Exploring the Design Space of Interactive Link Curvature in Network Diagrams

Nathalie Henry Riche[†], Tim Dwyer[‡], Bongshin Lee[†]
[†]Microsoft Research, [‡]Microsoft Corp.
One Microsoft Way, Redmond, WA, 98052
{nath,timdwyer,bongshin}@microsoft.com

Sheelagh Carpendale[†§]
[§]University of Calgary
2500 University Dr. NW, Calgary, AB, Canada
sheelagh@cpsc.ucalgary.ca

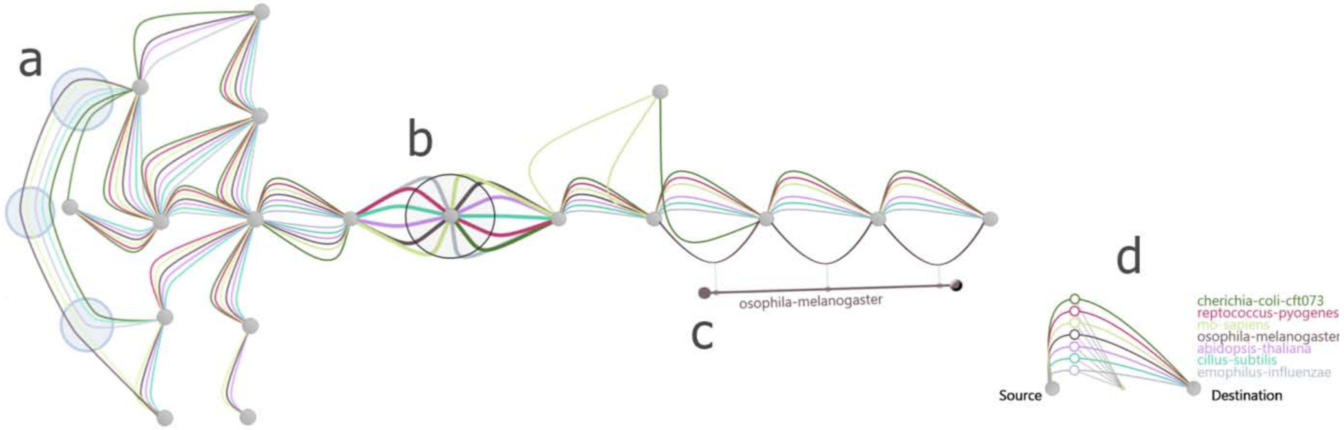

Figure 1. Integrating four interactive techniques to control link curvature in a biological network: (a) interactive bundling, (b) interactive link fanning (c) link magnets and (d) interactive legends.

ABSTRACT

When exploiting the power of node-link diagrams to represent real-world data such as web structures, airline routes, electrical, telecommunication and social networks, link congestion frequently arises. Such areas in the diagram—with dense, overlapping links—are not readable: connectivity, node shapes, labels, and contextual information are obscured. In response, graph-layout research has begun to consider the modification of link shapes with techniques such as link routing and bundling. In this paper, we delve into the interactive techniques afforded by variant use of link curvature, delineating a six-dimensional design space that is populated by four families of interactive techniques: bundling, fanning, magnets, and legends. Our taxonomy encompasses existing techniques and reveals several novel link interactions. We describe the implementation of these techniques and illustrate their potential for exploring dense graphs with multiple types of links.

Categories and Subject Descriptors

H.5.2 [Information Systems]: User Interfaces – misc.

General Terms

Design

Keywords

Link curvature, edge bundling, graph visualization, interaction techniques, design space, taxonomy.

AVI'12, May 21-25, 2012, Capri Island, Italy.
Copyright 2012 ACM 978-1-4503-1287-5/12-05…$10.00.

1. INTRODUCTION

With entities represented as nodes and relationships as links, examples of node-link diagram use span social networks, the World Wide Web, genealogy trees, and gene ontologies. In spite of their representational power and their broad applicability, node-link diagrams pose well recognized problems in the layout of their nodes and the organization of their links. These problems become increasingly difficult as the diagrams become denser and need to support multiple types of links. Such categorical link data is easy to come by: e.g., transportation networks with different links handled by different companies or metabolic networks with reactions from different organisms superimposed (§5).

While real-world graph (network) visualization offers myriad examples where link congestion completely obscures graph readability, there are still relatively few link-centric methods. Traditional graph-layout techniques have tended to consider curved routing as a way to minimize crossings or maximize angular resolution (e.g., layered layout, topology-shape metrics, shortest-path routing [14]). This may be due to an historical focus by researchers on visualization of smaller, sparser graphs. More recent work, however, has begun to address readability of densely connected graphs either through aggregated links (e.g., "bundling" [9]) or through various types of interaction.

In this paper, we explore the design space for interactive link exploration techniques focusing on curvature and we create a six-dimensional taxonomy that encompasses existing and novel techniques (§3). Using this taxonomy, we identify four families of interactive techniques (§4): interactive bundling, fanning, magnets and interactive legends. For each of these families, we expand the scope within them, positioning existing techniques and proposing new ones. Figure 1 illustrates an application of these four techniques to the visualization of a heterogeneous biological network. We present a few more applications (§5) before concluding on future work.

2. RELATED WORK

The graph-visualization community has made great progress in finding effective layouts for relatively sparse graphs (e.g., overview in [14]). However, as researchers focus their efforts on denser graphs, congestion amongst links becomes a major issue in producing readable node-link diagrams. Over the past five years there has been a shift in research attention from computing ideal node placements to drawing graphs with more readable link organization.

2.1 Algorithms for Computing Curved Links

To improve the readability of node-link diagrams, there have been two primary purposes for layout algorithms that generate curved links: reducing the number of link crossings and routing links to avoid overlapping with nodes.

In visualizations of small graphs, link crossings have been shown to be an important factor in readability [18][19]. For example, when two links cross with acute angle, it becomes difficult to disambiguate their individual trajectories. By using smooth curves that cross each other at more oblique angles and that differ in their curvature, their trajectories are more salient and become easier to follow [25]. An alternative way to assign link curvature is as a way to convey properties such as directionality [11]. This shows the utility of curvature to convey meaning, but does little for the congestion problem when many links are involved.

When reading dense graphs, links that overlap nodes to which they are not connected make it difficult to identify the actual sources and destinations of links. To tackle this problem, routing algorithms have been developed (e.g., [4]) to compute smooth, curved links that avoid a set of obstacles (i.e., nodes). This resolves the ambiguity caused by intersections between nodes and links, but increases the occurrence of shared link paths, and hence, makes path following more difficult. More recent work on "bundling" link curves together (§2.3) further aids the visibility of individual nodes and reduces the impression of clutter in the overall diagram but makes collinear links even more prevalent.

Recent work has attempted to avoid such collinear links while retaining the benefits of bundling by careful separation of links within bundles [16][20]. Another possibility is to allow people to resolve such ambiguity through interactive exploration of links and link bundles (§2.2).

2.2 Interactive Techniques

Several interactive techniques have been developed to assist people in disambiguating and following links in a region of focus on the diagram. For example, transient filtering techniques address the problem by displacing link curves from a locality using interactive lenses (EdgeLens) [12][28]. Contrary to standard filtering or aggregation techniques, the links remain in the view but their curvature is temporarily modified to avoid occlusion in a focal region. EdgePlucking [27] also temporarily modifies the curvature of links by allowing people to pull one or more curves out of the way through direct interaction.

In addition to EdgePlucking, other novel techniques have been applied to multi-touch surfaces [22]. They allow people to modify link curvature, to pluck individual links, or to group them in bundles through direct manipulation using multi-touch gestures. One of the techniques (called TouchStrumming) makes use of motion, as introduced in [2], to provide a visual cue to help people identify a specific link and to indicate source and destination nodes.

These interactive techniques temporarily improve the readability of dense regions of a graph. They complement algorithms that provide static layouts and can increase the performance of low-level tasks. However, since these techniques are locally applied to a specific region, higher-level tasks such as finding connectivity patterns may remain difficult.

2.3 Link Bundling

A recent line of research investigates link "bundling" [10] where the curvature of close links is modified to aggregate them into bundles. The bundles are reminiscent of "highways" on a road-map, and thus, can help to reveal gross connectivity. At the same time, however, detail of individual link paths is sacrificed.

Over the past five years, a wide variety of techniques have been investigated to compute such bundling. The techniques differ algorithmically (e.g., use of visibility graphs [15] and steadily improving run time [5][8]), in the way links are assigned to bundles (e.g., purely spatial [3], based on data-hierarchy [9], or topology [16]) and by the type of curvature used [7][20]. To date, we do not have a clear metric to assess the quality of the resulting patterns.

More recently, an effort to adapt the use of curvature to show directionality in large dense graphs has incorporated bundling. Using a divided-bundling technique (i.e., one bundle per direction), Selassie *et al.* showed the asymmetry in air traffic patterns by computing eastbound and westbound bundles of flights over the continental US [23]. This demonstrates the power of link bundling to highlight connectivity patterns in data.

We extend this direction, harnessing this power to reveal more patterns in the data by providing interactive techniques that enable custom link bundling and custom modification of link curvatures on the fly. In particular, in §5 we reflect on the use of curvature for exploring heterogeneous graphs (i.e., graphs with multiple types of relations). These graphs raise a wide range of analysis questions that remain largely unaddressed such as identifying groups of nodes with similar sets of relations or relations with similar patterns. The goal of this paper is to advance the techniques needed for this type of analysis through a principled exploration of the design space.

3. Design Space

While initial research into use of link curvature [11][23] seems promising and techniques such as bundling have experienced rapid adoption, the full potential of the space of interactive link curvature is still untapped. To harness the full power of interactive link curvature, we work towards a better understanding of this design space. In this section, we delineate a set of dimensions to provide a vocabulary that can distinguish between different types of interactive link curvature. We propose six design dimensions: intention, link subset, spatial and temporal scopes, interaction distance, and interaction freedom.

These dimensions (or axes) specify key decision points when designing interactive tools for manipulating link curvature in graph visualization, editing and exploration tasks. Characterizing existing techniques along these axes (Figure 2) reveals possible variations and suggests extensions. By exploring clusters of paths across these axes we identified four general families of interactive link curvature manipulation techniques which we further refine in §4, grouping them into distinct yet complementary techniques that make use of link curvature manipulation.

In the following, we present a definition of each dimension, briefly discuss the categories along each dimension and illustrate them with an example technique.

3.1 Intention

By *intention* we refer to the goal of the interaction. Link curvature may be applied with the intention of *grouping* links together to reduce clutter and exhibit connectivity trends (as in bundling). Conversely, curvature may help by *distinguishing* links from one another, helping people identify the source and destination nodes of a specific link (e.g., EdgeLens), or it may provide emphasis for certain types of links. These two extremes can be combined in different ways. For example, divided bundling [23] groups links in two distinct bundles based on their directionality; while ordered bundling [20]can allow distinguishing each link within a bundle.

The intention of curve manipulation tools in vector graphics authoring software (e.g., [1][13]) is neither to group nor to distinguish links (*other*). These tools aim to provide full control over the link curvature without presuming any specific intention from the user but offer fine control for editing a curve. In contrast, graph visualization systems must support people in uncovering connectivity patterns. In this context, fine control and the interaction burden associated with it takes a back seat. The priority of information visualization systems is generally to offer people a range of goal-specific interaction techniques with maximum interaction efficiency.

3.2 Link Subset

An interactive technique for manipulating link curvature may apply to more than one link. We can differentiate techniques based on the subset of links from a larger diagram to which they are typically applied. For example, interaction might occur on a specific part of a link according to *geometry* (e.g., Illustrator [1]) on links emerging from a node (e.g., LinkSliding [17]), to links that have a specific property based on *topology*, to links of a given data attribute (e.g., directionality [11]) with a *data-driven* selection, or on an arbitrary *user-defined* selection (e.g., EdgePlucking [27]).

3.3 Spatial Scope

Spatial scope refers to the region affected by the interaction technique. For example, changing the curvature of a given link in the diagram may affect links within a specific radius if the spatial scope of the interaction is *local* (e.g., as in EdgeLens). The technique can affect the curvature of links in the entire diagram if the scope is *global*, as is typically the case in bundling methods. Note that we consider interactions that affect all links of a certain data type as spatially *global*.

An interaction technique may offer control of the boundary of this spatial scope, smoothly covering the whole spectrum (from a single link, to a subset of links, to all links in the diagram). In this context, this dimension becomes a continuum rather than two discrete categories. However, we preserve the two categories—local and global—as they indicate the design rationale behind the technique. For example, while EdgeLens may use lenses of any radii, even encompassing the entire diagram, the design rationale remains the disambiguation of a *local* area.

3.4 Temporal Scope

Temporal scope captures the duration of the effect of interaction. For example, when bending a link to see what lies behind it, the interaction on link curvature is *transient*. Bending all links of a specific type to provide emphasis can be a *permanent* interaction. Note that we do not imply a *permanent* interaction is necessarily irrevocable. Rather, this term refers to the reach of a more stable state in contrast to a transient one.

In existing techniques, we observed that spatial scope and temporal scope are often related. For example, local interaction (such as EdgeLens) is likely to be intended as a transient change while exploring a specific region of the graph.

3.5 Interaction Distance

This dimension captures the distance between the point of interaction (mouse or touch-contact point) and the effect on the target links. The interaction may occur directly on the link (*contact*), close to it (*adjacent*), or be external to the representation (*external*). For example, adjusting a cardinal spline is done through a direct *contact* with the points of the spline, while manipulation may also occur on *adjacent* control points in the case of a B-spline. Controlling curvature through a global setting (e.g., a slider controlling the degree of bundling) is considered *external* to the representation.

3.6 Interaction Degree

Interaction techniques may have a number of parameters for people to control. The curve manipulation tools in general vector drawing software have *infinite* interaction freedom in that they allow people to control potentially all points on a curve. By contrast, in the EdgePlucking method people manipulate *one* point on a link to modify its curvature to distinguish a particular link from others nearby.

There is an interesting trade-off between the freedom offered to control the curvature (i.e., the number of parameters to adjust link curvature) and the interaction burden put on a person to achieve this control. This dimension is tightly linked to the intention of the technique defined earlier. Interaction techniques solving a specific readability goal, such as EdgePlucking, offer maximal interaction efficiency (low interaction degree).

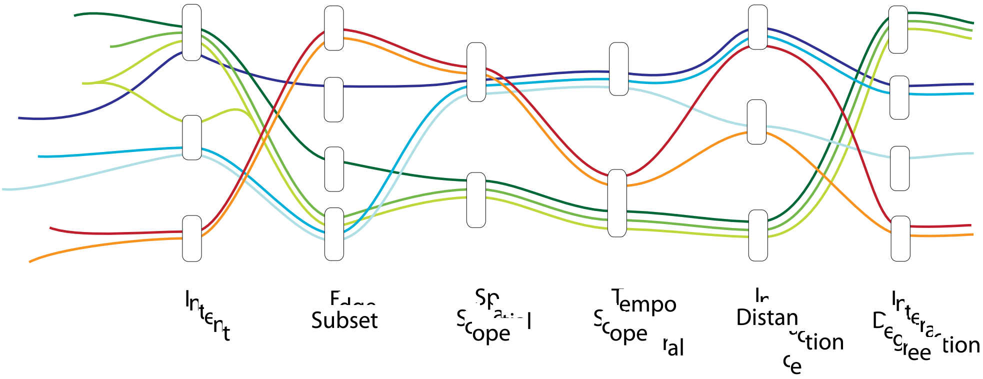

Figure 2. Our six-dimensional taxonomy used to differentiate existing techniques.

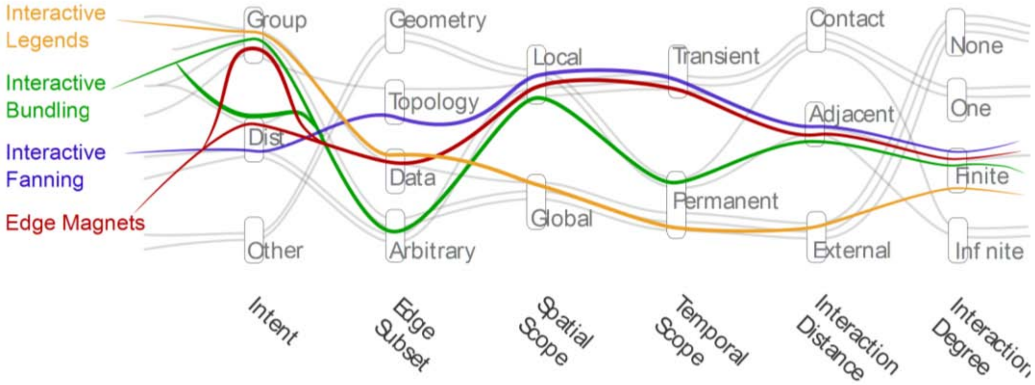

Figure 3. Patterns of our four families of interactive techniques.

4. FOUR FAMILIES OF TECHNIQUES

To illustrate these design dimensions, we present four families of link curvature interactions (Figure 3), each of which offers new possibilities for disambiguating link structure. These families build from existing techniques, extending their scope to consider the challenges of multiple types of links. For each family, we include example interaction affordances and a sketch of a potential implementation, the prototypes for which can be viewed in our accompanying video.

4.1 Interactive Bundling

Interactive Bundling adds manual bundle creation, direct editing, and manipulation of bundle attenuation points to the basic idea behind a recent automated bundling method, Bundled Routing [20], that generates neatly offset, crossing-minimal bundles that are routed to avoid node shapes. The automatic method for creating and routing the bundles provided no facility for editing or otherwise manipulating the resulting diagram.

In interactive bundling the intention includes both *grouping* links and *distinguishing* between them. The fact that a person can create a bundle manually makes *arbitrary* groupings of link subsets possible. The spatial scope is *local* since the bundling is only applied to particular regions of the diagram. The temporal scope tends towards *permanent* as people apply their semantic knowledge of the dataset to unclutter or group links. For example, they might select a group of related links and reroute them in a bundle through a less cluttered part of the diagram. Interaction distance is *adjacent* since it is the circles surrounding the bundles that are manipulated. Interaction degree is *finite* but often quite large, since many bundle control circles may be created, but too many may become unmanageable.

4.1.1 Interaction

There are two parts to the interaction offered by interactive bundling: (1) the creation and editing of the bundle attenuation circles; and (2) positioning and resizing them. In our prototype the attenuation points are created by drawing a line through the link curves to be bundled and the resulting bundle circle is centered at the middle of this line. Two-handed multi-touch affords some compelling editing gestures. For example, after creating the initial bundle circle by drawing a line through one or more links, the circle can be selected with one index-finger and the second index-finger can be used to drag links onto, or off, the circle.

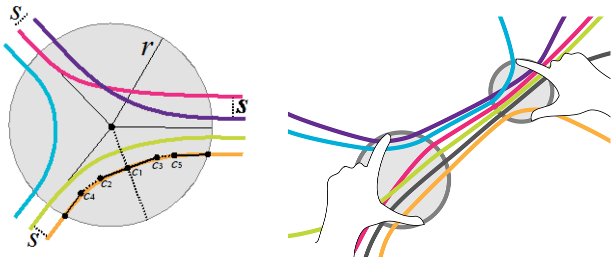

Figure 4. (a) Constructution of a bundling control point, (b) example multi-touch gesture to control link bundles.

Moving and resizing the circles is simple. In our mouse-based prototype, clicking anywhere in the interior of the circle initiates a drag. Clicking on the circumference allows resizing. With multi-touch both movement and resizing can be done simultaneously. Figure 4b shows a potential bi-modal touch interaction gesture for positioning and resizing multiple circles at once. Hence, the orientation, thickness at both ends and length of an entire bundle segment can be manipulated in a single gesture.

4.1.2 Implementation

Figure 4a shows the construction of the bundling attenuation circles. As in the ordered-bundling scheme, the order of links entering and leaving the attenuation points is chosen to minimize crossings [20]. Remaining crossings (e.g., between the pink and purple links in Figure 4a) occur inside the attenuation circles—rather than along the spans between them—in order to maximize the angle of intersection. The radius r of the circle and the minimum separation s between adjacent links are manually controlled. The spline control points for a particular link c_i are computed automatically: e.g., c_1 is on the bisector of the center lines of entry and exit bundles for the link. The points c_1, c_2 and c_3 are collinear. The entry and exit tangents (c_4, c_5) are parallel with the center lines of the entry and exit bundles respectively.

4.1.3 Extensions

As mentioned previously, large numbers of control points could become unwieldy if they all need to be sized and positioned manually. An automatic method provides a useful starting bundling for the whole diagram. However, a good compromise for interactive bundling would probably be a semi-automatic method for locating subsets of dense links, bundling them by type and suggesting optimal routings for the resulting bundles.

4.2 Interactive Link Fanning

The technique proposed in this section, Interactive Link Fanning, provides an interaction that can recover the local detail of any particular node. For example, for a chosen node, or possibly a small set of nodes, the angles between individual connecting links can be maximized. Creating space between links in this way also allows for the possibility of clearly showing labels or link decorators (e.g., arrowheads) for the individual links. Providing local node clarity can be important in any form of graph layout

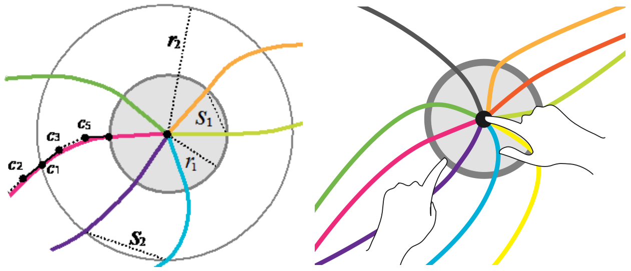

Figure 5. (a) Construction of link fanning, (b) example of multi-touch gesture to control link fanning.

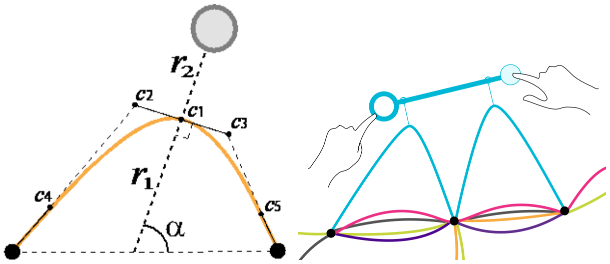

Figure 6. (a) Construction of a curve attracted by a magnet, (b) example of multi-touch gesture to control a line-magnet.

and link routing that considers the graph as a whole must compromise the readability of connectivity to a single node. For example, bundling reduces the clutter of the overall drawing by coalescing sets of nearby links. In doing so, it must reduce the angular resolution of links entering individual nodes. Link Fanning can counteract this.

The intention of this technique is to *distinguish* between links based on their *topology* (connectivity to a particular node or nodes) and the interaction scope is *local.* Closer inspection of a particular node is most likely *transient*. Interaction distance is *adjacent* since the interaction is with the connected node rather than with the links themselves.

4.2.1 Interaction

To trigger Link Fanning, simply touch a node or nodes to automatically spread the links connected to those nodes. For an individual node, a two finger gesture can be used to select the node for link spreading and simultaneously control the radius of the spreading circle (Figure 5b). This is easiest to achieve using bi-modal interaction on a large screen: e.g., index finger of the dominant hand to select the node and the index finger of the other hand to control the spreading radius.

4.2.2 Implementation

For optimal readability there are two requirements for such a technique:

1) Crossings between the links attached to the selected nodes should be minimized. If only a single node is selected then the link curves can be drawn in a planar way.
2) The link fan should be oriented such that the link curvature and length is minimized. That is, the link curves should follow the most direct route possible to their target nodes while maximizing separation between adjacent curves.

To this end, the links are sorted according to the compass direction from the selected, or *source*, node to each link's *target* node. Then, we find a first control point for each link by spacing all the link points equally around the circumference of the spreading circle. Then, we orient the circle such that the control points are as close as possible to each link's target. We can determine the orientation for the fan that minimizes the total squared distance between control points and targets by Procrustes analysis [26].

This basic technique maximizes the angle between adjacent links. However, it may be preferable to achieve some minimum separation (or angle) between links rather than try to spread them out equally. For example, if only a small number (e.g., half dozen) of links enter a node from one side in the original embedding, we may want only to provide enough space between links to insert labels while keeping the links to the original side of the node. In other words, we do not want to create unnecessary curvature. Figure 5a, shows a fanning achieving such a minimum separation S_1. In this latter case we can use an iterative technique to repeatedly add links that are in violation of the minimum angle constraint to the circle and optimize the orientation of the circle. Allowing only addition of links to the set provides a rapid heuristic for solving this problem. It can also be solved optimality by allowing removal of links when they are found to curve apart after orientation.

4.2.3 Extensions

A planar embedding for all links attached to a given node can be found by repeatedly applying the above technique to circles of increasing radii to find additional control points, until the radius of the circle exceeds the distance of the farthest target from the source. Figure 5a shows an outer circle oriented in this way to achieve a secondary minimum separation constraint S_2. Link fanning for all links between a set of nodes can be found by cycling application of the above algorithm to each source.

4.3 Link Magnets

We take the concept of interactive magnets [29][24] and apply it to links that have data attributes, creating Link Magnets. Link magnets are positioned at will to attract or repulse links of a given type within a given radius. This concept inverts techniques such as EdgeLens in deforming the curvature of links in a given region of a graph. Interactive magnets were initially introduced to provide interaction with multidimensional data [29]. When a magnet—a visual representation of a data attribute—is dragged, data items are displaced toward the magnet by an amount depending on the data attribute value of the item. MagnetViz [24] used this concept to attract nodes based on their data attributes and, with Link Magnets, we extend this interaction to links.

The *intention* of this method could be either to *distinguish* between different types of links if link-type is specified by the data or to *group* links of similar type. Thus the selected link subset would be based on *data*. The *spatial scope* is *local*: within some radius of the magnet control, and hence the *interaction distance* is *adjacent*. The interaction degree is finite; possibly affecting a large but not infinite set of edges.

4.3.1 Implementation

Link magnets are associated with a type of link. People can control the strength of the attraction. Figure 6a presents the effect of a magnet on a simple Bezier spline. We can specify the area affected by a link magnet in several ways. For example, the strength of the effect may be a function of the distance to the magnet. While this mimics the properties of physical magnets, it introduces subtleties in the way curvature is affected that may be difficult to perceive. Another way is to set finite boundaries to the force field, similar to lens techniques. While this gives more

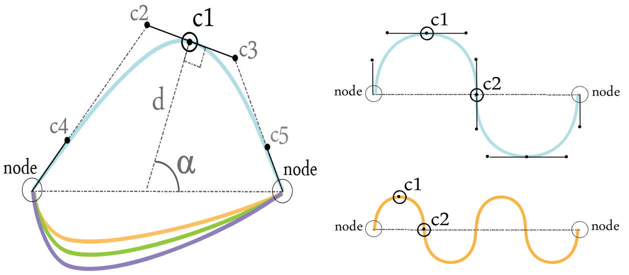

Figure 7: (a) model curve using a simple Bezier spline, (b) model curve based on the repetition of a motif.

freedom to people using the technique, it could burden them with additional parameters to control.

4.3.2 Extensions

Magnets have generally been represented by an object with a clear center point (e.g., circle or rectangle) with a circular range of influence. We can extend the concept to other shapes. For example, line magnets can reveal interesting patterns for linear and layered node layouts (Figure 1c). More complex shapes, for example polygonal magnets, may be constructed to give very fine control of multiple link curves.

4.4 Interactive Link Legends

With Interactive Link Legends, we investigate the use of link curvature to encode semantic information such as different types of links in heterogeneous graphs. This technique extends the use of curvature to encode directionality in links [6] by integrating the concept of interactive legends [21]. The shape for each type of link is represented by a model curve shown in a legend next to the diagram. This legend presents a summary of the visual encoding and allows people to modify encodings of the entire visualization by interacting with representative models.

The goal of an interactive model curve is to support the creation of a set of smooth and aesthetically appealing curves, easily differentiable in the diagram. There is trade-off between the expressiveness of the curve (e.g., the number of parameters to control its shape) and the burden put on people to generate distinct curves. Each link in the diagram has a different position of its source and destination nodes. Thus, we need to also consider how to adapt the model curve to each of the links (e.g., by scaling and rotating them appropriately).

With the intention of *grouping* edges of the same type, Link Legends are an example of *external interaction distance* with a *global spatial scope*. The most obvious use-case is when the dataset has different types of links; hence we give the *link subset* as *data*.

4.4.1 Implementations

Figure 7a shows a model using a Bezier spline. When people control the amplitude (d) and skewness (α) of the curve by interacting with a single point ($c1$) on the curve, the curvature is adjusted on each link instance by scaling and rotating it appropriately. This solution preserves the overall shapes of links but may produce strong variations in amplitude within links of a given type, depending on the distance between source and destination nodes.

Figure 7b shows a different model of curve based on the repetition of a motif. When people control the amplitude and periodicity of a sinusoidal curve by interacting with two points ($c1$, $c2$) on the curve, the shape is adjusted on each link instance by repeating it appropriately. This solution preserves the amplitude and overall shape of links independent of the distance between their source and destination nodes.

4.4.2 Extensions

An interesting extension is to encode the combination of links between two nodes into a single curve instead of representing each type of link independently. This technique parallels the color encoding of multiple types of links through color stripes on a single link. Combining multiple curve models into a single aggregate can be achieved in multiple ways. For example, one can sum them following a Fourier series principle. More sophisticated methods include fractal-type aggregation in which a curve is used as a base path to draw the next curve. In this case, the order of the aggregation produces different visual output. The evident limitation for these techniques is the number of aggregated curves that remain independently identifiable. Therefore, this technique is probably not suitable for datasets that exhibit many types of relations between two nodes.

5. Applications

We illustrate how interactive link curvature techniques support the analysis of heterogeneous networks with three different datasets.

5.1 Biological Network

Figure 1 presents a metabolic network. A metabolic network maps the biochemical reactions for a particular process inside a cell. The particular network for a given process may vary slightly across different organisms. Studying these differences can be thought of as a problem of network comparison. Links are directed and indicate a reaction occurring from a substrate (source node) to a product (destination node). This particular network shows the reactions for seven different species, each encoded by a different color in the diagram.

We used interactive legends (Figure 1d) to associate a representative curvature for each species. Without curvature, links between pairs of node are superimposed and species are not differentiable. We encoded directionality by the skewness of the curve and chose an amplitude that makes all species visible. This encoding reveals two general flows of reactions: left-to-right and bottom-to-top (for the left part of the diagram). It also highlights that a single pair of nodes has bi-directional reactions. This may raise interesting biological questions regarding the dual substrate/product role of these two nodes.

Figure 1 also showcases the other three techniques: interactive bundling (Figure 1a) improves the diagram readability by removing link crossings; interactive fanning (Figure 1b) showcases the symmetry of incoming and outgoing reactions for a particular node; and link magnets (Figure 1c) highlights reactions of a particular species in a specific area of the diagram.

5.2 Air-Traffic Network

Figure 8 presents an air-traffic network in which nodes represent the aggregated airports of a given state. Links show the flight traffic between states. This data shows flights to/from Washington State for a dozen airlines, each color-encoded.

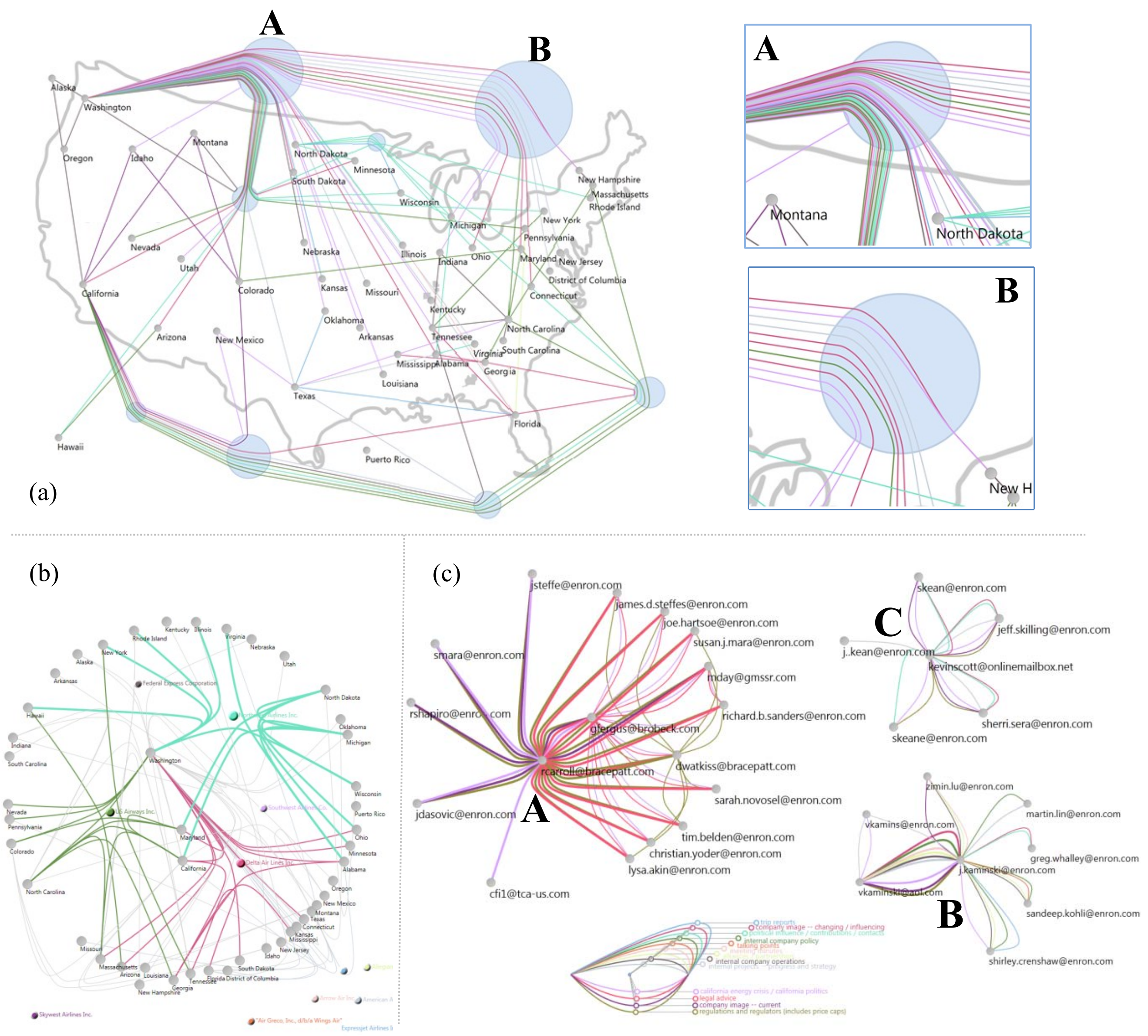

Figure 8. Three applications of link curvature on heterogeneous networks: (a) air traffic map, (b) air traffic with cicular layout, (c) social network with a force-directed layout.

Figure 8a adopts a geographic approach in which nodes are laid out according to their position in the US. Without any link curvature arrangement, this diagram suffers from occlusion as links overlap and cross each other. We used interactive bundling to reduce crossings and route most of the links outside the map to improve the readability of each node. We created two large bundles. The top one divides three high-traffic routes between (roughly) the states of the north-west, states of the mid-west and states of the east. Bundle control **A** indicates the division in three sub-bundles. Bundle control **B** forces the links to go horizontally, clarifying their crossing with others. Bundle control **B** is enlarged to show individual airlines. The lower bundle highlights that many flights to and from Washington go through California.

Figure 8b uses a different strategy as it aims to compare the coverage of three different airlines (and how they overlap). In this context, we gave less importance to geography and arranged nodes around a circle. We used edge magnets to attract the flights of each airline to a different point inside the circle. This Figure highlights two nodes covered by the three airlines (placed in the middle of the diagram). They also indicate the different coverage of states by each company and their overlap: turquoise has the highest number of dedicated states, whereas the other two have the highest number of overlapping states.

5.3 Social Network

Figure 8c presents a social network. Social networks represent relations between people. This example is a small subset of the Enron email data. Links represent an email exchange between two addresses. Email tagged in 13 general categories encoded here by color and curvature. This sample illustrates the email patterns of three different actors: the email communication of **A** reveals his dual-role (legal advice in shades of red and image advice in shades of purple) with two distinct groups of people. In the second connected component, the highlighted links reveal the strong connection between **B** and another email address. These happen to be the work and personal email address of the same person. Finally, **C** appears to have a more balanced email communication.

6. Discussion and Future Work

In this paper, we investigated the design space of link curvature in node-link diagrams. Many other visual encodings are available to improve link readability and convey data attributes. For example, in our figures we represented link type with both curvature and color. An empirical study to investigate the relative and combined merits of color and curvature may yield interesting insights.

We created a taxonomy to explore the design space of link curvature. It has proven useful in suggesting new ways to look at existing techniques, in discovering gaps in functionality, and in indicating new synthesis. Using this mapping of the space, we identified four families of techniques. For each, we proposed an implementation and suggested a few extensions. Many of the dimensions could be opened by making different choices and thus creating interactive link variations. In practice, we either chose the most likely use-case or assigned the technique to multiple categories. For example, magnets could be either a way to distinguish links of different types by drawing them away from each other, or for grouping links of the same type by drawing them together. We believe that further investigation of this design space may reveal more dimensions/categories and their combination may bring more techniques to light.

Finally, we prototyped a few techniques to showcase their potential. We applied them to heterogeneous networks from three different domains. We plan to pursue this work by developing more robust prototypes and implementing multi-touch gestures to control them. Such system will make it possible to study how people use these interactions for analyzing complex networks.

7. REFERENCES

[1] Adobe Illustrator CS5, www.adobe.com/Illustrator.

[2] Bartram, L. and Ware, C. 2002. Filtering and brushing with motion. *Proc. InfoVis '02*, 66–79.

[3] Cui, W., Zhou, H., Qu, H., Wong, P. C., and Li, X. 2008. Geometry-Based Edge Clustering for Graph Visualization. *IEEE Trans Vis Comput Graph*, 14, 6, 1277–1284.

[4] Dobkin, D. P., Gansner, E. R., Os, E. K., and North, S. C. 1997. Implementing a general-purpose edge router, *Proc. GD '97*, 262–271.

[5] Ersoy, O., Hurter, C., Paulovich, F. V., Cantareiro, G., and Telea, A. 2011. Skeleton-Based Edge Bundling for Graph Visualization, *IEEE TVCG (Proc. InfoVis '11)*, 17, 12, 2364–2373.

[6] Fekete, J., Wang, D., Dang, N., Aris, A., and Plaisant, C. 2003. Overlaying graph links on treemaps. *InfoVis '03 Conference Compendium (demo)*.

[7] Freivalds, K. 2001. Curved Edge Routing. *Proc. FCT '01*, 126–137.

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

[9] Holten, D. 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data, *IEEE TVCG (Proc. InfoVis '06)*, 12, 5, 741–748.

[10] Holten, D. and van Wijk, J. J. 2009. Force-Directed Edge Bundling for Graph Visualization, *Computer Graphics Forum (Proc. EuroVis '09)*, 28, 3, 983–990.

[11] Holten, D. and van Wijk, J. 2009. A User Study on Visualizing Directed Edges in Graphs, *Proc. CHI '09*, 759–766.

[12] Hurter, C., Ersoy, O., and Telea, A. 2011. MoleView: An Attribute and Structure-based Semantic Lens for Large Element-based Plots. *IEEE TVCG (Proc. InfoVis '11)*, 17, 12, 2600–2609.

[13] Inkscape. Draw Freely., http://inkscape.org.

[14] Kaufmann, M. and Wagner, D. (Eds) 2001. Drawing Graphs: Methods and Models, *LNCS 2025*.

[15] Lambert, A., Bourqui, R., and Auber, D. 2010. Winding Roads: Routing edges into bundles, *Computer Graphics Forum (Proc. EuroVis '10)*, 29, 3, 853–862.

[16] Luo, S.-J., Liu, C.-L., Chen, B.-Y., and Ma, K.-L. 2011 Ambiguity-Free Edge-Bundling for Interactive Graph Visualization, *IEEE TVCG*, in press.

[17] Moscovich, T., Chevalier, F., Henry, N., Pietriga, E., and Fekete, J.-D. 2009. Topology-aware navigation in large networks, *Proc. CHI '09*, 2319–2328.

[18] Purchase, H. C. 1997. Which aesthetic has the greatest effect on human understanding? *Proc. GD '97*, 248–261.

[19] Purchase, H. C., Carrington, D., and Allder, J. 2004. Evaluating graph drawing aesthetics: defining and exploring a new empirical research area. *Computer Graphics and Multimedia*, 145–178.

[20] Pupyrev, S., Nachmanson, L., Bereg, S., and Holroyd, A. E. 2011. Edge Routing with Ordered Bundles, *Proc. GD '11*, 135–146.

[21] Riche, N.H., Lee, B., and Plaisant, C. 2010. Understanding Interactive Legends: a Comparative Evaluation with Standard Widgets, *Computer Graphics Forum (Proc. EuroVis '10)*, 29, 3, 1193–1202.

[22] Schmidt, S., Nacenta, M. A., Dachselt, R., and Carpendale, S. 2010. A Set of Multi-touch Graph Interaction Techniques, *Proc. ITS '10*.

[23] Selassie, D., Heller, B., and Heer, J. 2011. Divided Edge Bundling for Directional Network Data, *IEEE TVCG (Proc. InfoVis '11)*, 17, 12, 2354–2463.

[24] Spritzer, A. and Freitas, C. M. 2011. Design and Evaluation of MagnetViz - A Graph Visualization Tool. *IEEE Trans Vis Comput Graph*.

[25] Ware, C. 2004. Information Visualization: Perception for Design, 2nd Ed.

[26] Wikipedia, Procrustes Analysis: http://en.wikipedia.org/wiki/Procrustes_analysis

[27] Wong, N. and Carpendale, S. 2007. Supporting interactive graph exploration with edge plucking. *Proc. VDA '07*.

[28] Wong, N. Carpendale, S., and Greenberg, S. 2003. Edgelens: An interactive method for managing edge congestion in graphs. *Proc. InfoVis '03*, 7–15.

[29] Yi, J. S., Melton, R., Stasko, J., and Jacko, J. A. 2005. Dust & magnet: multivariate information visualization using a magnet metaphor, *Information Visualization*, 4, 4.