VisLink: Revealing Relationships Amongst Visualizations

Christopher Collins[1], Sheelagh Carpendale[2]
[1]Department of Computer Science, University of Toronto
[2]Department of Computer Science, University of Calgary

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 at will, side by side, in parallel, or in chosen placements that provide favoured views. Relationships, connections and patterns between visualizations can be revealed.

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, or even expect, visual explanations also continues to grow. For example, while creating visualizations of multi-variate data remains a familiar challenge, the visual portrayal of two set of relationships, one primary and one secondary, within a given visualization is relatively new (e. g. [6, 14, 8]). With VisLink, we extend this latter direction, making it possible reveal relationships between one or more primary visualizations. This enables new types of questions. For example, consider a linguistic question such as whether the formal hierarchical structure as expressed through the `IS-A` relationships in WordNet [13] 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.

VisLink supports the display of multiple 2D visualizations, each with its own use of spatial organization to best visualize its data relationships and each is 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. For example, alphabetic clustering, while a common word organization for many word search operations, is not particularly good for finding synonyms. Propagating an edge from a selected word in the clustered graph to a WordNet hierarchy will find this word with in its `IS-A` structure, propa-

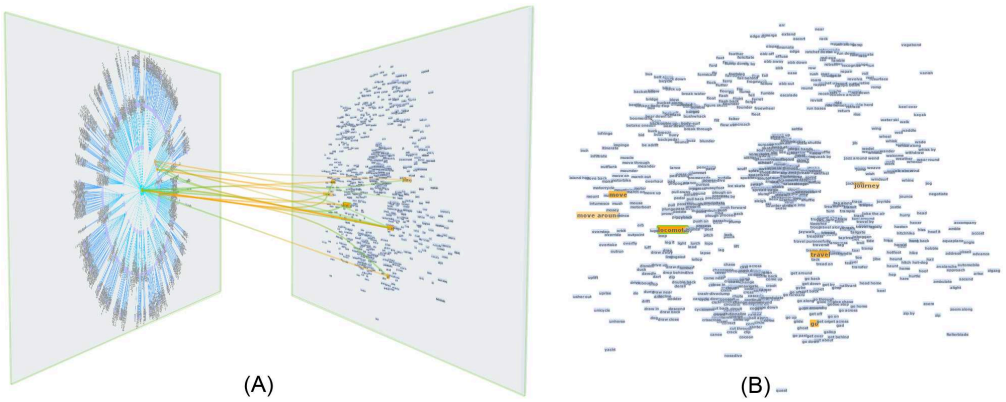

Figure 1: Individual nodes on any visualization plane may be activated. Activated nodes are drawn with a green border (see 'locomotion' at left of B). Inter-visualization edges are drawn from activated nodes. The target nodes of these edges, are, in turn, activated. As the activation spreads between planes, the level of activation drops off exponentially. The level of activation is mapped to the alpha transparency of the edges and the orange fill of activated nodes. Note that the 2D equivalency view at right reveals some nodes activated by spreading (orange without green border), and their level of activation. Spreading activation reveals that these nodes, while they have no explicit relation on this visualization plane, are related on another plane. Here we see that activated nodes are synonyms of the activated node 'locomotion', as discovered through the VisLink inter-visualization relations.

gating back will find its synonyms within their alphabetic structure (see Figure 1). VisLink enables the reuse of the spatial variables, thus supporting efficient information encoding and providing for powerful visualization bridging which in turn allows inter-visualization queries. Relationships, connections and patterns between visualizations can be revealed.

2 Formalizing Visualizations of Multiple Relations

VisLink extends existing approaches to visualizing multiple relationships to provide for revealing relationships among 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 Austin, Texas, 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 $R_A(D_B)$

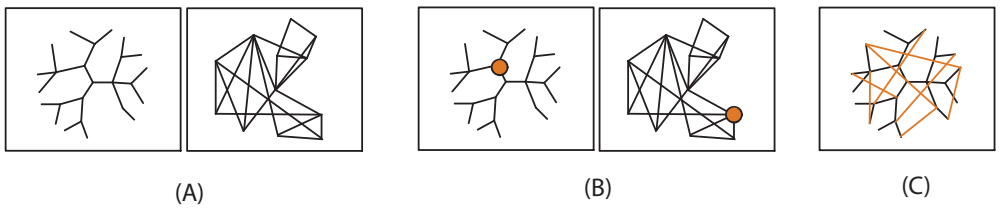

Figure 2: Current approaches to comparing visualizations include (A) manual comparison (e. g. printed diagrams or separate programs), (B) linked multiple views with highlighting (e. g. Snap-together-visualization [15]), and (C) layout based on one relationship, other relationships drawn upon it (e. g. ArcTrees [14] and TreeMap overlays [6]).

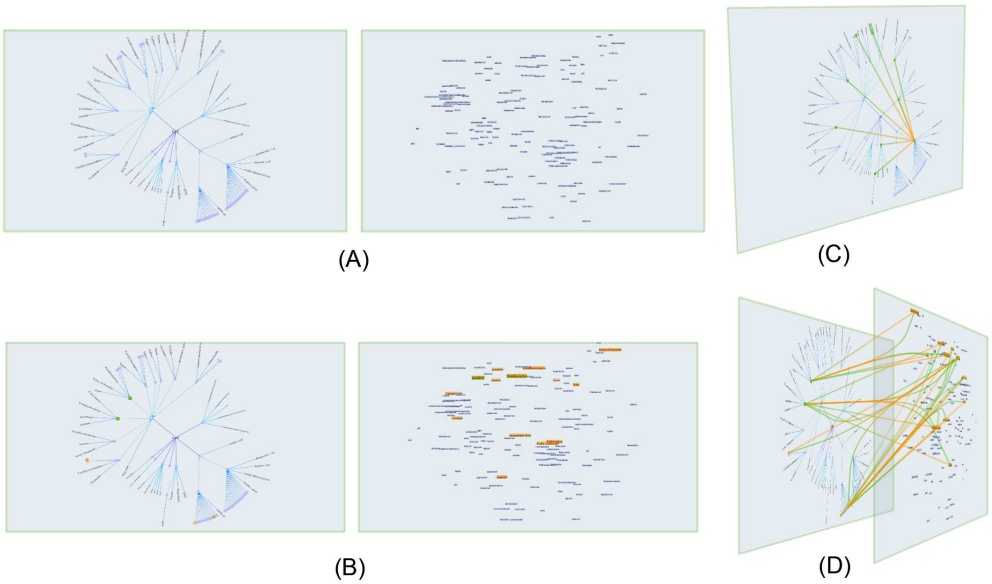

Figure 3: VisLink encompasses existing multiple views techniques of (A) manual comparison, (B) linked comparison with highlighting, and (C) layout based on one relationship type with other relationships drawn atop. VisLink extends this continuum to direct linking of any number of multiple views (D).

would be property tax on houses in San Francisco. Creating a first visualization, Vis_A, of these relationships $R_A(D_A)$ we will write as $Vis_A \rightarrow R_A(D_A)$. A second visualization, Vis_B, of the same set of relationships would be $Vis_B \rightarrow R_A(D_A)$.

In the remainder of this section, we define each of the previously mentioned current approaches to relating visualizations using this notation, compare and contrast them, and 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 need for the current task. Any relations on any data may be compared manually in this way (see Figure 2A).

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 [15]). 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 action and/or highlights in one view can be reflected on other views. This temporarily activated visual connection can be a great advantage over finding the related data item manually but the relationships themselves are not explicitly visualized (see Figure 2B).

2.3 Compound Graph Visualizations

There are now a few examples of compound graph visualizations, such as overlays on TreeMaps [6], ArcTrees [14], and Edge Bundles [8]. Figure 2C shows a simple diagram of this. Compound graph visualizations are created as follows:

Given: Data set D_A containing a primary relationship $R_A(D_A)$ and a secondary relationship $R_B(D_A)$

Problem: Show second set of relationships in conjunction with a visualization of the first set of relationships on a given data set.

Step 1: 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 Vis_A 'spatial rights'.

Step 2: Create a visualization of $R_B(D_A)$ 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 $Vis_B \rightarrow R_B(D_A)$. Edge Bundles 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 Substrate Visualizations

Shneiderman and Aris [18] 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}, ..., D_{A_n}.

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

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

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

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 $R_A(D_A)$, $R_B(D_A)$, ..., $R_N(D_A)$.

Problem: Provide 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 are related to the structure in another set of relationships.

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 $R_A(D_A)$, $R_B(D_A)$, ..., $R_N(D_A)$ 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-visualization 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 $D_A, D_B, \ldots, D_N$ and the existence meaningful relationships, $T(D_i, D_j)$, among datasets such that (i, j) are any of A, B, ..., N, VisLink can be used as is with no further extensions to relate $Vis_A \rightarrow R_A(D_A)$, $VisB \rightarrow R_B(D_B)$, ..., $Vis_N \rightarrow R_N(D_N)$, by using $T(D_i, D_j)$ to create inter-visualization edges.

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

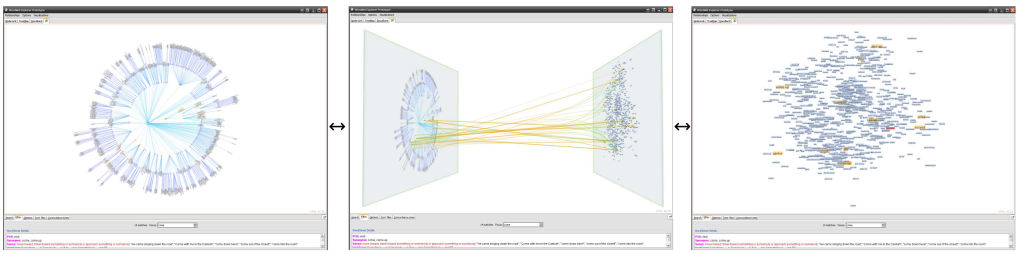

Figure 4: VisLink visualization planes are the same dimensions as the view-port, allowing for short cut keys which provide quick access to animate camera to view plane for equivalent-to-2D interaction and examination. Plane and camera orientations are saved and restored when 2D equivalency mode is ended. (A) View of plane 1, showing hyponyms of verb 'move', with highlighted search results for 'come'. (B) Search results on plane 1 activate inter-visualization edges. Nodes connected to search results are highlighted on plane 2, a similarity clustering of words related to 'move'. Propagated results are also visible when plane 2 is viewed in 2D equivalency mode (C).

multi-relation visualization approaches (see Figure 3A–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-fig:fourStylesD).

3 VisLink: Comparison with Visualization Planes

In order to provide for a visualization space in which multiple data-related visualizations can be related, 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 a series of queries answered by inter-visualization edge propagation are 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 linguistics, and several visualizations using lexical data have been reported (e. g., [10, 17, 5]).

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 [13] describes the hierarchical `IS-A` relation over synsets, which are sets of synonymous words. For example, *Siamese cat* `IS-A` *house cat*. The `IS-A` relation is also called hyponymy, so *chair* is a hyponym of *furniture*. We use hyponymy to build

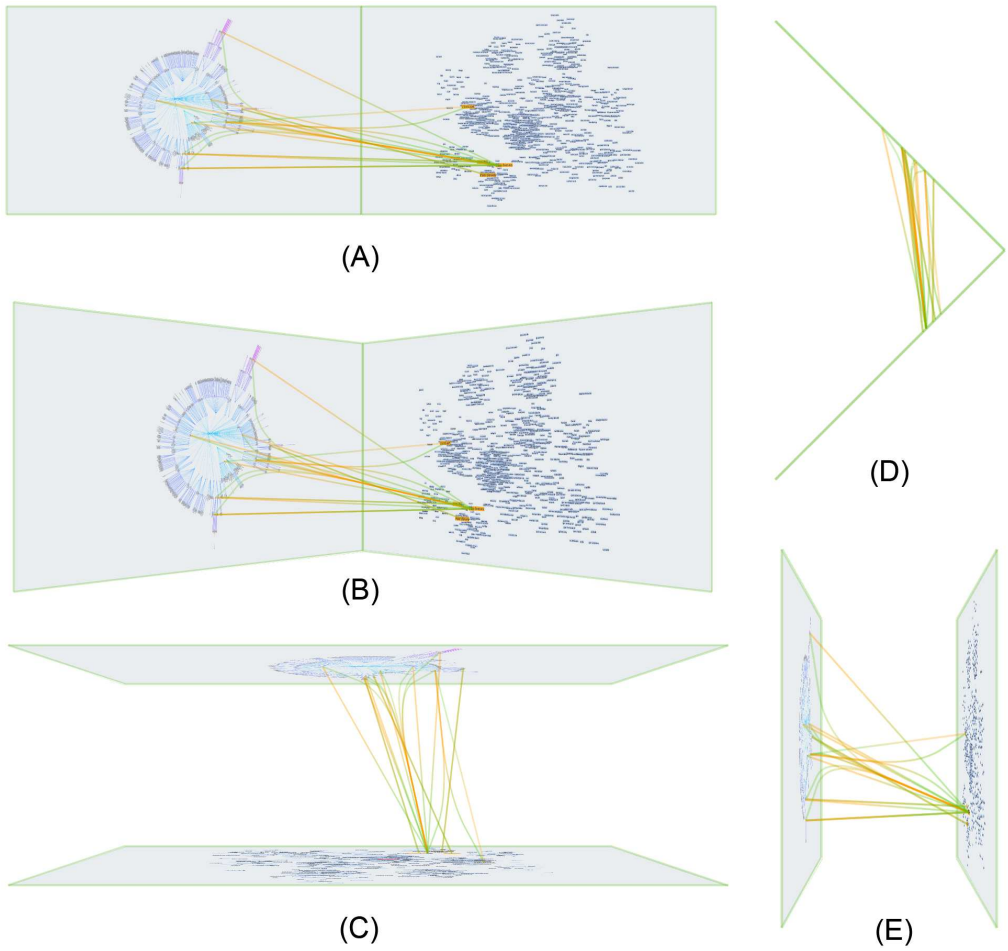

Figure 5: Keyboard short cuts provide for animated transition to default views, easing navigation in the 3D space. Views are (A) flat, (B) book, (C) top, (D) book top, and (E) side.

animated radial graphs [19], which serve as our $Vis_A \rightarrow R_A(D_A)$. Synsets are show 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 4A. We also show a TreeMap [9] view of this data in some figures, which will be $Vis_B \rightarrow R_A(D_A)$.

Words can also be related by their similarity. Similarity can be a surface feature, for example, orthographic (alphabetic) similarity clustering, or it can be based on a semantic feature. Pedersen et al. [16] review several semantic similarity measures, for example similarity as measured by lexical overlap in the dictionary definitions of words. We use a force-directed layout [1] to perform orthographic similarity clustering on words. 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 4C.

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 VisLink by looking up a specific 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 the 2D transparent visualization planes are positioned. These visualization planes act as virtual displays, upon which any data visualization can 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 [11] 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-visualization edges are deactivated, and the display becomes equivalent to 2D (see Figure 4). 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. 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 mouse

click, activating radial layout animation [19]. 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.

While usual capabilities for navigation in the 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 to preset viewpoints [11]. These viewpoints allow visualization planes to be viewed from the front (planes parallel and side by side) (see Figure 5A), with relative plane orientation of book view (planes perpendicular and meet at an edge) (see Figure 5B), top (see Figure 5C and D), or in opposition (planes parallel and stacked) (see Figure 5C and E). By choosing one of these viewpoints, the user can recover from any disorienting view manipulation.

As a solution to 2D plane interaction in a 3D space, we follow McGuffin et al. [12] and provide for manipulation of visualization plane position and orientation using a set of restricted movement widgets. Edge widgets provide for hinge movement about the opposite edge, and a center widget provides for translation, accordion style, along the axis between the planes (see Figure 6). Widgets become visible when the pointer is over their position, otherwise they are hidden from view to prevent data occlusion.

3.3 Adding Inter-Visualization Edges

Directed edges, using orange to green colour blending to show direction, 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 (nodes among visualization planes have some relation defined within the data). Indirect relations can also be generated from rules relating the data on each plane. 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-visualization 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 inter-visualization edges reveal a second-order relation: the relationship between different types of node relations on the same data. This visualization can answer questions about how well the formal structure of WordNet, as designed by cognitive linguists, represents the real-world usage of words as gathered from statistical similarity data. 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 tree of hierarchical tags from a folksonomy, for example a photo database. The inter-visualization connections could be derived by the rule `PERSON used TAG`, an indirect relation.

Visualization planes in VisLink are connected by directed edges. All inter-visualization

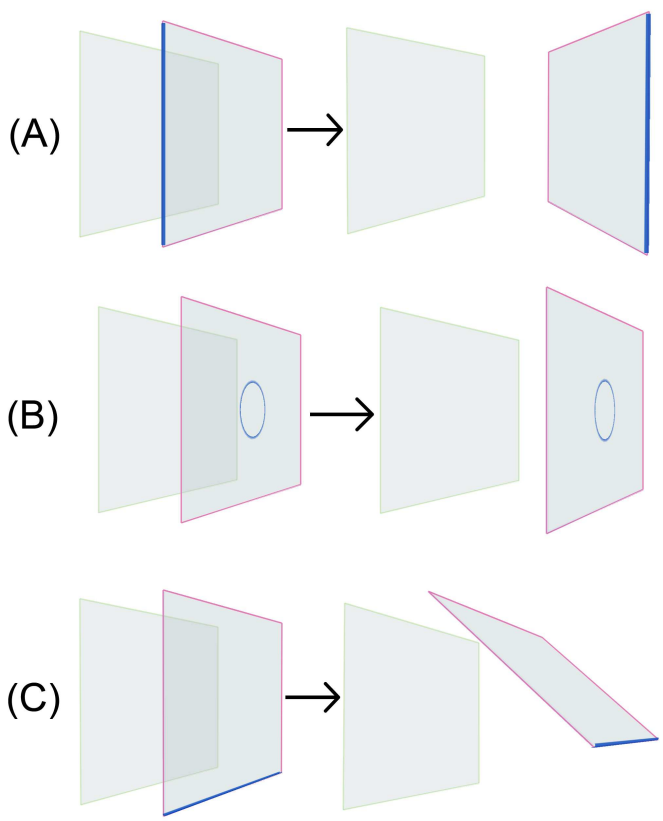

Figure 6: Visualization planes are independently manipulated with three widgets: (A) side 'book pages' rotation, (B) center 'accordion' translation, and (C) bottom 'garage door' rotation.

edges are specified with a single source node on plane i and one or more target nodes on plane j. Direction is shown using colour blending from orange to green. 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 collinear and split as they approach the targets. By using alpha blending, the more semi-transparent curves that are coincident, the stronger the bundled edges appear (see Figure 7).

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

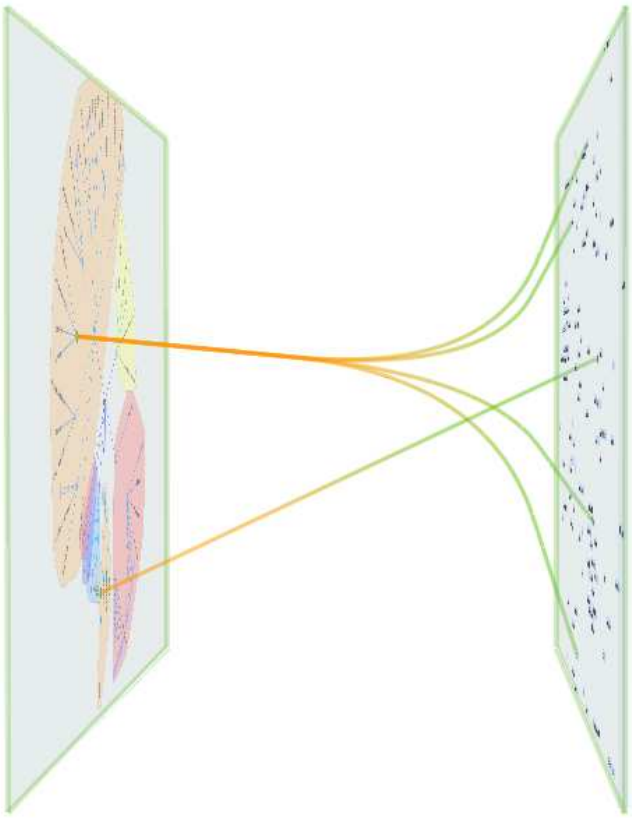

Figure 7: VisLink inter-visualization edge detail: one-to-one edges are straight, one-to-many edges are bundled. Alpha blending provides for stronger appearance of bundled edges.

another, the rendering is updated to show the relations between the new neighbours.

3.4 Using Inter-Visualization Edges

Inter-visualization edges can be activated either on a per-plane basis (see Figure 8A) or a per-node basis (see Figure8B). Activating an entire plane could 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 activated on one plane, it is highlighted in orange with a green border and all inter-visualization edges originating at that node are revealed. The target nodes for those edges are then activated, although to a lesser degree. Edges originating at these nodes are then drawn and the activation is propagated iteratively up to a user-selected number of 'reflections' between planes. The level of activation exponentially decays with each iteration. Level of activation is inversely related to the alpha transparency of activated nodes and the inter-visualization edges. So,

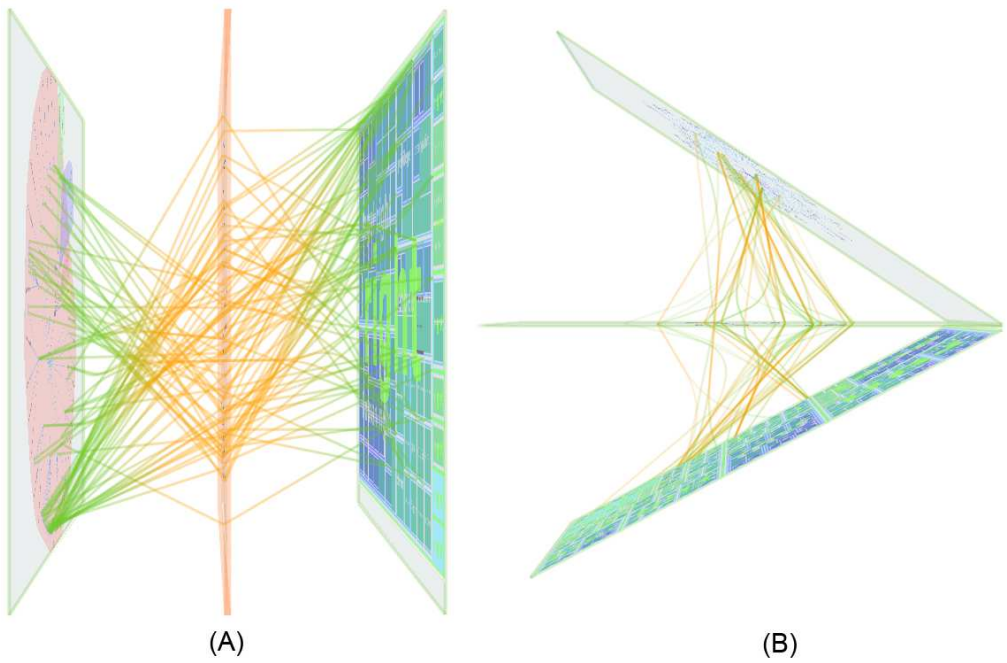

Figure 8: (A) Side view of 3 visualization planes. The center plane (lexical similarity clustering) is activated, indicated by its orange frame. All inter-visualization edges from this plane are drawn to its neighbours. At left, a radial graph of WordNet hyponymy, at right, a TreeMap of the same data. (B) Top view of 3 visualization planes in 'book' orientation. Several nodes are activated on the outer planes, spreading their activation through inter-visualization edges.

the more transparent an activated node or edge, the further it is from a user-selected fully-activated node. Figure 1 illustrates an analytic outcome of this technique.

Inter-visualization 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-visualization edges. By filtering the view on a plane, the inter-visualization edges can also be filtered (see Figure 9). Conversely, search techniques can be provided to reveal and activate nodes that match a query, thereby also activating their inter-visualization edges (see Figure 4).

4 Implementation Details

VisLink is implemented in Java, using the Java2d-Java Opengl bridge to import any Java2d rendering onto a visualization plane. We have augmented the popular prefuse interactive visualization toolkit [7] with the VisualizationPlane class which implements the same API as the default 2D prefuse Display. The result is that our visualization plane can accept any

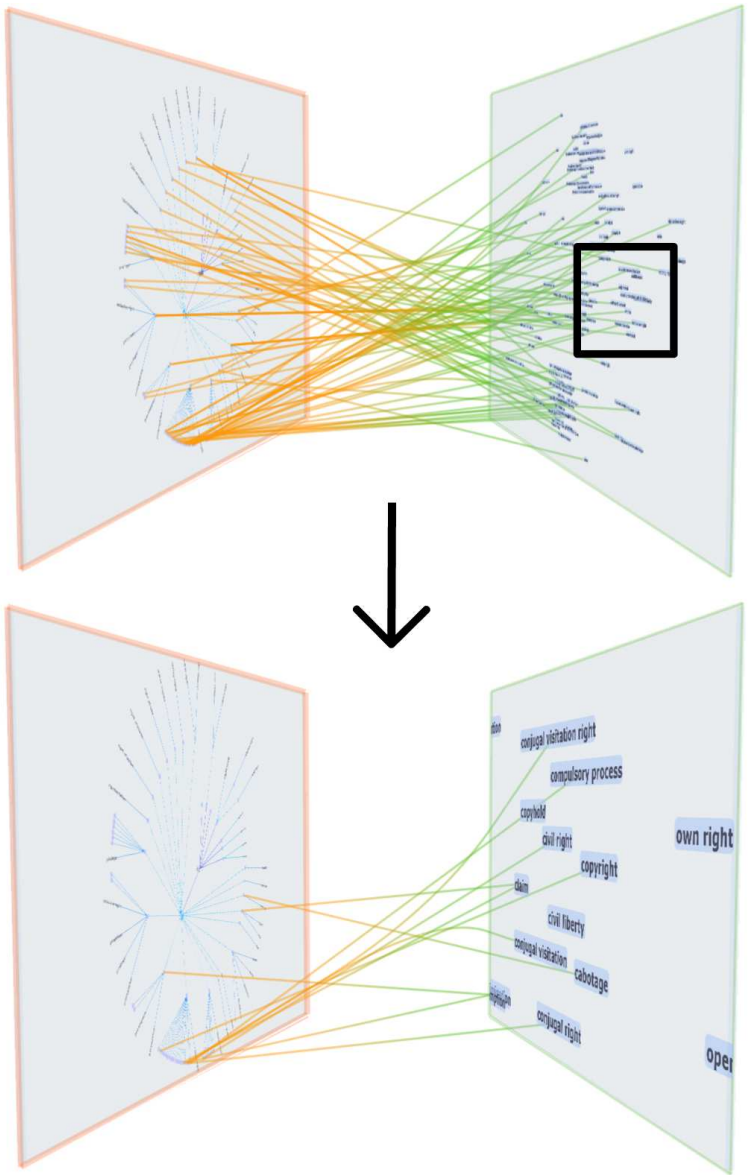

Figure 9: By zooming the visualization on the right plane to a cluster of interest, the inter-visualization edges are filtered to show only those connecting nodes visible on both planes, revealing that this lexical cluster is related to a region of the WordNet hyponymy tree near the bottom.

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-visualization edges can be specified in the data set by referencing target visualization plane and node indices, or can be defined by a rule, such as "create inter-visualization edges among nodes with matching labels". 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 Conclusions and Future Work

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 1, 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.

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. For example, different semantic relatedness measures have been algorithmically compared by Pedersen et al. [16], and we plan to use VisLink to visually augment this analysis. 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.

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. L. 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 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. Wang, 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] 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.

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

[9] 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.

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

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] S. Ploux and H. Ji. A model for matching semantic maps between languages (French/English, English/French). *Computational Linguistics*, 29(2):155–178, June 2003.

[18] 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, September-October 2006.

[19] 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*. IEEE Computer Society, 2001.