VisLink: Revealing Relationships Amongst Visualizations

Christopher Collins and Sheelagh Carpendale

Abstract— We present VisLink, a method by which visualizations and the relationships between them can be interactively explored. VisLink readily generalizes to support multiple visualizations, empowers inter-representational queries, and enables the reuse of the spatial variables, thus supporting efficient information encoding and providing for powerful visualization bridging. Our approach uses multiple 2D layouts, drawing each one in its own plane. These planes can then be placed and re-positioned in 3D space: side by side, in parallel, or in chosen placements that provide favoured views. Relationships, connections, and patterns between visualizations can be revealed and explored using a variety of interaction techniques including spreading activation and search filters.

Index Terms—Graph visualization, node-link diagrams, structural comparison, hierarchies, 3D visualization, edge aggregation.

◆

1 INTRODUCTION

As information visualizations continue to play a more frequent role in information analysis, the complexity of the queries for which we would like visual explanations also continues to grow. While creating visualizations of multi-variate data is a familiar challenge, the visual portrayal of two sets of relationships, one primary and one secondary, within a given visualization is relatively new (*e.g.*, [6, 10, 17]). With VisLink, we extend this direction, making it possible to reveal relationships, patterns, and connections between two or more primary visualizations. VisLink enables reuse of the spatial visual variable, thus supporting efficient information encoding and providing for powerful visualization bridging which in turn allows inter-visualization queries. For example, consider a linguistic question such as whether the formal hierarchical structure as expressed through the `IS-A` relationships in WordNet [16] is reflected by actual semantic similarity from usage statistics. This is best answered by propagating relationships between two visualizations: one a hierarchical view of WordNet `IS-A` relationships and the other a node clustering graph of semantic similarity relationships. Patterns within the inter-visualization relationships will reveal the similarities and differences in the two views of lexical organization.

VisLink supports the display of multiple 2D visualizations, each with its own use of spatial organization and each placed on its own interactive plane. These planes can be positioned and re-positioned supporting inter-visualization comparisons; however, it is VisLink's capability for displaying inter-representational queries that is our main contribution. Propagating edges between visualizations can reveal patterns by taking advantage of the spatial structure of both visualizations. In this paper we will explain our new visualization technique in comparison to existing multi-relationship visualizations.

2 FORMALIZING VISUALIZATIONS OF MULTIPLE RELATIONS

VisLink extends existing approaches to visualizing multiple relationships by revealing relationships amongst visualizations while maintaining the 'spatial rights' of each individual relationship type. In order to discuss more precisely the distinctions between previous work and our contribution, we will first introduce some notation for describing multiple view visualizations.

Given a data set, D_A, and a set of relationships, R_A, on D_A, we will write this as $R_A(D_A)$. Note that with the relation R_A we are not referring to a strict mathematical function, but rather any relation upon a data set, for example, a type of edge among nodes in a general graph. A second set of relationships on the same data set would be $R_B(D_A)$, while the same set of relationships on a different but parallel data set would be $R_A(D_B)$. For example, if the data set D_A was housing information in Montreal, an example of R_A could be the specific house to property tax relation $R_A(D_A)$ and a different relationship R_B could be the house size as related to the distance from transit routes $R_B(D_A)$. Then an example $R_A(D_B)$ would be property tax on houses in Toronto. Creating a first visualization, Vis_A, of these relationships $R_A(D_A)$ we will write $Vis_A \rightarrow R_A(D_A)$ (for example, a geographic map with houses coloured based on their property tax). A second visualization, Vis_B, of the same set of relationships would be $Vis_B \rightarrow R_A(D_A)$ (for example, a histogram of number of houses in each property tax range).

In the remainder of this section, we use this notation to define, compare, and contrast each of the current approaches to relating visualizations. We will show how VisLink provides capability beyond what is currently available.

2.1 Individual Visualizations

As a viewer of any given set of visualizations it is possible to do the cognitive work of developing cross visualization comparisons. For instance, visualizations can be printed and one can, by hand with pen and pencil, create annotations and/or new visualizations to develop the comparisons needed for the current task. Any relations on any data may be compared manually in this way (see Figure 1A).

2.2 Coordinated Multiple Views

Coordinated views provide several usually juxtaposed or tiled views of visualizations that are designed to be of use in relationship to each other (*e.g.*, Snap-Together Visualization [18]). These can be of various flavours such as Vis_A, Vis_B and Vis_C of $R_A(D_A)$ or perhaps Vis_A of $R_A(D_A)$, $R_B(D_A)$ and $R_C(D_A)$. The important factor for this visualization comparison discussion is that these coordinated views can be algorithmically linked such that actions and highlights in one view can be reflected on other views. Coordinated views allow for reuse of the spatial visual variable, thus each relationship type is afforded spatial rights. The temporarily activated visual connections can be a great advantage over finding the related data items manually but the relationships themselves are not explicitly visualized (see Figure 1B).

2.3 Compound Graph Visualizations

There are now a few examples of compound graph visualizations, such as overlays on Treemaps [6], ArcTrees [17], and Hierarchical Edge Bundles [10]. Figure 1C shows a simple diagram of this. Compound graph visualizations are created as follows:

Given: Data set D_A, containing two (or more) types of relationship: $R_A(D_A), R_B(D_A), \ldots, R_N(D_A)$.

Problem: Show multiple relationship types on the same visualization.

- *Christopher Collins is a PhD Candidate with the Computer Science Department at the Univeristy of Toronto, E-mail: ccollins@cs.utoronto.ca.*
- *Sheelagh Carpendale is a Professor with the Computer Science Department at the University of Calgary, E-mail: sheelagh@cpsc.ucalgary.ca.*

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

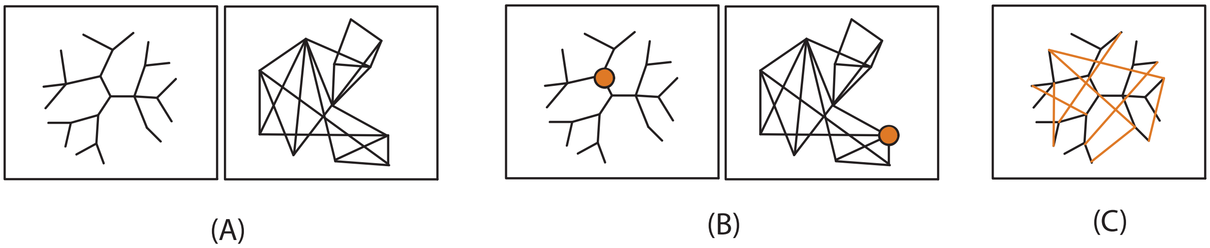


Fig. 1. Current approaches to comparing visualizations include (A) manual comparison (printed diagrams or separate programs), (B) coordinated multiple views (linked views with highlighting), and (C) compound graphs (layout based on one relationship, other relationships drawn upon it).

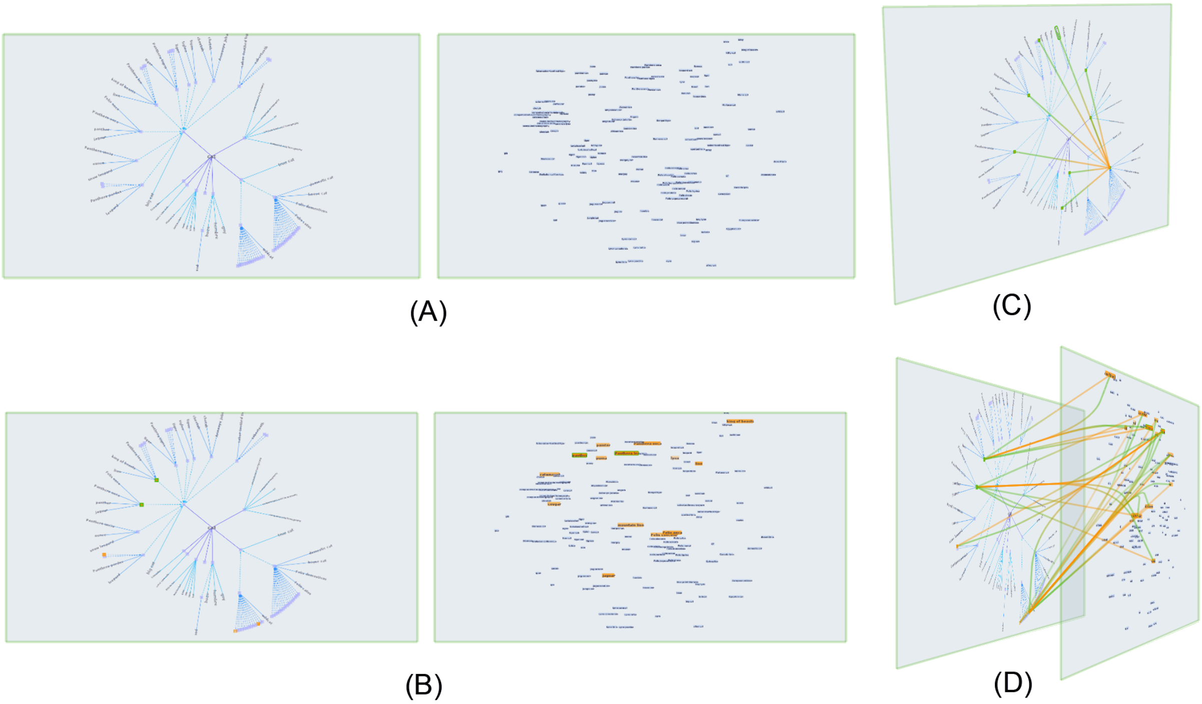


Fig. 2. VisLink encompasses existing multiple views techniques of (A) manual comparison, (B) coordinated multiple views, and (C) compound graphs. VisLink extends this continuum to direct linking of any number of multiple views (D).

Step 1: Choose a relationship type, *e.g.*, R_A, to be the primary relationship.

Step 2: Create a visualization $Vis_A \rightarrow R_A(D_A)$, providing an appropriate spatial layout. Since spatial organization is such a powerful factor in comprehending the given relationships, we refer to this as giving R_A 'spatial rights'.

Step 3: Create a visualization of $R_B(D_A)$ (and any other desired secondary relations) atop $Vis_A \rightarrow R_A(D_A)$.

This in effect creates $Vis_A \rightarrow R_A, R_B(D_A)$ using the spatial organization of $Vis_A \rightarrow R_A(D_A)$. While this is an exciting step forward in comparative visualization, note that $R_B(D_A)$ has no spatial rights of its own. That is, while viewing how the relationships in $R_B(D_A)$ relate to $R_A(D_A)$ is possible, there is no access to a visualization $Vis_B \rightarrow R_B(D_A)$. Hierarchical Edge Bundles [10] started an interesting exploration into using the spatial organization of $R_A(D_A)$ to affect the readability of the drawing of $R_B(D_A)$ atop $Vis_A \rightarrow R_A(D_A)$ and also indicated possibilities of addressing the readability needs of $R_B(D_A)$ by altering the spatial drawing of $Vis_A \rightarrow R_A(D_A)$ so that $R_B(D_A)$ and $R_A(D_A)$ occupy different spatial areas. This gives $R_B(D_A)$ partial spatial rights in that its presence affects the $Vis_A \rightarrow R_A(D_A)$ layout.

2.4 Semantic Substrates Visualizations

Shneiderman and Aris [20] introduce Semantic Substrates, a visualization that is both quite different and quite similar in concept to VisLink. We will use our notation to help specify this:

Given: Data set D_A and a set of primary relationships $R_A(D_A)$.

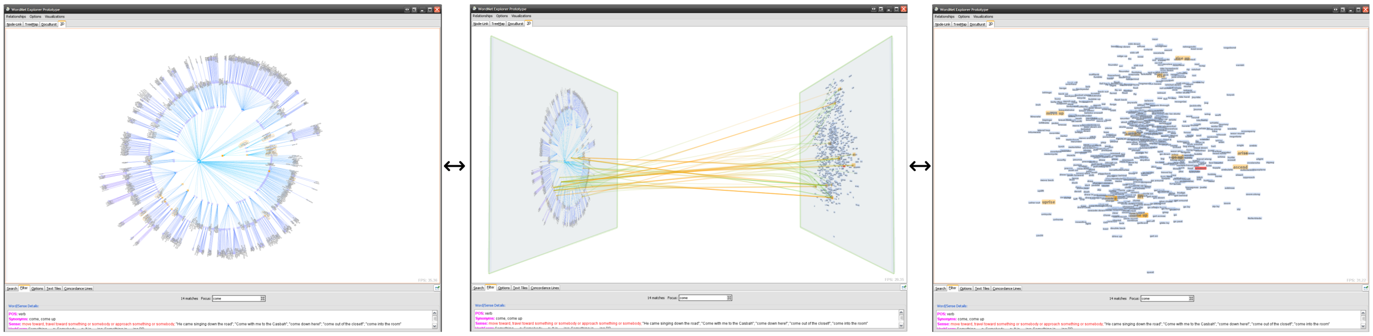
Problem: A given unified visualization creates too complex a graph for reasonable reading of the visualization.

Step 1: Partition the data set D_A into semantically interesting subsets, $D_{A_1}, D_{A_2}, \ldots, D_{A_n}$.

Step 2: Use the same visualization Vis_A, with spatial rights, to create visualizations of the subsets $Vis_A \rightarrow R_A(D_{A_1})$, $Vis_A \rightarrow R_A(D_{A_2})$, $\ldots$, $Vis_A \rightarrow R_A(D_{A_n})$.

Step 3: Juxtapose one or more of $Vis_A \rightarrow R_A(D_{A_1})$, $Vis_A \rightarrow R_A(D_{A_2}), \ldots, Vis_A \rightarrow R_A(D_{A_n})$, aligned in a plane.

Step 4: Draw edges of $R_A(D_A)$ across $Vis_A \rightarrow R_A(D_{A_1})$, $Vis_A \rightarrow R_A(D_{A_2}), \ldots, Vis_A \rightarrow R_A(D_{A_n})$ to create $Vis_A \rightarrow R_A(D_A)$.



14 matches Focus: come

POS: verb
Synonyms: come, come up
Sense: move toward, travel toward something or somebody or approach something or somebody; "He came singing down the road"; "Come with me to the Casbah"; "come down here!"; "come out of the closet!"; "come into the room"

(A) (B) (C)

Fig. 3. Viewing modes. (A) 2D equivalency view of plane one, showing hyponyms of verb ‘move’, with highlighted search results for ‘come’. (B) Search results on plane one activate inter-plane edges, visible in 3D mode. Nodes connected to search results are highlighted on plane two, a similarity clustering of words related to ‘move’. Propagated results are also visible when plane two is viewed in 2D equivalency mode (C).

2.5 VisLink Visualizations

Now we will use our notation to clarify the contribution of the VisLink visualization:

Given: Data set D_A and a set of primary relationships $R_A(D_A)$, $R_B(D_A), \ldots, R_N(D_A)$.

Problem: Provide a visualization that aids in improving the understanding of $R_A(D_A)$, $R_B(D_A)$, …, $R_N(D_A)$ by indicating how one set of relationships is related to the structure in another.

Step 1: Create visualizations $Vis_A \rightarrow R_A(D_A)$, $Vis_B \rightarrow R_B(D_A)$, …, $Vis_N \rightarrow R_N(D_A)$, each with full spatial rights for any of $R_A(D_A)$, $R_B(D_A), \ldots, R_N(D_A)$ that are of interest.

Step 2: Place selected visualizations $Vis_A \rightarrow R_A(D_A)$, $Vis_B \rightarrow R_B(D_A)$, …, $Vis_N \rightarrow R_N(D_A)$ on individual planes to support varying types of juxtaposition between visualizations (at this point we are limiting these to 2D representations).

Step 3: Draw edges of second order relations $T(R_A, R_B, \ldots, R_N(D_A))$, from $Vis_i \rightarrow R_i(D_A)$ to $Vis_{(i+1)} \rightarrow R_{(i+1)}(D_A)$ and $Vis_{(i-1)} \rightarrow R_{(i-1)}(D_A)$ to create VisLink inter-plane edges between neighbouring planes.

So, where Semantic Substrates operates with a single visualization type and single relation across multiple subsets of a data set, VisLink can operate on multiple visualization types and multiple relationship types on a single dataset. A natural extension of VisLink is to inferred or indirect relations across multiple data sets:

Given: Data sets D_A, D_B, …, D_N and the existence meaningful relationships, $T(D_i, D_j)$, among datasets such that (i, j) are any of A, $B, \ldots, N$.

Visualize: VisLink can be used with no further extensions to relate $Vis_A \rightarrow R_A(D_A)$, $Vis_B \rightarrow R_B(D_B)$, …, $Vis_N \rightarrow R_N(D_N)$, by using $T(D_i, D_j)$ to create inter-plane edges. An example of cross-dataset visualization is presented in Section 5.

We have presented a series of multi-relation visualizations, differing in the level of visual and algorithmic integration between relations and the amount of spatial rights accorded to secondary relations. VisLink can be used equivalently to any of the mentioned multi-relation visualization approaches (see Figure 2A–C) and extends the series to simultaneously provide equal spatial rights to all relations for which a visualization can be created, along with close visual and algorithmic integration of different relations (see Figure 2D).

3 VISLINK: COMPARISON WITH VISUALIZATION PLANES

In order to provide for a visualization space in which multiple data-related visualizations can be analyzed, we have developed VisLink. We start our explanation with a very brief description of the lexical data set and the lexical data relationships which are used to illustrate VisLink’s functionality and interactive capabilities. Next we show a sample set of 2D lexical visualizations displayed on visualization planes within VisLink, followed by the possible interactions with these visualization planes. Then the inter-visualization edges are explained and the ability to use inter-plane edge propagation to answer complex queries is presented.

3.1 Visualizations of Lexical Data

The example figures in this paper are drawn from application of VisLink to a lexical data set. This is an area of interest to computational linguists, and several visualizations using lexical data have been reported (*e.g.*, [5, 13]).

Using our formalism, we have a dataset D_A containing all the words in the English language. There are many types of relationships among words, for example, the lexical database WordNet [16] describes the hierarchical `IS-A` relation over synsets, which are sets of synonymous words. For example, {*lawyer, attorney*} `IS-A` {*occupation, job*}. The `IS-A` relation is also called hyponymy, so *chair* is a hyponym of *furniture*. We use hyponymy to build animated radial graphs [22], which serve as our $Vis_A \rightarrow R_A(D_A)$. Synsets are shown in the radial graph as small squares, and the synonymous words that make up the set are shown as attached, labelled, nodes. An example 2D radial hyponymy graph is in Figure 3A.

Words can also be related by their similarity. Similarity can be a surface feature, for example, orthographic (alphabetic) similarity, or it can be based on underlying semantics. We use a force-directed layout [1] to perform similarity clustering on words. In our examples we use orthographic similarity, so that all words are connected to all others by springs whose tension coefficient is inversely related to number of consecutive character matches in the substring, starting at the beginning. Words that start with the same letters will cluster together. This is a very different structure than hyponymy and serves as $Vis_B \rightarrow R_B(D_A)$. An example 2D alphabetic clustering visualization is in Figure 3C. We have also experimented with clustering using the semantic similarity measures implemented by Pedersen *et al.* [19], for example similarity as measured by lexical overlap in the dictionary definitions of words. However, those measures did not produce visible clusters and further investigation is needed into the appropriate relationship between the similarity measure and the spring coefficient.

Using VisLink, we investigate relations between the hyponymy layout of synsets and the orthographic clustering layout of words. With this, we can investigate questions such as: do some synsets contain high concentrations of orthographically similar words?

Data is loaded into the VisLink lexical visualization by looking up a synset in WordNet to root the hyponymy tree. The orthographic clustering is then populated with the relevant words from the dataset.

3.2 Navigation and Plane Interaction

VisLink is a 3D space within which any number of 2D semi-transparent visualization planes are positioned. These visualization planes act as virtual displays, upon which any data visualization can

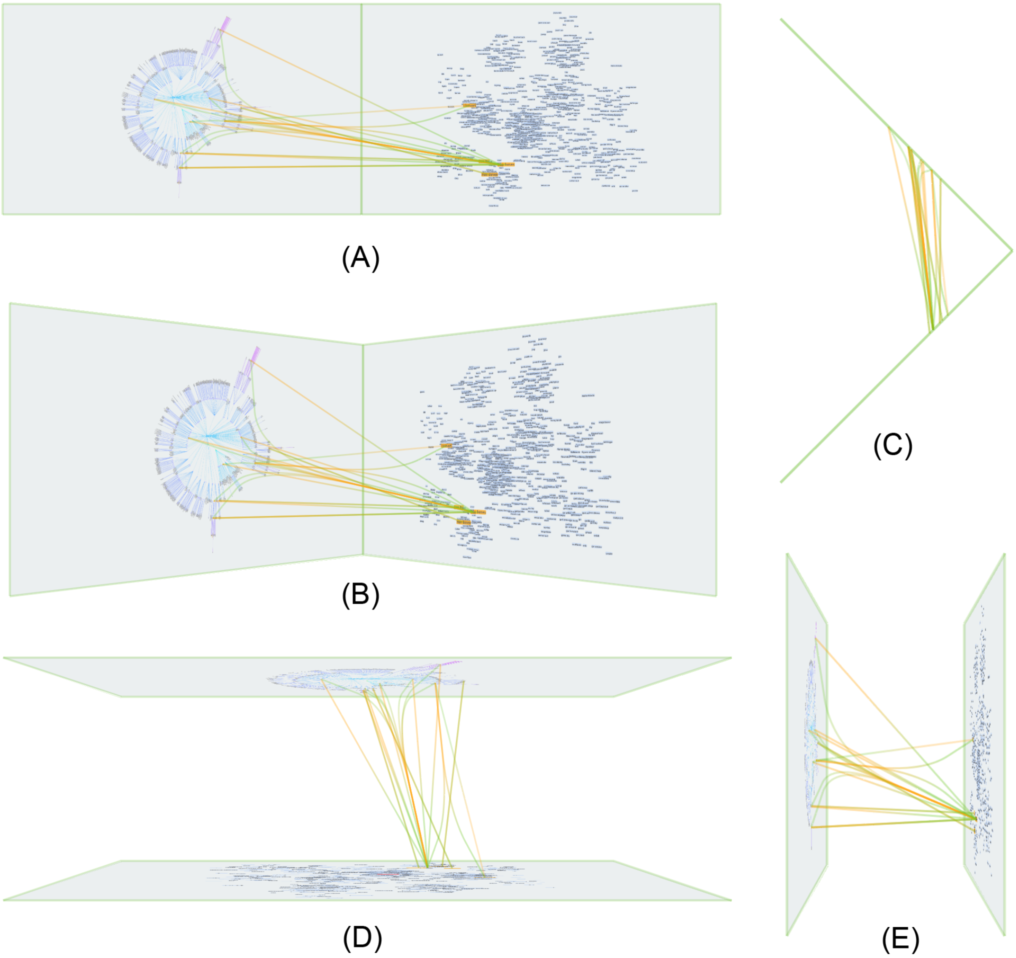

(A)
(B)
(C)
(D)
(E)

Fig. 4. Keyboard shortcuts provide for animated transition to default views, easing navigation in the 3D space. Views are (A) flat, (B) book, (C) book top, (D) top, and (E) side.

be drawn and manipulated. They can be rotated and shown side by side similar to multi-program or coordinated views, or rotated in opposition with included connections. Interaction and representation with each plane remains unchanged (representations do not relinquish any ‘spatial rights’ nor any ‘interaction rights’).

While VisLink is a 3D space, the visualization planes are 2D equivalents of a display, similar to windows in Miramar [14] or view-ports in the Web Forager [3]. We provide view animation shortcuts to transition between 2D and 3D views. Similar to interaction provided by Miramar, any visualization plane may be selected, activating an animated transition in which the selected plane flies forward and reorients to fill display space. When a plane is selected, 3D interaction widgets and inter-plane edges are deactivated, and the display becomes equivalent to 2D (see Figure 3). Because VisLink visualization planes have the same virtual dimensions as the on-screen view-port, transition between 2D plane view and 3D VisLink view does not require any resizing of the selected plane. When the plane is deselected, it falls back into the VisLink space, reverting to the original 3D view.

Interaction with the visualization on a visualization plane is always equivalent to 2D: mouse events are transformed to plane-relative coordinates and passed to the relevant visualization (irrespective of the current position and orientation of the plane). Visualizations can be manipulated directly in the 3D space (using equivalent-to-2D mode is not necessary). Thus interaction techniques developed for 2D visualizations become immediately available in VisLink. For example, we provide for a radial node-link view of the WordNet hyponymy (`IS-A`) relation, restricted with a generalized fish eye view to show only nodes of distance *N* or less from the central focus. The focus node can be reselected by a mouse click, activating radial layout animation [22]. Double clicking any node restricts the view to the tree rooted at that node, providing for drill-down capability. Drill down and other data reload interactions are propagated to all planes. Interaction techniques such as panning and zooming in 2D are provided by clicking and dragging on a visualization plane the same as one would on an equivalent stand-alone 2D visualization.

In addition to interaction with the visualizations on VisLink planes, we also provide for interaction with the planes themselves. While the usual capabilities for navigation in a 3D space (pan, zoom, rotate of camera position) are available in VisLink, in providing a 3D perspective projection virtual space, we must address the difficulties that arise from 6-degrees-of-freedom (DOF) control with 2-DOF input devices [2]. Free navigation can result in disorientation and non-optimal viewing positions, while free manipulation of 3D objects can result in difficulty achieving precise desired positioning.

Therefore, we also provide shortcuts for cinematic animated repositioning of the camera and planes to preset viewpoints [14]. These viewpoints allow visualization planes to be viewed from the front (planes parallel and side by side) (see Figure 4A), with relative plane orientation of book view (planes perpendicular and meet at an edge) (see Figure 4B), top (see Figure 4C and D), or in opposition (planes parallel and stacked) (see Figure 4D and E). By choosing one of these viewpoints, users can recover from any disorienting manipulation.

As a solution to 2D plane interaction in a 3D space, we follow McGuffin *et al.* [15] and provide for manipulation of visualization

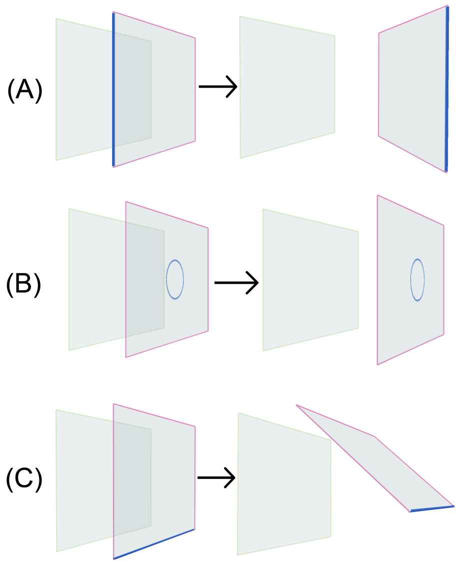


Fig. 5. Visualization planes are independently manipulated with three widgets: (A) side 'book pages' rotation, (B) center 'accordion' translation, and (C) bottom 'garage door' rotation.

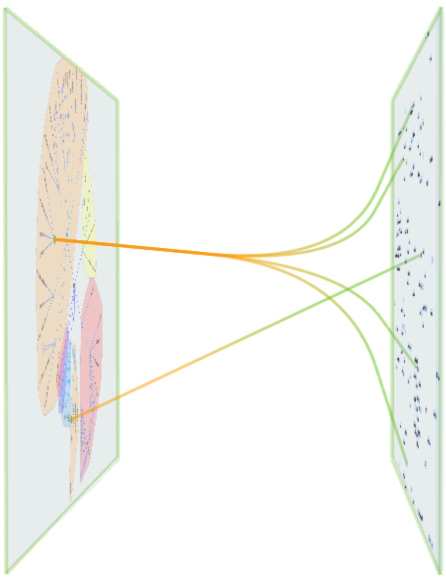


Fig. 6. VisLink inter-plane edge detail: one-to-one edges are straight, one-to-many edges are bundled. Alpha blending provides for stronger appearance of bundled edges.

plane position and orientation using a set of restricted movement widgets. Edge widgets provide for hinge movement (up to 90 degrees) about the opposite edge, and a center widget provides for translation, accordion style, along the axis between the planes (see Figure 5). Widgets become visible when the pointer is over their position, otherwise they are hidden from view to prevent data occlusion.

3.3 Adding Inter-Plane Edges

Edges are drawn in 3D to bridge adjacent visualization planes. Relationships between the visualizations can either be direct (nodes representing the same data are connected across planes) or indirect (items on different planes have relations defined within the data).

For example, in our lexical visualization, we examine the formal structure of WordNet hyponymy (the `IS-A` relation) on one plane, and the clustering of words based on their similarity on another. The inter-plane relationship in this case is direct: nodes on plane one represent the same data as nodes on plane two. In this case, it is the difference in the spatial organization of the layouts that is of interest. In essence, the pattern of inter-plane edges reveals a second-order relation: the relationship between different types of node relations on the same data. If the clustering by similarity approximates the formal structure, edges from synonyms in the structured data will go to the same cluster (*i.e.*, edges from synonyms will be parallel).

Indirect relations can also be visualized. For example, a visualization plane could be populated with a general graph about self-declared friendships in a social networking system. A second visualization plane could be populated with a tag cloud from a folksonomy, for example a bookmark sharing database. A third visualization plane could be populated with a visualization of the hypertext links between bookmarked pages. The three types of indirect inter-plane connections could be derived from three cross-dataset rules: `PERSON used TAG`, `PAGE tagged with TAG`, and `PERSON bookmarked PAGE`. With effective inter-plane edge management and data filtering, patterns between planes in such a visualization could reveal people who share tagging habits, or bookmarked pages with similar tag sets.

All inter-plane edges are specified with a single source node on plane i and one or more target nodes on plane j. Single source to single target edges are drawn as straight lines. Single source to many target edges are drawn using multiple curves calculated with corner-cutting [4]. For each curve from the source to a target, the starting control point is set as the source node, a middle control point is set as the average (world coordinates) position of all target nodes and the source, and the end point is set as the target. Five iterations of corner-cutting provide for smooth curves which start along the same straight line and then diverge as they approach their targets. By using alpha blending, the more semi-transparent curves that are coincident, the stronger the bundled edges appear (see Figure 6). Inter-plane edge positions are recalculated as appropriate so that edges remain fluidly attached to their source and target nodes throughout all manipulations of the constituent visualizations, plane positions, and the 3D viewpoint.

For visual clarity, edges are drawn between items on adjacent planes only. For more than two visualization planes, if the data contains relations among all visualizations, these relations can be explored by reordering the visualization planes using the center translation (accordion) widget to move planes along the inter-plane axis. As a plane passes through another, the rendering is updated to show the relations between the new neighbours. Similar to axis ordering in parallel coordinates plots [11], the ordering of visualization planes strongly effects the visibility of interesting patterns in the data. Investigation into methods for choosing plane orderings is left for future research.

3.4 Using Inter-Plane Edges

Inter-plane edges can be revealed either on a per-plane basis (see Figure 7) or a per-node basis (see Figure 8). Activating an entire plane can reveal structural patterns that may exist between the visualizations, while individual node activation provides for detailed views of particular relations.

We provide for spreading node activation between planes, which adds additional analytic power to VisLink. When a node is manually activated on one plane, it is highlighted in orange with a green border and all inter-plane edges originating at that node are revealed. The target nodes for those edges are then activated. Edges originating at these nodes are then drawn and the activation is propagated iteratively up to a user-selected number of 'reflections' between planes. Deactivation of a node reverses the process, spreading the deactivation and hiding edges. The level of activation exponentially decays with each iteration.

Nodes are assigned activation values from 0 (deactivated) to 1 (manually activated by user through selection, search, or plane acti-

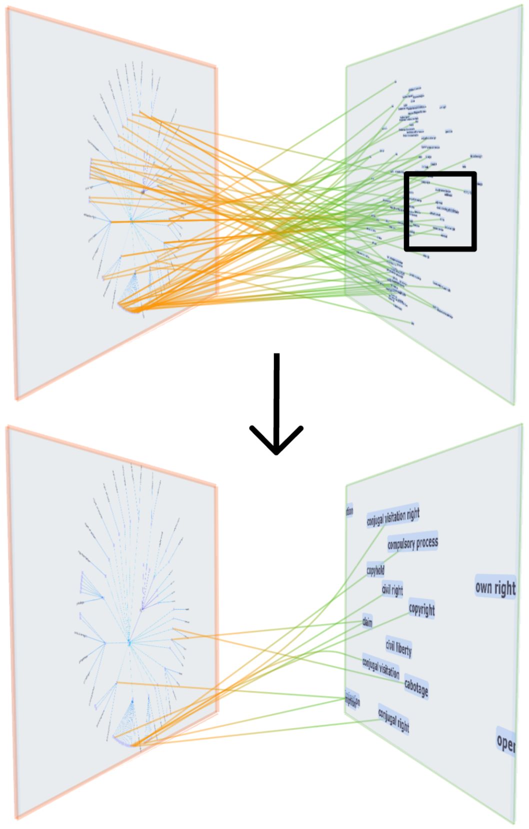


Fig. 7. The left plane is activated, revealing all edges from it. Through a click and drag on the right plane, a 2D zoom is performed, isolating a cluster of interest. The inter-plane edges are filtered in real time to show only those connecting visible nodes, revealing that this lexical cluster is related to a region of the WordNet hyponymy tree near the bottom.

vation). Node activation values determine inter-plane edge visibility: edges between nodes with non-zero activation are revealed. Level of activation is inversely related to the alpha transparency of activated nodes and the inter-plane edges. So, the more transparent an activated node or edge, the further it is from a user-selected fully-activated node. Edge colour is used to indicate the direction of spreading activation. For each edge, the third closest to the source of edge activation is orange, the middle third is interpolated from orange to green and the final third, closest to the edge target, is green. Along with edge transparency decay, edge colouration will help an analyst follow the path of spreading activation. However, tracing a series of edges across planes may be a difficult task, even with the visual support provided through colouration and transparency. We plan to investigate techniques such as animated edge propagation to help trace relationships amongst visualizations.

Inter-plane edges support cross-visualization queries. For example, alphabetic clustering, while a common organization for word search, is not useful for finding synonyms. Using VisLink to propagate an edge from a selected word in the clustered graph to a WordNet hierarchy will find this word within its synset structure, propagating back will find its synonyms within their alphabetic structure, allowing quick answers to questions such as, "*Across all senses, which synonyms of 'locomotion' start with 't'?*" This analysis is illustrated in Figure 8.

Inter-plane edges are only shown among visible nodes. So, if a technique such as filtering through degree-of-interest or distance measures, or clipping through zooming and panning the visualization on a plane causes some nodes to be invisible, their edges are not drawn. This can be used as an advantage for exploring the space of inter-plane edges: by filtering the view on a plane, the inter-plane edges can also be filtered (see Figure 7). Conversely, search techniques can be provided to reveal and activate nodes that match a query, thereby also activating their inter-plane edges (see Figure 3).

4 IMPLEMENTATION DETAILS

VisLink is implemented in Java, using the Java2D-Java Opengl (JOGL) bridge to import any Java2D rendering onto a visualization plane. We have augmented the popular *prefuse* interactive visualization toolkit [9] with the VisualizationPlane class, which implements the same API as the default 2D prefuse Display, and the InterPlaneEdge class, which handles edge drawing between planes. The result is that our visualization plane can accept any prefuse visualization without any changes. Interaction techniques on prefuse visualizations are also handled equivalently. In addition to providing for easy integration of existing visualizations with VisLink, this implementation provides for efficient rendering of the 3D space, achieving frame rates greater than 30fps on standard hardware (Intel Pentium 4, 3.9GHz processor with an ATI Radeon 550 graphics card). The prefuse visualizations are shown on the visualization planes as textures, updated only when prefuse calls for a display repaint. Inter-plane edges can be specified in the data set by referencing source and target visualization plane and node indices, or can be defined by a rule, such as, "*Create inter-plane edges among nodes with matching labels*" (rules such as these must be translated into code that produces paired node indices). Because the prefuse visualizations are drawn as textures on a 2D plane, VisLink could easily be extended to draw other shapes of visualization objects, such as cubes or spheres.

5 LINKING EXISTING VISUALIZATIONS

To demonstrate the ability of VisLink to add analytic power to existing prefuse-based visualizations, we used VisLink to bridge several of the demonstration applications that are distributed with the prefuse source code [9] (with minor colour changes). Data on the occupations of members of the 109th Congress before election was mined from the Congressional Directory,[1] along with the zip codes they represent. This was combined with databases of zip code locations and fundraising totals of candidates in three recent federal elections, both provided with the prefuse distribution. We used three visualization planes and defined indirect relations between them.

First, a prefuse Treemap [12] was used to show the relative popularities of various occupations before election (Figure 9, left). This was linked through the rule CANDIDATE had OCCUPATION to the prefuse-provided *congress* visualization by Heer [8]. *congress* is a scatterplot of individual fundraising success, ordered along the x-axis alphabetically by state of candidacy (Figure 9, center). This plot shows the candidates' party through node colour and whether they were running for the House or Senate through node shape. The y-axis shows fundraising success, and the range can be interactively altered with a slider (not shown in figure). This was linked to the prefuse reimplementation of the *zipdecode* [7] visualization of zip code geographic locations (Figure 9, right) through the rule CANDIDATE represents ZIP CODE. Inter-plane edges link occupations to candidate nodes and candidates to map regions they now represent. Complex questions such as, "*Where did the most successful fundraising former journalist get elected?*" can be quickly answered. To implement this visualization, the bulk of the work came through creating and parsing the new database (occupations and zip codes) to generate inter-plane edges from our rules.

6 DISCUSSION

The VisLink technique offers a new way to look at the relationships amongst visualizations, but there remain several difficulties and unresolved issues for future research. The creation of a VisLink visualization starts with the selection of the constituent visualizations to compare. Making this selection — finding appropriate data and choosing appropriate representations — is as difficult within VisLink as it is in everyday visual analytics work, and may be best handled by data and

---

[1] http://www.gpoaccess.gov/cdirectory

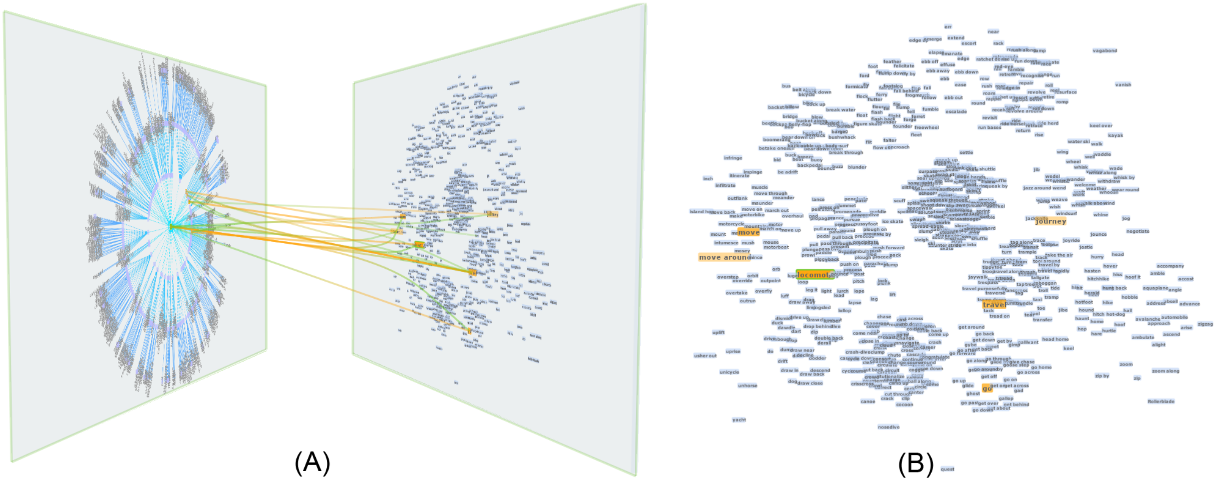


Fig. 8. Node activation and edge propagation. Nodes highlighted through spreading activation (orange without green border) reveal the alphabetic clustering of synonyms of the manually activated node (‘locomotion’, orange with green border), as discovered through spreading activation to the WordNet hyponymy graph.

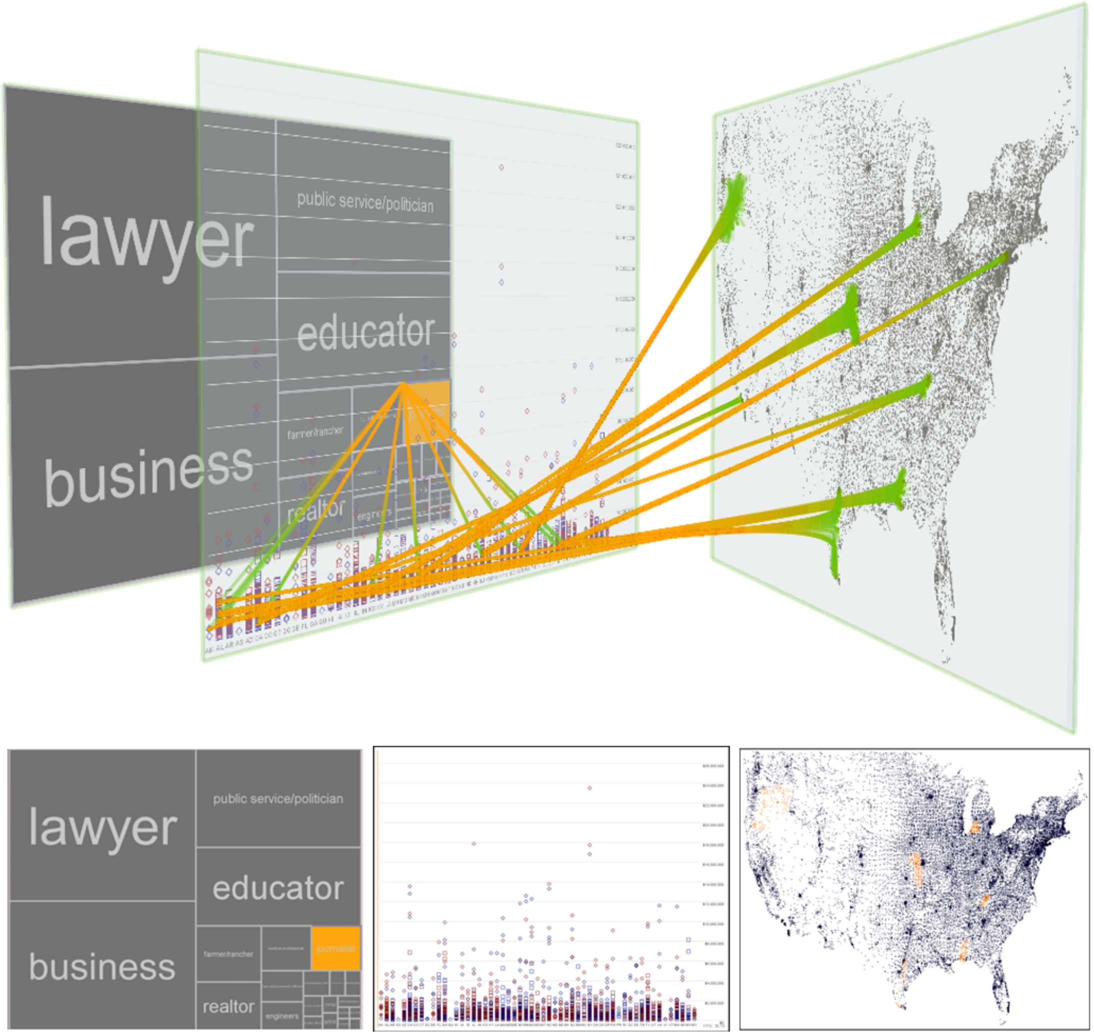


Fig. 9. VisLink was applied to bridge existing *prefuse* visualizations. Views of the constituent visualizations, from 2D equivalency mode, are shown along the bottom. The Treemap node ‘journalist’ is activated, propagating inter-plane edges to the scatterplot (showing journalists are not particularly outstanding fundraisers), and onward to the zip code regions that elected journalists now represent.

visual experts. Some visualizations, such as node-link diagrams, seem to work better with inter-plane edges than others, such as Treemaps and other types of embedded hierarchy, where it is more difficult to see the connections to non-leaf nodes.

For visualizations with rich sets of inter-plane relations, the familiar spaghetti graph of edge congestion can quickly become a problem. Through bundling of edges, individual node activation, filtering techniques, and the ability to view the edge set from a series of angles, we have attempted to provide tools to handle this. However, additional techniques, for example edge lenses [21] for 3D spaces, may improve the situation. The edge bundling technique we use works only for one-to-many edge sets. Many-to-many edge bundling as reported by Holten [10] requires a hierarchical structure as an invisible backbone. In the datasets we used, such a structure was not available, but this may be a promising area for future research.

Because VisLink contains any number of visualizations which may be pre-existing, the selection of colours for inter-plane edges is challenging. The orange-to-green colour scheme was selected because it interfered the least with the existing (predominantly blue) visualizations we imported into VisLink, and worked well both against a white background (for print) and a black background (on screen). However, orange-to-green is difficult to perceive for people with some forms of colour blindness. Inter-plane edge colouring will likely have to be customized to the constituent visualizations.

When working in a 3D space, issues of perspective must be considered. It is possible that perspective projection introduces a visual bias for closer regions of the planes and closer inter-plane edges. Directional bias may be introduced by the default views (side view presents bias toward vertical inter-plane patterns). 2D false symmetry effects may also occur. An analyst must be careful to view a VisLink visualization from several directions before drawing conclusions about apparent patterns in the data.

We have described VisLink primarily with examples from a single data set. In future work, we will apply VisLink to a rich set of problems in linguistic data analysis and other areas. The techniques and prototype we have described have not yet been experimentally evaluated. A comparative study against the existing techniques for multiple relationship visualization is necessary to understand the usability and utility of VisLink in more detail. Opportunities also exist to expand the capabilities of inter-representational queries, for example, by providing for a rich query language that can filter each visualization plane separately.

7 CONCLUSION

In this paper we have described VisLink, a visualization environment in which one can display multiple 2D visualizations, re-position and re-organize them in 3D, and display relationships between them by propagating edges from one visualization to another. Through reuse of the powerful spatial visual variable, we have introduced a method for visualizing multiple relations without any relation relinquishing its spatial rights.

The VisLink environment allows the viewer to query a given visualization in terms of a second visualization, using the structure in the second visualization to reveal new patterns within the first. By choosing a set of data items in visualization A and doing a one level propagation to visualization B, VisLink shows where items in A are related to items in B. Propagating the edges back again reflects the information gathered from visualization B to the structure of visualization A. Thus, using the example in Figure 8, starting from a similarity-based word visualization A, propagating edges from a chosen word into WordNet visualization B and back again reveals synonyms of the selected word in visualization A. Through spreading activation, bundled edges can be propagated between visualizations to any chosen depth.

VisLink displays multiple 2D visualizations on visualization planes while maintaining full 2D interactivity for each component visualization. 3D interaction widgets are provided to simplify 3D interaction and navigation. Relationships among visualizations can be revealed using methods such as selection and filtering for addressing edge congestion. Ongoing research will investigate techniques for managing edge congestion, such as alternative bundling techniques and the use of interaction tools to isolate edge sets of interest. In future work, through application to additional problems, and evaluation against related techniques, we will develop a clearer understanding of the usability and utility of the techniques and prototype we have described.

ACKNOWLEDGEMENTS

Thanks to Gerald Penn, Petra Neumann, Mark Hancock, Tobias Isenberg, Uta Hinrichs, and Matthew Tobiasz for their assistance. Funding for this research was provided by NSERC, iCore, and NECTAR.

REFERENCES

[1] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[2] D. A. Bowman, E. Kruijff, J. J. LaViola, Jr., and I. Poupyrev. *3D User Interfaces*. Addison-Wesley, 2005.

[3] S. K. Card, G. G. Robertson, and W. York. The WebBook and the Web Forager: An information workspace for the world-wide web. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 1996.

[4] G. A. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3(12):346–349, 1974.

[5] C. Collins. Docuburst: Radial space-filling visualization of document content. Technical Report KMDI-TR-2007-1, Knowledge Media Design Institute, University of Toronto, 2007.

[6] J.-D. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant. Overlaying graph links on Treemaps. In *Proc. of IEEE Symp. on Information Visualization, Poster Session*, pages 82–83, 2003.

[7] B. Fry. zipdecode. http://acg.media.mit.edu/people/fry/zipdecode/, 2007.

[8] J. Heer. congress. http://www.prefuse.org/gallery/congress, 2007.

[9] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM Press, Apr. 2005.

[10] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE Symp. on Information Visualization*, 12(5):741–748, Sept.–Oct. 2006.

[11] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proc. of IEEE Visualization*, pages 361–378, 1990.

[12] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proc. of IEEE Visualization*, pages 284–291. IEEE Computer Society, 1991.

[13] J. Kamps and M. Marx. Visualizing WordNet structure. In *Proc. of the 1st International Conference on Global WordNet*, pages 182–186, 2002.

[14] J. Light and J. Miller. Miramar: A 3d workplace. In *Proc. of IEEE Intl. Professional Communication Conf.*, pages 271–282, 2002.

[15] M. J. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Proc. of IEEE Visualization*, pages 401–408, Oct. 2003.

[16] G. A. Miller, C. Fellbaum, R. Tengi, S. Wolff, P. Wakefield, H. Langone, and B. Haskell. WordNet: A lexical database for the English language, Mar. 2007.

[17] P. Neumann, S. Schlechtweg, and M. S. T. Carpendale. Arctrees: Visualizing relation in hierarchical data. In K. W. Brodlie, D. J. Duke, and K. I. Joy, editors, *Proc. of Eurographics - IEEE VGTC Symp. on Visualization*, pages 53–60. The Eurographics Association, 2005.

[18] C. North and B. Shneiderman. Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In *Proc. of Advanced Visual Interfaces*, pages 128–135, May 2000.

[19] T. Pedersen, S. Banerjee, and S. Patwardhan. Maximizing semantic relatedness to perform word sense disambiguation. Technical Report UMSI 2005/25, University of Minnesota Supercomputing Institute, 2005.

[20] B. Shneiderman and A. Aris. Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE Symp. on Information Visualization)*, 12(5):733–740, Sept.–Oct. 2006.

[21] N. Wong, S. Carpendale, and S. Greenberg. EdgeLens: An interactive method for managing edge congestion in graphs. In *Proc. of IEEE Symp. on Information Visualization*, pages 51–58, 2003.

[22] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *Proc. of IEEE Symp. on Information Visualization*, pages 43–50, 2001.