

Using Phylogenetic Trees to Generate Semantic Meaningful Edge Bundles

Fabio Sikansi and Fernando V. Paulovich
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos, Brasil
fhenrique@usp.br, paulovic@icmc.usp.br

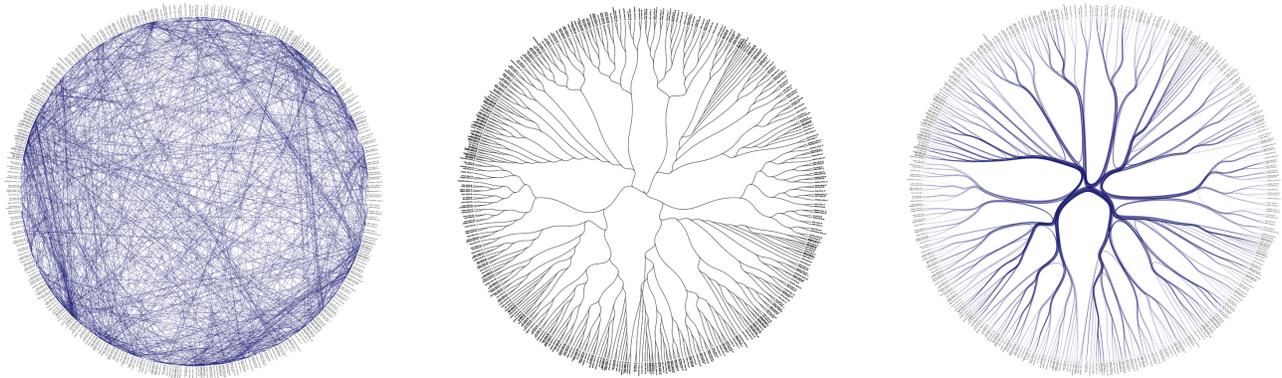


Fig. 1. Graph of citations in a subset of the 2014 IEEE Infovis Contest dataset, comprising papers published on the IEEE Infovis Conference between 1990 and 2014. An edge linking two papers means that one paper cites or was cited by the other one (left). The vertices are placed in the visual space using a radial layout algorithm (middle). The Neighbor Joining Bundling was used to produce a graph with clutter reduction and grouped edges (right).

Abstract—Graphs have been successfully applied in a range of problems and applications. It is the object of study of different areas, from modeling and analysis to the construction of visual representations. Different approaches exist for graph visualization, however, most of them suffer from the severe clutter when the number of nodes or edges is large. Amongst the techniques of graph visualization that handle such problem, Edge-bundling techniques attained relative success on improving the quality of the visual representations by bending and aggregating edges in order to reduce overlapping. Despite this success, most of them just perform the bundle based only on the visual space information, that is, there is no explicit connection between the produced bundled visual representation and the data. Some of them look upon edge information as a complement to the visual information, but it is just an additional measure, not their main goal. In this paper, we present a novel edge-bundling technique, called *Neighbor Joining Bundling (NJB)*, for graph visualization that tackles this problem by considering the similarity between the nodes when performing the edges bending.

Keywords—edge bundling; graph visualization;

I. INTRODUCTION

Graphs are applied in a range of problems, being useful to model different kinds of relationship between elements. The visual representation of data sets modeled as graphs is frequently used in data analytics, being the the node-link diagram the most common visual representation. However, graph visualization presents several challenges [1], [2]. Specially, when the number of vertices and edges is large, the

visualization suffers from the visual-clutter problem, reducing the power of data analysis. The clutter reduction is a frequent thread in different areas of study in data visualization [3] and aims at reorganizing or transforming the visual elements so that the attained representation can reveals patterns that are hidden on the original representation.

Among the graph visualization techniques, Edge-Bundling [4] has obtained great success reducing the visual-clutter on node-link diagrams. The main goal of this technique is to transform a straight node-link representation by bending and aggregating the edges using a set of control points. Edges that share control points are grouped, reducing overlapping thus the visual clutter. The smoothness provided by the curves also helps improving the data analysis in the visual representation.

After Holten [4], many others edge-bundling based techniques have been published, proposing different ways to execute the bundling and grouping the edges. Some examples include, strategies based on force-directed [5], geometry processing [6], [7], clustering [8] and image processing [9], [10], [11]. However, these techniques mainly use the visual space information to perform the edge bundling, ignoring the underlying data, thereby creating aggregations that do not explicitly reflect the data. In some techniques, the underlying data may be used as a complement to the visual information. However, this is not used as their main goal.

Contributions: In this paper, we propose a novel technique, called *Neighbor Joining Bundling (NJB)*, to build an edge bundled graph that employs a similarity phylogenetic tree to guide the edges bending and aggregation. This tree is also used to place the vertices on the visual space, thereby the resulting graph has its edges grouped and the nodes placed considering similarity relationships presented on the data set.

II. RELATED WORK

Edge bundling has been used to reduce visual clutter in different information visualization techniques [12]. The original idea was proposed by Holten [4], a technique called Hierarchical Edge Bundling (HEB), that employs hierarchical data to determine paths to guide the edges bending and grouping. The hierarchical information is also used to draw the vertices by a tree drawing algorithm, such as radial or hyperbolic. The edges are drawn as B-Splines curves using the intermediary points from the hierarchical tree as control points. Holten also presented a points transformation model that can control the smoothness of the bundling.

Most of following edge bundling techniques proposed methods to avoid the use of more information besides the adjacency data. The Geometry-Based Edge Clustering (GBEC) [6] allowed the clutter reduction in a common graph layout. It builds a mesh layout based on the original graph to group similar edges. However, the mesh construction can be a complex process, with high computational cost, and normally produced unpleasant layouts. The Winding Roads [7] approach processes the visual space information using a hybrid approach that combines QuadTree decomposition and a Voronoi diagram to discretize the visual space, creating roads when the edges are drawn.

In a different way, the Force Directed Edge Bundling (FDEB) [5] creates a system of forces over the edges and execute an iterative process, bending the edges, until it gets stable. Each edge is segmented in a set of points. The points are connected in a spring-mass system with the others points in the same edge. An electrostatic force manages the approximation between points in different edges. A compatibility edges measure is also used to avoid grouping different edges. The major problem of the force-directed model is the computational cost. Our approach presents a better trade-off between quality and computational cost.

Gansner et al. [8] propose a different approach to cluster edges using an ink-minimization algorithm. The algorithm groups the nearest edges until the amount of ink cannot be reduced. It was the first technique that used a quantitative measure to perform and evaluate the edge bundling.

A large group of edge bundling techniques is constituted by image processing edge-bundling approaches with GPU-based implementations. There are three techniques in that group. The Image Based Edge Bundling (IBEB) [9] processes a previous edge bundling layout to improve the graph visualization with better groups separation. Skeleton-Based Edge Bundling [10] uses image processing to create a skeleton that determines the paths of edges bending. Finally, the Kernel

Density Estimation Edge Bundling (KDEEB) [11] draws the bundled edges using the kernel density estimation process. Finally, the Attribute-Driven Edge Bundling (ADEB) [13] extends KDEEB that also uses numerical edge attributes to determine the bundling flow map. Common to almost all these techniques, the information contained on the data-set is not used to guide the edge bundling, so there is no connection between the visual representation and the underlying data. FDEB and SBEB look upon the use of edge information to measure edge compatibility, but it is not their main goal. Only ADEB effectively uses numerical edge attributes, e.g. weight, to perform the bundling. In this paper, any associated attribute with a vertex is used to guide the edge bundling. Our approach uses a similarity tree to guide the edges bending and grouping. In the following section, we detail our approach.

III. NEIGHBOR JOINING BUNDLING

Fig. 2 outlines the process to construct the *Neighbor Joining Bundling (NJB)*. It involves the computation of the bundling skeleton. To do this, a distance matrix is create considering the data, then the Neighbor Joining (NJ) [14] is executed to create a similarity tree that is used to define the control points which are used to guide the edge bending and to define the place of the nodes. In this section, we describe the NJ algorithm and the graph drawing approach based on the similarity tree structure.

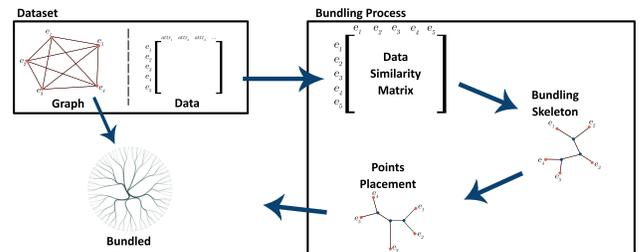


Fig. 2. Neighbor Joining Bundling overview

A. Neighbor Joining

The NJ algorithm creates a phylogenetic tree containing vertices that represents all the N data elements and other $N-2$ virtual vertices. It starts with the distance matrix D and a star-like tree, in which all data vertices are connected in one virtual vertex. Then, the algorithm finds a pair of vertices (i, j) by the criterion of minimum evolution, that is, the pair with the smallest sum of branch lengths $S_{i,j}$ given by the following equation:

$$S_{i,j} = \frac{1}{2(n-2)} \sum_{k \neq i,j}^n (D_{ik} + D_{jk}) + \frac{1}{2} D_{ij} + \frac{1}{n-2} \sum_{(k,l \neq i,j) \wedge (k < l)}^n D_{kl}$$

When the pair (i, j) is selected, a new virtual vertex X is created, with the vertices i and j as its children and connected to the common ancestor of i and j . The distance matrix D

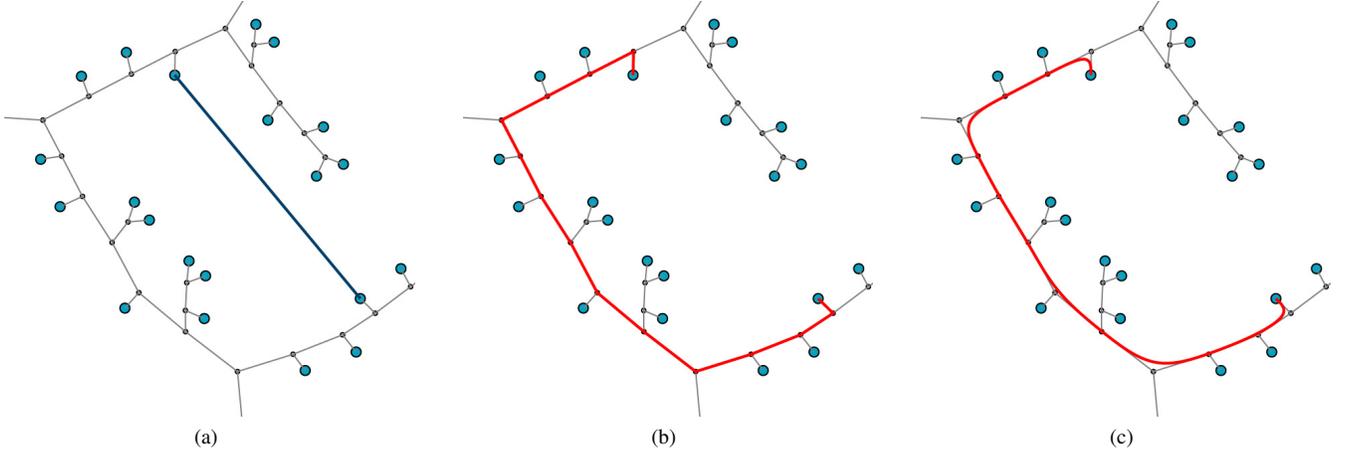


Fig. 3. Curving an edge through the NJ tree; First, the original edge is selected (a); the path between the source and the target is found in the NJ structure (b); a B-Spline curve is drawn using the NJ virtual vertices as control points (c).

is then updated by removing the distances that involves the vertices i and j , and adding the distances from X to the remaining vertices. In order to calculate these distances the following equation is used:

$$D_{i-j,k} = \frac{D_{ik} + D_{jk}}{2}$$

This process finishes when there are only two nodes remaining in D . At the end, the similarity tree will have the original N vertices of the data set and $N - 2$ virtual vertices. We use the implementation proposed by Studier et al. [15], in which NJ is obtained with complexity $O(V^3)$, where V is the number of vertices. Although the NJ algorithm is mainly used in biological studies, it had been applied in information visualization before. Cuadros [16] creates a NJ from collections of documents to perform a visual analysis of text similarity. Despite the requirement of a distance matrix, it can be generated by any kind of data associated with the vertices of the original graph.

B. Graph Drawing

Once we have the similarity tree, the first step is to place the points on the plane. To do this, we take advantage of the NJ structure and use an tree drawing algorithm. In this paper, we present our results using the radial layout [17]. With the radial layout, the vertices are placed around a circular structure, while the virtual vertices are placed in the middle of the visual space. Fig. 1 shows the NJ tree drawn using the radial layout algorithm.

Finally, the last process is to draw the edges. To do that, for each edge S we first find its source and target vertices. Then, we search the path between these vertices on the phylogenetic tree. This path is composed by virtual vertices, that are employed as control points to draw the edge S as a B-Spline curve. An example of this process is shown in the Fig. 3. Due to the characteristics of the NJ tree, the computational complexity to determine the control points is $O(V * E)$, where V is the number of vertices and E is the number of edges.

In our approach we allow users to change the edge straightening parameter, similar to the transformation proposed by [4]. This parameter (β) controls the movement of the control points of an edge S towards the original original straight edge, so that the curvature is reduced. The transformation of each curved edge S is given by the following equation:

$$S'(t) = S(t)\beta + (1 - \beta)(P_0 + t(P_{N-1} - P_0))$$

this is then applied in each control point by the equation:

$$P'_i = P_i\beta + (1 - \beta)(P_0 + \frac{i}{n}(P_{N-1} - P_0))$$

Small values of β produce a less distorted graph, but with small clutter reduction. On the other hand, β values close to 1 produce the best result in terms of clutter reduction, but with more distortion. There is no conclusive experiments about the best values of β , so it is useful to allow users to change this parameter interactively. Another parameter that can be changed on real-time is the global edge opacity (α). The edges opacity can be helpful to highlight the level of aggregation in each section of the bundling.

IV. RESULTS

We used *Neighbor Joining Bundling (NJB)* to visualize the graph of citations in a subset of the 2014 IEEE Infovis Contest dataset [18], comprising papers published on the IEEE Infovis Conference between 1990 and 2014. This has 490 vertices, representing the papers. Edges represent the citation between these papers, resulting in a set of 1,547 edges. To generate the required similarity matrix, we perform a text processing similar to what was presented on Cuadros [16].

Fig. 1 shows the graph with bundled edges. In this graph, we set the parameters $\beta = 0.98$ and $\alpha = 0.33$. In this visualization, it is possible to better identify, in distinct levels of details, the edges connections between the different groups of papers. Bundling benefits are better shown in graphs with more edges. The Fig. 4 shows a synthetic dataset with 175

vertices and 9,500 edges. Each edge has its color based on its original length. On the original drawn, with straight edges, it is difficult to identify patterns of connections, while on the bundled representation these patterns are better identified.

We developed a web-based environment with D3js [19], in which the user can interact with the visualization, by changing the visualization parameters (α and β), and highlighting edges by the selection of a vertex or group of vertices. This interaction is useful by allowing the user to analyze different aspects of the original graph.

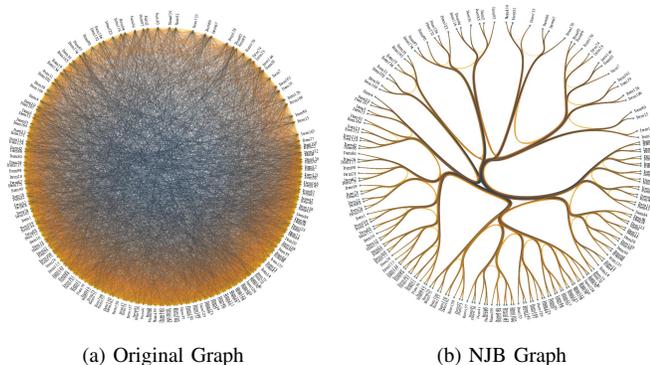


Fig. 4. NJB resulting graph for a synthetic data set with 175 vertices and 9,500 edges, each edge has its color based on its original size.

A. Limitation and Future Work

The NJB requires more than just an initial graph to construct the edge bundling layout. Since the NJ employs a distance matrix, a way to calculate distances between the vertices must be provided. If this is not provided, the NJB cannot be applied. NJB cannot also be applied to graphs with a given vertices placement, such as geographic information. However, the added semantic on bending the vertices according to similarity relationships can represent an improvement over other technique that execute the bundling without considering the data set.

Another limitation is its computational complexity. As future work, we intend reduce this complexity by creating an strategy that approximates the NJ but with a small complexity.

V. CONCLUSION

In this paper, we present a novel edge-bundling graph drawing technique to reduce clutter on graph visualization. Our technique, called *Neighbor Joining Bundling (NJB)*, uses a phylogenetic tree to guide the bending and grouping of edges. Different from the current state-of-the-art edge-bundling approaches, the bundled edges are obtained considering similarity relationships, potentially improving the data analysis for different kinds of data sets. Although our technique presents a high computational cost, it is considerable easier to implement, and it does not demand special hardware resources to be executed, such as GPUs. The results show that NJB might be useful in the visualization of similarity relationships on

graph data sets with vertices organized in different clusters, providing benefits not found in the existing techniques.

ACKNOWLEDGMENT

The authors thank to FAPESP (#2014/18665-1 and #2011/22749-8), Capes and CNPq for the financial support.

REFERENCES

- [1] I. Herman, G. Melançon, and M. S. Marshall, "Graph visualization and navigation in information visualization: A survey," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 6, no. 1, pp. 24–43, 2000.
- [2] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner, "Visual analysis of large graphs: State-of-the-art and future research challenges," in *Computer graphics forum*, vol. 30, no. 6. Wiley Online Library, 2011, pp. 1719–1749.
- [3] G. Ellis and A. Dix, "A taxonomy of clutter reduction for information visualisation," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1216–1223, 2007.
- [4] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 741–748, 2006.
- [5] D. Holten and J. J. Van Wijk, "Force-directed edge bundling for graph visualization," in *Computer Graphics Forum*, vol. 28, no. 3. Wiley Online Library, 2009, pp. 983–990.
- [6] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li, "Geometry-based edge clustering for graph visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 6, pp. 1277–1284, 2008.
- [7] A. Lambert, R. Bourqui, and D. Auber, "Winding roads: Routing edges into bundles," in *Computer Graphics Forum*, vol. 29, no. 3. Wiley Online Library, 2010, pp. 853–862.
- [8] E. R. Gansner, Y. Hu, S. North, and C. Scheidegger, "Multilevel agglomerative edge bundling for visualizing large graphs," in *Pacific Visualization Symposium (PacificVis), 2011 IEEE*. IEEE, 2011, pp. 187–194.
- [9] A. Telea and O. Ersoy, "Image-based edge bundles: Simplified visualization of large graphs," in *Computer Graphics Forum*, vol. 29, no. 3. Wiley Online Library, 2010, pp. 843–852.
- [10] O. Ersoy, C. Hurter, F. V. Paulovich, G. Cantareiro, and A. Telea, "Skeleton-based edge bundling for graph visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2364–2373, 2011.
- [11] C. Hurter, O. Ersoy, and A. Telea, "Graph bundling by kernel density estimation," in *Computer Graphics Forum*, vol. 31, no. 3pt1. Wiley Online Library, 2012, pp. 865–874.
- [12] H. Zhou, P. Xu, X. Yuan, and H. Qu, "Edge bundling in information visualization," *Tsinghua Science and Technology*, vol. 18, no. 2, pp. 145–156, 2013.
- [13] V. Peysakhovich, C. Hurter, and A. Telea, "Attribute-driven edge bundling for general graphs with applications in trail analysis," in *Visualization Symposium (PacificVis), 2015 IEEE Pacific*. IEEE, 2015.
- [14] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Molecular biology and evolution*, vol. 4, no. 4, pp. 406–425, 1987.
- [15] J. A. Studier, K. J. Keppler *et al.*, "A note on the neighbor-joining algorithm of saitou and nei," *Molecular biology and evolution*, vol. 5, no. 6, pp. 729–731, 1988.
- [16] A. M. Cuadros, F. V. Paulovich, R. Minghim, and G. P. Telles, "Point placement by phylogenetic trees and its application to visual analysis of document collections," in *IEEE VAST*, 2007, pp. 99–106.
- [17] C. Buchheim, M. Jünger, and S. Leipert, "Improving walker's algorithm to run in linear time," in *Graph Drawing*. Springer, 2002, pp. 344–353.
- [18] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko, "Visualization publication dataset," Dataset: <http://vispubdata.org/>, 2015, published Jun. 2015. [Online]. Available: <http://vispubdata.org/>
- [19] M. Bostock, V. Ogievetsky, and J. Heer, "D³ data-driven documents," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2301–2309, 2011.