TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 04/08 pp145-156
Volume 18, Number 2, April 2013

Edge Bundling in Information Visualization

Hong Zhou*, Panpan Xu, Xiaoru Yuan*, and Huamin Qu

Abstract: The edge, which can encode relational data in graphs and multidimensional data in parallel coordinates plots, is an important visual primitive for encoding data in information visualization research. However, when data become very large, visualizations often suffer from visual clutter as thousands of edges can easily overwhelm the display and obscure underlying patterns. Many edge-bundling techniques have been proposed to reduce visual clutter in visualizations. In this survey, we briefly introduce the visual-clutter problem in visualizations. Thereafter, we review the cost-based, geometry-based, and image-based edge-bundling methods for graphs, parallel coordinates, and flow maps. We then describe the various visualization applications that use edge-bundling techniques and discuss the evaluation studies concerning the effectiveness of edge-bundling methods. An edge-bundling taxonomy is proposed at the end of this survey.

Key words: edge bundling; visual clutter; graph visualization; parallel coordinates; flow maps

1 Introduction

In many visualization techniques, the edge is an important visual primitive for encoding data. For example, edges can encode relational data in graphs and multidimensional data in parallel coordinates plots. However, visualizations often suffer from visual clutter when the number of data items increases. The visual clutter caused by millions of edges not only affects the aesthetic quality of the representation, but also makes it difficult to obtain information from the visualization.

Many clutter reduction techniques, such as filtering, sampling, and clustering, have been presented to uncover patterns and trends in data with large numbers of items. An excellent survey on clutter reduction techniques can be found in a paper introduced by Ellis and Dix[1]. Recently, edge bundling has become a common clutter reduction technique in information visualization research. In edge bundling, similar edges are deformed and grouped into bundles, thus providing an abstract and uncluttered view of the original edge-cluttered visualization. Edge-bundling algorithms use geometric graphs, trees, or parallel coordinates plots, where each node has a predefined location, as inputs for generating the curved representations of each edge. Edge-bundling algorithms follow many different paradigms. Force-directed edge-bundling algorithms[2,3] and the visual clustering method[4] use physical force simulation. The geometry-based edge-bundling algorithm[5] and the winding roads method[6] discretize the visualization plane into grids and then search the plane for the configuration of each edge. Image-based edge-bundling algorithms[7,8] use data-clustering methods to build the edge hierarchy and then draw edge bundles as separate shaded shapes. Although these edge-bundling algorithms follow different paradigms and rely on

• Hong Zhou is with Shenzhen University, Shenzhen 518060, China. E-mail: hzhou@szu.edu.cn.
• Panpan Xu and Huamin Qu are with the Hong Kong University of Science and Technology, Hong Kong, China. E-mail: {pxu; huamin}@cse.ust.hk.
• Xiaoru Yuan is with Peking University, Beijing 100871, China. E-mail: xiaoru.yuan@pku.edu.cn.
∗ To whom correspondence should be addressed.
Manuscript received: 2012-12-08; accepted: 2013-02-14

different heuristics to compute curved edges, they enable high-level edge patterns to be more evident, thus reducing visual clutter.

Edge bundling is a promising approach for reducing visual clutter in information visualization. Therefore, this approach has recently been given increasing interest not only for graph visualization but also for flow map visualization, multidimensional visualization, and other types of visualizations. Despite the popularity of edge bundling, little effort has been devoted to specifically surveying and studying this technique. To our knowledge, this paper is the first attempt to gather, review, and analyze the vast body of research involving edge bundling.

In this survey, we briefly introduce the visual-clutter problem, which can be addressed by edge-bundling approaches. Then we review the cost-based, geometry-based, and image-based edge-bundling methods. After that, we describe the various visualization applications that use edge-bundling techniques and discuss the evaluation studies concerning the effectiveness of these methods. We propose an edge bundling taxonomy at the end of this survey.

2 Visual-Clutter Problem

Visual clutter occurs when excess items or their representations lead to congestion in limited screen areas. Edge-bundling approaches are introduced to address the visual-clutter problem caused by dense edges that overwhelm the display. In information visualization, the effectiveness of graphs, parallel coordinates, and flow maps is dramatically reduced because of edge clutter; in this section, we describe the visual-clutter problem in these three techniques.

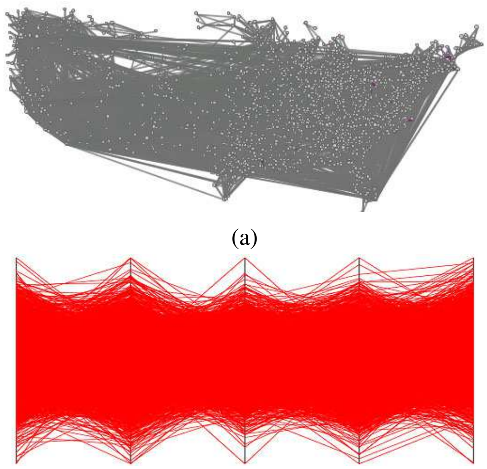
Graphs have been widely used to visualize complex relationships such as the trade relationships among cities and the friend relationships in social networks. A graph consists of nodes and edges where the nodes represent entities and the edges represent the relationships among the entities/nodes. For example, graphs can show vivid trade routes by representing cities as nodes and trade relationships as edges. However, real datasets are usually very large; thus, large data sizes often bring out the visual clutter problem in graphs. Understanding information in cluttered graphs where excessive edges obscure the underlying patterns is difficult. For example, Fig. 1a illustrates a real world graph that represents the migration between states in the United States.

Parallel coordinates[9, 10] are one of the most popular techniques for visualizing and analyzing multivariate data. In a parallel coordinates plot, the attributes of multivariate data are represented by parallel vertical axes that are linearly scaled within their data range; and each data item is represented by a polygonal line that intersects each axis at its respective attribute data value. Parallel coordinates are useful for the study of correlations between attributes by spotting the locations of intersection points. Parallel coordinates are also effective for revealing data distributions and functional dependencies. However, as the data size grows, parallel coordinates also experience the visual-clutter problem. Figure 1b shows a cluttered parallel coordinates plot.

Flow maps[11] are special graphs that indicate the movement of objects (e.g., people or goods) among the locations in maps. Flow maps comprise source nodes and target nodes. One or more (but not many) source nodes are linked to many target nodes by edges. The thickness of the edge represents the amount of flow between the source node and the target node. Flow maps also suffer from visual clutter caused by dense edges because this technique uses edges in visualizing the flow.

3 Edge-Bundling Techniques

Curved edges have been widely used to illustrate aggregated relationships between nodes in hand-drawn graphs. The automated approach, which uses attracted


(a)

(b)

Fig. 1 Visual-clutter examples: (a) a cluttered graph[5]; (b) a cluttered parallel coordinates plot[4].

control points in drawing curved edges, was first proposed by Brandes and Wagner[12]. However, the term edge bundling was formally introduced by other people several years later. Confluent graph drawing[13, 14] and flow map layouts[11] also bundle edges together in smooth curves to reduce edge crossings. To our knowledge, the term edge bundling was first formally used in a paper introduced by Holten[15]. In this paper[15], edge bundling is designed for graphs containing both hierarchical structures and adjacency relations. Nevertheless, the term edge bundling and the idea of bundling similar edges together to reduce visual clutter have recently gained considerable research attention. A number of following methods have been introduced for graphs[3,5,7] and other techniques[4, 16, 17].

Edge bundling is a promising approach for reducing visual clutter caused by dense edges in techniques such as graphs, parallel coordinates, and flow maps. In Section 2, we describe and explain the visual-clutter problem in these visualizations. In this Section, we classify edge-bundling algorithms into three categories based on the underlying paradigms: cost-based techniques, geometry-based techniques, and image-based techniques. In the edge-bundling stage of the aforementioned techniques, the shapes of the edges are changed but not the positions of the nodes. The positions of the nodes can have semantic meanings, such as geographical locations; these positions can also be computed by a layout method such as force-based models[18].

3.1 Cost-based techniques

In cost-based edge-bundling approaches, the costs of ink or energy are used to determine the shapes of curved edges.

3.1.1 Ink minimization

The circular graph layout is a visualization paradigm wherein all graph nodes are arranged in a circular pattern. Gansner and Koren[19] proposed three techniques for improving circular layout graphs. One of these techniques is edge-bundling. This technique deforms the edges that link the nodes placed on the perimeter of a circle and computes the drawing of curved edges by reducing the use of ink (i.e., total line length). In this approach, each edge is assigned two control points: the meeting point of the sources and the meeting point of the targets. Therefore, the shape of an edge is controlled by four points: the source node, the meeting point of the sources, the meeting point of the targets, and the target node. The positions of the source node and the target node cannot be changed in the bundling stage, whereas the two control points (i.e., the meeting point of the sources and the meeting point of the targets) can be moved freely. The edge-bundling algorithm bundles edges by placing the control points of similar edges close together. The optimized positions of the control points are computed by ink minimization, which is solved by using a numerical method.

Influenced by the paper introduced by Gansner and Koren[19], Pupyrev et al.[20] also built bundles by minimizing the total ink used for layered graph layouts. In Ref. [20], two or more control points are assigned to each edge, and edges are bundled by grouping the control points of similar edges. The positions of the control points can be changed to obtain global ink minimization results. Given that layered graph layouts are different from circular graph layouts, the global optimization procedure may be too time consuming. To accelerate the global optimization procedure, the method presents three lookup schemes to balance the time complexity and bundling quality. Figure 2 shows a bundled circular graph layout[19] and a bundled layered

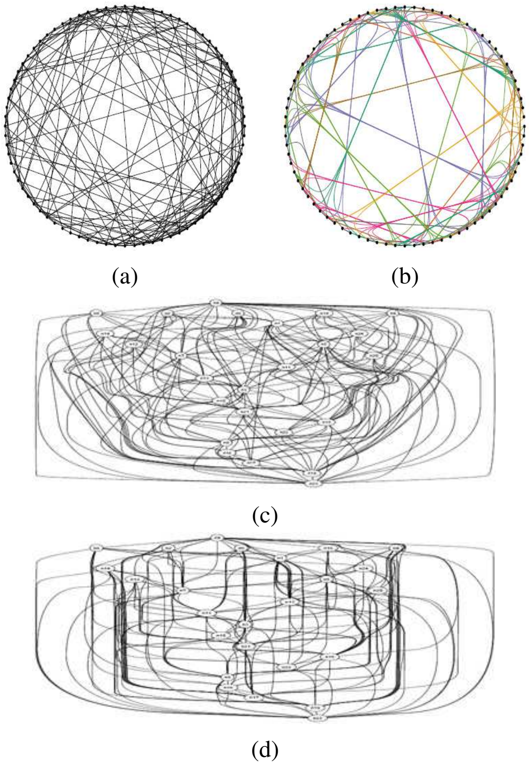


Fig. 2 Cost-based edge bundling with ink minimization: (a) before and (b) after applied on a circular graph layout[19]; (c) before and (d) after applied on a layered graph layout[20].

graph layout[20].

Gansner et al.[21] recently proposed a multi-level agglomerative edge-bundling method for general graph layouts instead of certain kinds of graphs such as circular graph layouts and layered graph layouts. Their approach uses ink-saving as a guiding principle and link curvature as an additional constraint. The algorithm in this approach is faster than previous ink minimization methods. An edge proximity graph is first constructed to guide the bundling decisions. An edge is bundled with its neighbors in the proximity graph only if the bundling saves ink. When all possible bundling edge pairs are identified and bundled, a newly bundled graph and its corresponding edge proximity graph are built. The edges in the newly bundled graph are grouped further based on the new edge proximity graph. The bundling process is repeated until all ink-saving opportunities have been exhausted. Based on the idea of bundling edges under ink minimization, the ordered-bundling method[22] attempts to draw the edges of each bundle as parallel as possible with a given gap.

3.1.2 Energy minimization

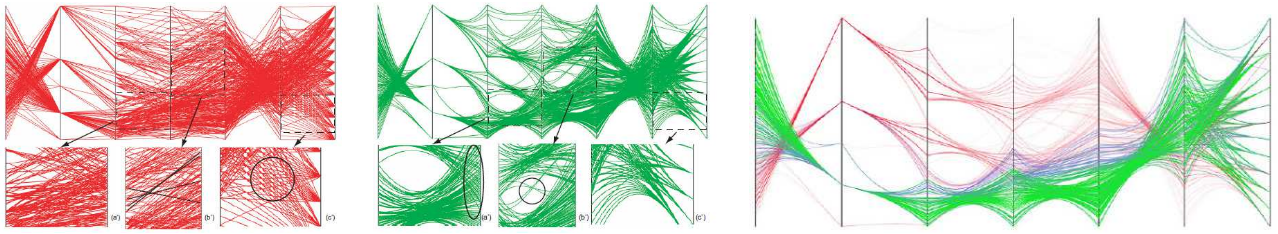
Instead of minimizing ink, many methods minimize the energy of the spring models generated for drawing systems.

Parallel coordinates. Zhou et al.[4] modeled the lines in parallel coordinates plots as flexible springs and applied attractive forces between lines. The attractive force tends to cluster spatially close and visually parallel lines together. Therefore, the attractive force is computed based on the Euclidian distance and the angular distance between nearby lines. The curvature energy term prevents each line from being excessively bent. An energy function was defined for this physical system, and the shapes of the lines were computed to minimize the energy function. The global optimization result was searched by linear programming. The bundled lines can be further enhanced by varying the color and opacity according to the local line density of the curves (see Fig. 3).

Graphs. The above-mentioned energy system, which considers the locations and directions of edges during the edge-bundling stage, is further extended for general graphs[2]. In parallel coordinates, nodes linked by lines are arranged on parallel axes. In general graphs, nodes are usually not arranged in certain patterns. Therefore, the sampling of edges and the movement of control points in general graphs are more complex compared with parallel coordinates. To address this problem, Zhou et al.[2] sampled graph edges into segments by using Delaunay triangulation to obtain the control points. This non-uniform sampling maintains the topology of the graph and allows the consideration of the midpoints of each sampled segment as nodes in a control map for bundling. The edges in the control map are modeled as springs with energy that attracts spatially close and visually parallel segments together. When the nodes in the control map are merged pair by pair, the edges in the original graph are bundled accordingly. The control map updates after each merging. The merging is repeated until all node pairs that can reduce the energy of the system are eliminated.

Holten and van Wijk[3] presented another force-directed edge-bundling algorithm where edges are modeled as flexible springs and electrostatic-attracting forces exist between the pairs of edges (see Fig. 4). A set of edge compatibility measures is used to prevent excessive bundling. An iterative computation scheme is applied to obtain the deformation of the edges. Finally, a smoothing tool is used in all edges by convolving the positions of the control points under a Gaussian kernel. The bundling results can intuitively show visible high-level patterns in graphs even though the algorithm has high time complexity.

Divided edge bundling[23] was developed based



(a) (b) (c)

Fig. 3 Energy-based edge bundling in a parallel coordinates plot[4]: (a) original plot; (b) after edge bundling; (c) with color and opacity enhancement.

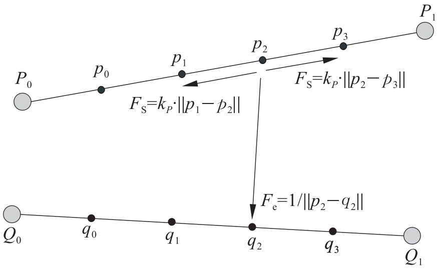
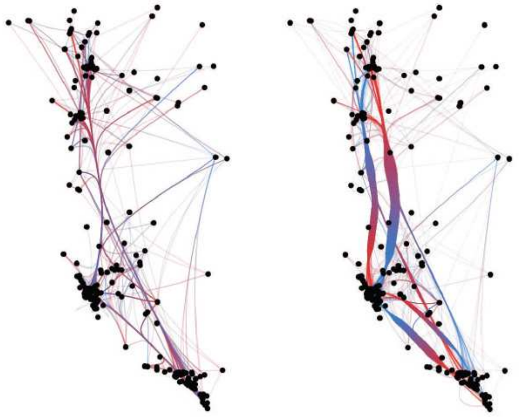


Fig. 4 The spring force F_s and the electrostatic force F_e which are exerted on control point p_2 by p_1, p_3, and q_2 are shown in Ref. [3].



(a) (b)

Fig. 5 Energy-based edge-bundling results for the same data: (a) force-directed edge bundling[3]; (b) divided edge bundling[23].

on force-directed edge bundling[3] with a set of extensions. To include the direction, weight, and graph connectivity in the bundling, divided edge bundling[23] incorporates directional lanes, connectivity compatibility, and edge-weight terms in the force model. The directional lanes and edge weights are designed to help users spot asymmetries more easily, whereas the connectivity compatibility is designed to prevent spurious inferences. This approach can reveal some patterns that are only discernable in a matrix view. Figure 5 shows a force-directed edge-bundling result[3] and its corresponding divided-edge-bundling result[23].

The edge-bundling framework[24] introduced by Nguyen et al. is based on the force-directed edge-bundling method[3]. This approach[3] presents four variations of the previous force-directed edge-bundling technique and two new edge compatibility measures based on network analysis and topology.

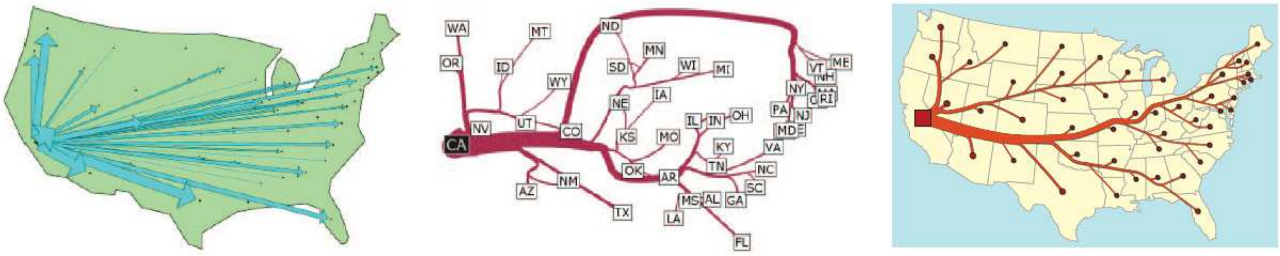
Flow maps. Verbeek et al.[17] presented an automated technique to generate edge-bundled flow maps. This method is based on spiral trees that are introduced in a companion paper[25]. A spiral tree is an angle-restricted Steiner tree, which is a cross-free tree with 120 degree angles at every internal control point. Computing optimal spiral trees is NP-hard. To accelerate the computation process and smooth the main branches in the spiral tree, Verbeek et al.[17] minimized a global cost function that consists of five costs (i.e., obstacle cost, smoothing cost, angle restriction cost, balancing cost, and straightening cost). They minimized the global cost function by moving control points and determining every local minimum. Although this approach is unable to search for the global optimal results, it can generate many good flow maps (see Fig. 6c).

3.2 Geometry-based techniques

In geometry-based edge-bundling approaches, trees or grids are used to determine the shapes of curved edges.

3.2.1 Tree-based edge bundling

Phan et al.[11] proposed an algorithm to bundle edges in single-source flow maps based on the primary hierarchical clustering of target nodes. Primary hierarchical clustering yields a tree structure of the input flow. Original straight edges are further curved and routed around the bounding boxes within the same hierarchical cluster. Good bundling results can be drawn; however, this approach may not smooth all curves when routing them around the bounding boxes



(a) (b) (c)

Fig. 6 Flow maps illustrating the same migration data: generated by (a) Tobler[26]; (b) Phan et al.[11]; (c) Verbeek et al.[17]

(see Fig. 6b).

Holten[15] proposed the use of edge bundling to visualize graphs with edges that represent adjacency relations in hierarchical data. Each edge is curved according to the hierarchical tree path that links the two endpoints of the edge. Two edges will be bundled at their common segments in their tree paths. This method demonstrates the effectiveness of edge bundling in clutter reduction (see Fig. 7). However, this technique is designed for graphs with known hierarchical structures.

3.2.2 Grid-based edge bundling

Meaningful hierarchical trees may not exist or may be difficult to compute for general graphs. Therefore, different kinds of grids are proposed and used to guide the edge-bundling process.

Triangle meshes. Qu et al.[27] introduced an edge-bundling method that is based on the control mesh generated by Delaunay triangulation. By adjusting the control mesh, the bundling results can be interactively controlled and the edges can be progressively clustered. The vertices of the control mesh are the nodes of the original graphs. The edges of the control mesh are computed by using Delaunay triangulation which takes the vertices as input. The intersections of the Delaunay edges and graph edges are set as the control points for the bundling. Thereafter, the graph edges are converted into a series of paths that link these control points. Different levels of edge bundling can be constructed by adjusting the merging of these control points. The algorithm designed by Zhou et al.[2] also adopts the Delaunay triangulation to segment graph edges; however, this algorithm uses the midpoints of each segments as control points and builds an energy minimization system to find optimal edge-bundling results.

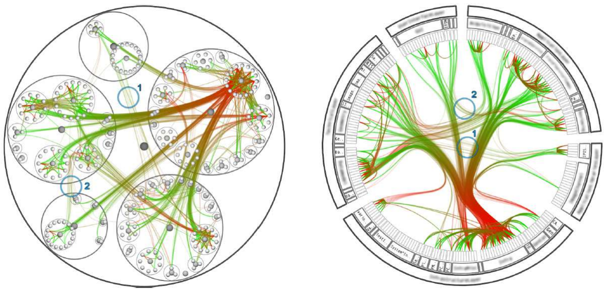

(a) (b)

Fig. 7 Hierarchical edge bundling[15] applied on: (a) a balloon layout; (b) a radial layout.

The above-mentioned methods generate many zigzag edges, which make the tracing of curves and the discerning of end points difficult for large graphs. Furthermore, Delaunay triangulation does not work for many graphs. To address this problem, Cui et al.[5] proposed another mesh generation method based on the underlying geometry of the graph. Figure 8 illustrates the framework of this technique. The information of the input graph is first sent to an analyzer to detect the directions of the representative primary edge in the graph. The mesh generator then generates some mesh edges perpendicular to the directions of the primary edge. The mesh generator then completes the control mesh by triangulating the mesh edges and some added nodes. Based on this control mesh, the edges of the original graphs are bundled to cross the control mesh edges in a cumulative manner, thus allowing the grouping and abstraction of large amounts of edges. Finally, an edge smoother reduces the number of zigzags in curved edges, and a visualizer provides advanced interaction techniques to explore the edge-bundling results further.

Non-uniform grids. Lambert et al.[6] proposed a method that uses a hybrid grid to guide the bundling process. The hybrid grid is computed based on both quadtrees and Voronoi diagrams. The size of quadtree cells is selected to generate different levels of edge bundling, and then Voronoi diagrams are used to construct the final grid graph. Finally, the graph edges are routed along the grid edges, and a self-organizing scheme is proposed to route similar edges along similar

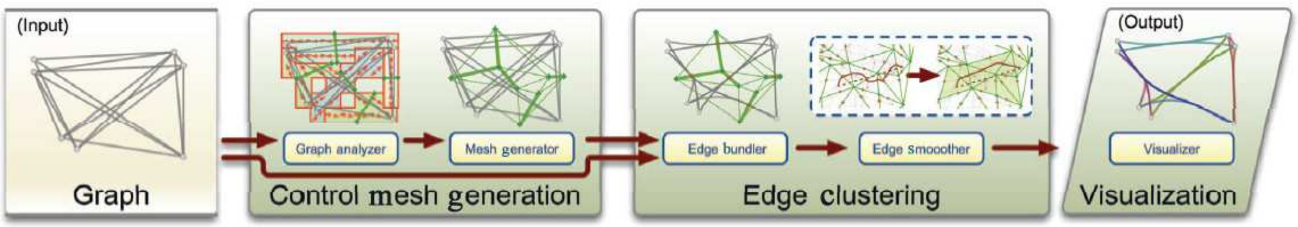


Fig. 8 The edge-bundling framework proposed by Cui et al.[5]

mesh edges. The method is further extended for 3D edge bundling by using a combination of octrees and 3D Voronoi diagrams in the grid-computation stage[28]. The computing power of GPUs is exploited to achieve real-time exploration of edge-bundled graphs[29]. To visualize yeast metabolic networks, Lambert et al.[30] presented another extension that can avoid twisty routes and respect certain edge-drawing constraints by using only a quadtree in the grid generation step. Figure 9 illustrates the same yeast metabolic networks bundled by winding roads[6] and pathway preserving representation[30].

Luo et al.[31] presented an ambiguity-free edge-bundling framework based on the quadtree partition (see Fig. 10). A quadtree is constructed according to the locations of the graph nodes. The information of each quadtree cell is occupied by which edge(s) and node(s) is/are recorded. The edge-ambiguity problem may occur when a certain edge passes near one or more unrelated nodes. Therefore, quadtree cells that may cause the edge-ambiguity problem are detected. The control points for routing ambiguity edges away from unrelated nodes are calculated based on the detected quadtree cells. The method curves and bundles the original straight edges to pass through these control points. An interactive detail-on-demand tool is further provided for the user exploration.

3.3 Image-based techniques

Image-based edge-bundling algorithms have been proposed for parallel coordinates or graphs with edge clusters computed by well-known data-clustering methods. Many image-enhancement tools have also been introduced to render the result of edge bundling.

Illustrative parallel coordinates[16] are a set of rendering techniques that are used to reveal interesting patterns in an artistic way. Edge clusters are detected using k-means clustering and are rendered by using splines or branches. A branch illustrates one edge cluster as a pair of spline curves that binds a group of quadrilaterals. Given that edge clusters usually overlap each other, silhouettes, shadows, and halos effects are further exploited to help users distinguish overlapping branches. Other image-based rendering tools such as faded histograms can be used to show the data density in each branch and density plots can be used to provide a high-level view of the data distribution. Figure 11 shows several effects generated by the illustrative methods.

Telea and Ersoy[7] proposed an image-based edge-bundling framework by combining distance-based

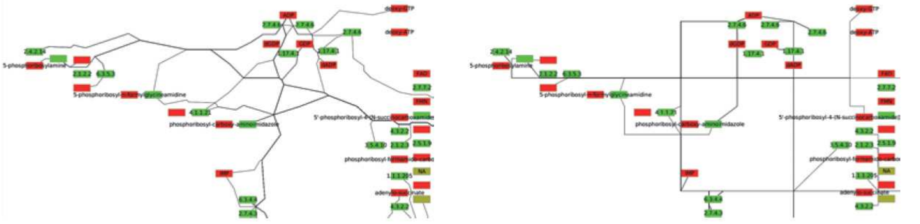

(a) (b)

Fig. 9 The same yeast metabolic networks bundled by: (a) winding roads[6]; (b) pathway preserving representation[30].

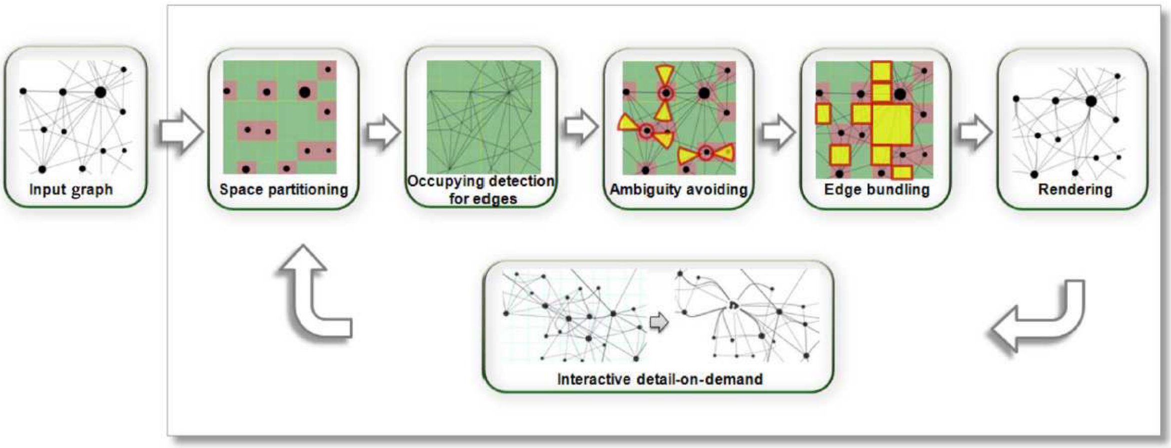
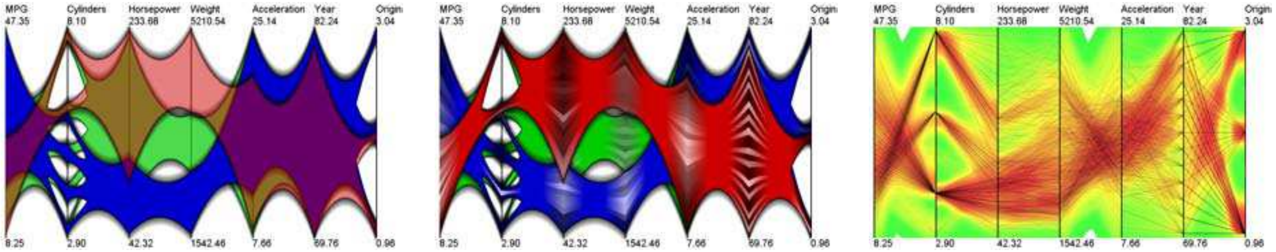


Fig. 10 The ambiguity-free edge-bundling framework[31].



(a) (b) (c)

Fig. 11 Several effects generated by illustrative parallel coordinates[16]: (a) branched clusters; (b) faded histograms; (c) density plots.

splatting and shape skeletonization. Given an edge-bundling layout computed by other methods, they constructed shaded shapes to visualize each edge set. The skeleton of each shape is computed and rendered in a convex style with dark and saturated borders and bright and white middles. The rendering order of all shaded shapes is determined by minimizing the occlusion of the shape. Finally, several interaction techniques such as brushing and digging lenses are introduced to explore overlapping bundles.

Ersoy et al.[8] adopted the idea of computing the skeletons of edge clusters[7]. They also extended the idea of computing skeletons of edge clusters by using a new skeleton-construction algorithm and a simple rendering technique. Instead of using skeletons only for shading, this method exploits skeletonization together with edge clustering and distance fields to construct bundled-edge layouts. Edges are iteratively curved and attracted towards the skeleton of the edge clusters; thus, individual edges are visible as alpha-blended curves. Figure 12 shows the different rendering results generated by the image-based edge-bundling framework[7] and the skeleton-based edge-bundling method[8].

Image-based edge-bundling techniques are rendered based on the given edge clusters. Therefore, these methods can be used together with other edge-bundling layout approaches. Many image-based rendering tools exist for the enhancement of bundled layouts: color and opacity enhancement based on the line density or edge direction[5], RGB interpolated color gradient to indicate the original edge directions[15], transparency or hue for local edge density[6], and rendering styles of convex shapes, density-luminance, density-saturation, bi-level, and outlines[7].

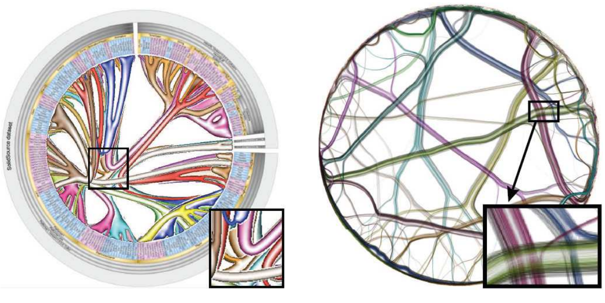

(a) (b)

Fig. 12 Rendering results generated by: (a) image-based edge bundling[7]; (b) skeleton-based edge bundling[8].

4 Edge-Bundling Applications

Edge-bundling algorithms have been proposed for many visualization techniques and applications. Edge-bundling approaches designed for graphs, parallel coordinates, and flow maps have been discussed in Section 3. In the case study sections of previously discussed edge-bundling works, many applications have been introduced, such as US migration data[3,5,6,8], US airline graph[3,5,8], and car information dataset[4,16]. The US migration data has 1790 nodes and 9798 edges. Figure 13 shows the US migration graphs generated by methods presented by Holten and van Wijk[3], Cui et al.[5], Lambert et al.[6], and Ersoy et al.[8] The US airline graph has 235 nodes and 2101 edges. The Car information dataset has 7 variables and 392 data items. In addition, edge-bundling methods have been widely adopted by other techniques for analyzing data, such as dynamic networks[32], one-to-many matched graphs[33], classificatory distribution in patent portfolios[34], and so on[35,36].

5 Edge-Bundling Studies

Numerous user studies have been conducted on the evaluation of graph visualization techniques[37]. In most of these studies, different visualizations were generated by different techniques for the same tasks. Users were asked to conduct tasks while their completion time and accuracy were recorded for comparison

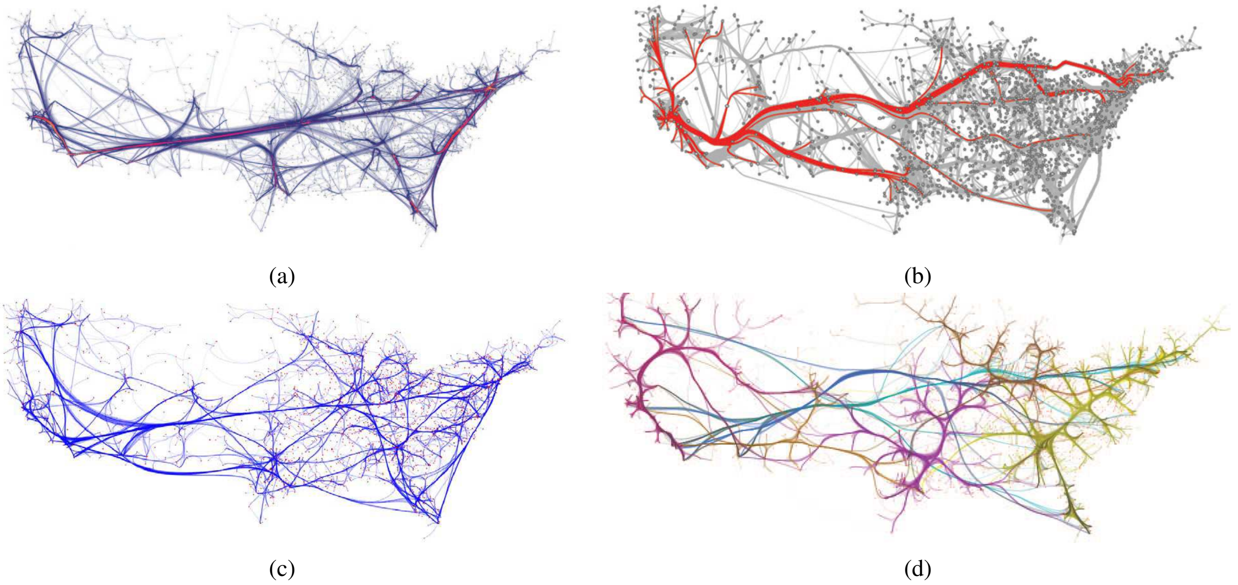


Fig. 13 US migration graphs generated by methods presented by: (a) Holten and van Wijk[3]; (b) Cui et al.[5]; (c) Lambert et al.[6]; (d) Ersoy et al.[8]

and analysis. Many important results were obtained from these studies and were further used to guide visualization applications.

However, few works have systematically examined edge-bundling techniques. Some related works only include a small number of studies on visualizations with curved edges.

Holten and van Wijk[38] tested six single-cue directed-edge representations, including a clockwise-curvature representation, and provided recommendations for the representation of directed edges in node-link diagrams. Holten and van Wijk[39] performed another user study to evaluate the cluster identification performance of nine parallel coordinates techniques. These parallel coordinates techniques differ in the usage of scatter plot embedding, the enhancement of fuzzy clusters, animations, and curved lines. Participants were asked to identify the number of clusters in 144 trials, and the response time and accuracy were recorded. Their results show that parallel coordinates with curved lines do not result in significant performance gains.

Telea et al.[40] compared hierarchical edge bundling[15] against node-link diagrams with software engineering related data. They found that all participants strongly prefer hierarchical edge bundling. However, whether this finding is also applicable to other kinds of data is not clear.

Xu et al.[41] recently conducted two experiments to study the effect of edge curvature on the readability of general graphs. The first experiment compared graphs with different edge-curvature levels (i.e., straight edge, slightly curved edge, and heavily curved edge). Their results show that high-curvature edges are detrimental to performance. The second experiment compared graphs with varying edge curvatures and graph sizes. The results of this experiment state that the straight edge is the most aesthetically pleasing and effective edge type and that edges with varying low curvatures are good alternatives to straight edges when curved edges are preferred. These results are limited by the experimental parameters and designed tasks. Some of these conclusions may not be applicable for the identification of clusters and overall comparison of pattern-searching tasks. Edge-bundling methods may have certain advantages when utilized with useful interaction tools. Therefore, systematic studies need to be conducted to assess the effectiveness of edge-bundling approaches.

6 Conclusions

Edge bundling has emerged as a promising and important technique for reducing visual clutter in visualizations. In this survey, we examined recent works regarding the edge-bundling technique. We first studied the visual-clutter problem, which can be addressed by edge bundling. Second, we classified edge-bundling techniques into three categories: cost-based techniques, geometry-based techniques, and image-based techniques. Finally, we reviewed edge-bundling-related applications and user studies.

We built an edge-bundling taxonomy based on our analysis (Table 1). In this table we take all the edge-bundling papers mentioned in Sections 3 and 5, and classified them in terms of the applied techniques (i.e., graphs, parallel coordinates, and flow maps). The sub-columns of the edge-bundling-techniques column are determined based on the classification defined in Section 3. Several cells are marked as "None", because no algorithm can be classified into this category. These categories with "None" mark maybe the future research directions for the development of edge bundling.

Despite gaining considerable research attention, edge bundling still has many challenging problems that need to be addressed. Some of these challenges include readability, algorithm complexity, and intuitive navigation issues. We hope that the analysis and taxonomy introduced in this survey will motivate and facilitate continued research in the development of edge-bundling techniques.

Acknowledgements

We thank anonymous reviewers for their valuable comments. This work is supported by Foundation for Distinguished Young Talents in Higher Education of Guangdong, China (No. LYM11113), the National Natural Science Foundation of China (Nos. 61103055 and 61170204, and 61232012).

References

[1] G. Ellis and A. Dix, A taxonomy of clutter reduction for information visualisation, *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1216-1223, 2007.

[2] H. Zhou, X. Yuan, W. Cui, H. Qu, and B. Chen, Energy-based hierarchical edge clustering of graphs, in *Proceedings of IEEE Pacific Visualization Symposium*, Kyoto, Japan, 2008, pp. 55-61.

[3] D. Holten and J. J. van Wijk, Force-directed edge bundling for graph visualization, *Computer Graphics Forum*, vol. 28, no. 3, pp. 983-990, 2009.

[4] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen, Visual clustering in parallel coordinates, *Computer Graphics Forum*, vol. 27, no. 3, pp. 1047-1054, 2008.

[5] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li, Geometry-based edge clustering for graph visualization, *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1277-1284, 2008.

[6] A. Lambert, R. Bourqui, and D. Auber, Winding roads: Routing edges into bundles, *Computer Graphics Forum*, vol. 29, no. 3, pp. 853-862, 2010.

[7] A. Telea and O. Ersoy, Image-based edge bundles: Simplified visualization of large graphs, *Computer Graphics Forum*, vol. 29, no. 3, pp. 843-852, 2010.

[8] O. Ersoy, C. Hurter, F. V. Paulovich, G. Cantareira, and A. Telea, Skeleton-based edge bundling for graph visualization, *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2364-2373, 2011.

[9] A. Inselberg, The plane with parallel coordinates, *The Visual Computer*, vol. 1, no. 2, pp. 69-91, 1985.

[10] A. Inselberg and B. Dimsdale, Parallel coordinates: A tool for visualizing multi-dimensional geometry, in *Proceedings of IEEE Visualization*, San Francisco, USA, 1990, pp. 361-378.

[11] D. Phan, L. Xiao, R. Yeh, P. Hanrahan, and T. Winograd, Flow map layout, in *Proceedings of IEEE Symposium on Information Visualization*, Minneapolis, USA, 2005, pp. 219-224.

[12] U. Brandes and D.Wagner, Using graph layout to visualize train interconnection data, in *Proceedings of International Symposium on Graph Drawing*, Montréal, Canada, 1998, pp. 44-56.

[13] M. Dickerson, D. Eppstein, M. T. Goodrich, and J. Y. Meng, Confluent drawings: Visualizing non-planar diagrams in a planar way, *Journal of Graph Algorithms and Application*, vol. 9, no. 1, pp. 31-52, 2005.

[14] D. Eppstein, M. Goodrich, and J. Meng, Confluent layered drawings, in *Proceedings of International Symposium on Graph Drawing*, New York, USA, 2004, pp. 184-194.

[15] D. Holten, Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data, *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 741-748, 2006.

[16] K. T. McDonnell and K. Mueller, Illustrative parallel coordinates, *Computer Graphics Forum*, vol. 27, no. 3, pp. 1031-1038, 2008.

[17] K. Verbeek, K. Buchin, and B. Speckmann, Flow map layout via spiral trees, *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2536-2544, 2011.

[18] A. Noack, An energy model for visual graph clustering, in *Proceedings of International Symposium on Graph Drawing*, Perugia, Italy, 2003, pp. 425-436.

Table 1 Edge bundling taxonomy.

Examples	Edge-bundling techniques					Edge-bundling studies
	Cost-based		Geometry-based		Image-based	
	Ink minimization	Energy minimization	Tree-based	Grid-based		
Graphs	[19-22]	[2,3,23,24]	[15]	[2,5,6,27-31]	[5-8,15]	[38,40,41]
Parallel coordinates	None	[4]	None	None	[16]	[39]
Flow maps	[17]	[11]	None	None	None	None

[19] E. R. Gansner and Y. Koren, Improved circular layouts, in *Proceedings of International Symposium on Graph Drawing*, Karlsruhe, Germany, 2006, pp. 386-398.

[20] S. Pupyrev, L. Nachmanson, and M. Kaufmann, Improving layered graph layouts with edge bundling, in *Proceedings of International Symposium on Graph Drawing*, Konstanz, Germany, 2010, pp. 329-340.

[21] E. R. Gansner, Y. Hu, S. North, and C. Scheidegger, Multilevel agglomerative edge bundling for visualizing large graphs, in *Proceedings of IEEE Pacific Visualization Symposium*, Hong Kong, China, 2011, pp. 187-194.

[22] S. Pupyrev, L. Nachmanson, S. Bereg, and A. E. Holroyd, Edge routing with ordered bundles, in *Proceedings of International Symposium on Graph Drawing*, Eindhoven, The Netherlands, 2011, pp. 136-147.

[23] D. Selassie, B. Heller, and J. Heer, Divided edge bundling for directional network data, *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2354-2363, 2011.

[24] Q. Nguyen, S.-H. Hong, and P. Eades, Tgi-eb: A new framework for edge bundling integrating topology, geometry and importance, in *Proceedings of International Symposium on Graph Drawing*, Eindhoven, The Netherlands, 2011, pp. 123-135.

[25] K. Buchin, B. Speckmann, and K. Verbeek, Anglerestricted steiner arborescences for flow map layout, in *Proceedings of the ISAAC*, Yokohama, Japan, 2011, pp. 250-259.

[26] W. R. Tobler, Experiments in migration mapping by computer, *Cartography and Geographic Information Science*, vol. 14, no. 2, pp. 155-163, 1987.

[27] H. Qu, H. Zhou, and Y. Wu, Controllable and progressive edge clustering for large networks, in *Proceedings of International Symposium on Graph Drawing*, Karlsruhe, Germany, 2006, pp. 399-404.

[28] A. Lambert, R. Bourqui, and D. Auber, 3D edge bundling for geographical data visualization, in *Proceedings of International Conference on Information Visualization*, London, UK, 2010, pp. 329-335.

[29] A. Lambert, D. Auber, and G. Melancon, Living flows: Enhanced exploration of edge-bundled graphs based on gpu-intensive edge rendering, in *Proceedings of International Conference on Information Visualization*, London, UK, 2010, pp. 523-530.

[30] A. Lambert, J. Dubois, and R. Bourqui, Pathway preserving representation of metabolic networks, *Computer Graphics Forum*, vol. 30, no. 3, pp. 1021-1030, 2011.

[31] S.-J. Luo, C.-L. Liu, B.-Y. Chen, and K.-L. Ma, Ambiguity-free edge-bundling for interactive graph visualization, *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 5, pp. 810-821, 2012.

[32] L. Shi, C. Wang, and Z. Wen, Dynamic network visualization in 1.5D, in *Proceedings of IEEE Pacific Visualization Symposium*, Hong Kong, China, 2011, pp. 179-186.

[33] E. D. Giacomo, W. Didimo, G. Liotta, and P. Palladino, Visual analysis of one-to-many matched graphs, *Journal of Graph Algorithms and Applications*, vol. 14, no. 1, pp. 97-119, 2010.

[34] M. Giereth, H. Bosch, and T. Ertl, A 3D treemap approach for analyzing the classificatory distribution in patent portfolios, in *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, Columbus, USA, pp. 189-190, 2008.

[35] B. Cornelissen, A. Zaidman, D. Holten, L. Moonen, A. van Deursen, and J. J. van Wijk, Execution trace analysis through massive sequence and circular bundle views, *Journal of Systems and Software*, vol. 81, no. 12, pp. 2252-2268, 2008.

[36] T. Moscovich, F. Chevalier, N. Henry, E. Pietriga, and J.-D. Fekete, Topology-aware navigation in large networks, in *Proceedings of the SIGCHI*, Boston, USA, 2009, pp. 2319-2328.

[37] D. Holten, P. Isenberg, J. J. van Wijk, and J.-D. Fekete, An extended evaluation of the readability of tapered, animated, and textured directededge representations in node-link graphs, in *Proceedings of IEEE Pacific Visualization Symposium*, Hong Kong, China, 2011, pp. 195-202.

[38] D. Holten and J. J. van Wijk, A user study on visualizing directed edges in graphs, in *Proceedings of the SIGCHI*, Boston, USA, 2009, pp. 2299-2308.

[39] D. Holten and J. J. van Wijk, Evaluation of cluster identification performance for different pcp variants, *Computer Graphics Forum*, vol. 29, no. 3, pp. 793-802, 2009.

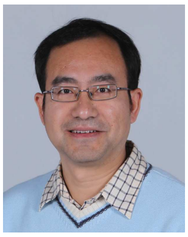
[40] A. Telea, O. Ersoy, H. Hoogendorp, and D. Reniers, Comparison of node-link and hierarchical edge bundling layouts: A user study, presented at Visualization and Monitoring of Network Traffic, 2009.

[41] K. Xu, C. Rooney, P. Passmore, D.-H. Ham, and P. H. Nguyen, A user study on curved edges in graph visualization, *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2449-2456, 2012.

Hong Zhou obtained a BEng degree in computer science and technology from Zhejiang University, China, in 2004 and a PhD degree in computer science and engineering from the Hong Kong University of Science and Technology in 2009. She is a lecturer in the College of Computer Science and Software Engineering at Shenzhen University. Her main research interests are in visualization and computer graphics.

Panpan Xu received a BEng degree in computer science and technology from Zhejiang University, China, in 2009. She is a PhD candidate in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. Her research interests are in information visualization and visual analytics.

Xiaoru Yuan received his BS degree in chemistry and BA degree in law from Peking University in 1997 and 1998 respectively. In 2005 and 2006, he received his MS degree in computer engineering and PhD degree in computer science at University of Minnesota at Twin Cities. He is now a faculty member at Peking University, in the Laboratory of Machine Perception (MOE). His primary research interests lie in the field of visualization and visual analytics with emphasis on large data visualization, high dimensional data and graph visualization, and novel visualization user interface.



Huamin Qu obtained a BS in mathematics from Xi'an Jiaotong University, China, an MS degree and a PhD degree (2004) in computer science from the Stony Brook University. He is an associate professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. He is in the steering committee of the IEEE Pacific Visualization Conferences, and served as the program co-chair for IEEE PacificVis 2011 and 2012 and the conference co-chair for VINCI 2011 and VINCI 2012. He receives Honorable Mention for Best Paper Award at IEEE Visualization 2009 and is a winner of 2009 IBM Faculty Award. His main research interests are in visualization and computer graphics.