–181. 2

[Fen97] FENG Q.: *Algorithms for Drawing Clustered Graphs*. PhD thesis, University of Newcastle, 1997. 2

[FR91] FRUCHTERMAN T. M. J., REINGOLD E. M.: Graph drawing by force-directed placement. *Software - Practice and Experience 21*, 11 (November 1991), 1129–1164. 2

[HMM00] HERMAN I., MELANCON G., MARSHALL M. S.: Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics 6*, 1 (2000), 24–43. 1

[Hol06] HOLTEN D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (Sept.-Oct. 2006), 741–748. 2

[Hv09] HOLTEN D., VAN WIJK J. J.: Force-directed edge bundling for graph visualization. *Eurographics/IEEE-VGTC Symposium on Visualization (Eurovis)* (2009), 983–990. 1, 2, 3

[KS10] KIENREICH W., SEIFERT C.: An application of edge bundling techniques to the visualization of media analysis results. In *Proceedings of the 14 International Conference on Information Visualization (IV201)* (2010), IEEE Computer Society Press. 2

[LBA10] LAMBERT A., BOURQUI R., AUBER D.: Winding roads: Routing edges into bundles. *Computer Graphics Forum 29*, 3 (2010), 853–862. 1, 2, 3

[LYRL04] LEWIS D. D., YANG Y., ROSE T. G., LI F.: Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research, 5* (2004), 361–397. 3

[MSG10] MUHR M., SABOL V., GRANITZER M.: Scalable recursive top-down hierarchical clustering approach with implicit model selection for textual data sets. In *Proceedings of the 2010 Workshops on Database and Expert Systems Applications* (2010), pp. 5–19. 2, 3

[Noa07] NOACK A.: Energy models for graph clustering. *Journal of Graph Algorithms and Applications 11*, 2 (2007), 453–480. 2

[Sch07] SCHAEFFER S. E.: Graph clustering. *Computer Science Review 1*, 1 (2007), 27 – 64. 2

[SMER06] SHEN Z., MA K.-L., ELIASSI-RAD T.: Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1427–1439. 1

[WMC*09] WONG P. C., MACKEY P., COOK K., ROHRER R., FOOTE H., WHITING M.: A multi-level middle-out cross-zooming approach for large graph analytics. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on* (oct. 2009), pp. 147 –154. 4