

Redgraph: Navigating Semantic Web Networks using Virtual Reality (PAPER 166)

Abstract

We present *Redgraph*, a generic virtual reality visualization program for network data based on Resource Description Framework (RDF), the primary standard of data underlying the Semantic Web. Our approach in visualizing this network-based data takes advantage of user-interaction to bypass a number of difficult questions in the 3-dimensional graph visualization layout problem by relying on the user themselves to interactively “extrude” the nodes and edges of a 2-dimensional network into the third dimension. We applied *Redgraph* to data from the United States Patent and Trademark Office in order to explore the social networks of innovation in the history of computer science. We present the results of a user study comparing extrusion vs. no extrusion on fine-grained connectivity tasks and broad network observation tasks. The use of extrusion showed a statistically significant improvement on the time to accomplish the fine-grained connectivity tasks. The broad network observation tasks showed no significant difference between the extrusion interface and the non-extrusion interface.

CR Categories: H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities—applications;

Keywords: networks, extrusion, visualization, virtual reality/virtual reality, network visualization, semantic web, extrusion

1 Introduction

While both the computational and social sciences are increasingly interested in networks and immense amounts of network data is available on the Web, intuitively understanding the structure of networks is still more of black art than science. Although rigorous mathematical frameworks are being developed to analyze networks, often even the simplest visualization can help clarify the structure of networks, allowing the human user to visually identify the parts of the network that are most of interest. While there has been research into powerful network visualization algorithms, much of the visualization of networks concentrates in two dimensions [Henry and Fekete 2006]. The two-dimensional visualization of networks often produces dense and confusing “spaghetti-like” structures that mere visual inspection can not easily grasp.

Our virtual reality-based network visualization program relies on user-directed extrusion thereby allowing the user to transform a network from 2-dimensions into 3-dimensions. While our work builds on previous work in three-dimensional visualization that allowed

users to reposition nodes [Crutcher et al. 1995], our network visualization program differs from previous work by allowing users to “extrude” nodes from 2-D to 3-D in fully immersive and interactive CAVETM-like environments [Cruz-Neira et al. 1993].¹ The toolset is not customized for any particular dataset or application, but is rather a fully generic toolset capable of visualizing any network that can be described using the standardized RDF (Resource Description Framework) data model [Hayes 2004], a World Wide Web Consortium (W3C) Recommendation for graph-based data models that is flexible enough to handle network data found on the web.

2 Network Layout

A network is as a mathematical graph, where a graph G is a set of nodes (vertices) n connected by links (edges) l , thus $G = (n, l)$. One important aspect to note about networks is that they are often less amenable to mathematical analysis than other discrete mathematical structures. For example, typical networks in “the wild” (as opposed to Erdős-Rényi random graph models [Bollobas 1985]) obey a power-law distribution of connectivity between nodes and links ($a = cx^b$) [Barabasi and Albert 1999]. One cannot do any statistics such as means, since these rely on properties that do not hold in power-law distributions [Newman 2005]. Therefore, visual inspection is crucial when understanding networks. From a visual perception point of view, network node layout is of primary importance. A large number of algorithms have been developed in both 2-D and 3-D network visualizations. Most two-dimensional visualization algorithms consider the links to be simulated “springs” and the nodes to have some repulsive force, and these simulated springs are used to model the forces between each pair of nodes. “Springer-embedding” algorithms like the the Kamada-Kawai algorithm minimize an energy function which is the sum of the forces on all nodes in the network, and have been shown to produce aesthetically pleasing diagrams [Kamada and Kawai 1989]. The initial 2-D layout of the network can use any algorithm. For our study, we have relied on the Kamada-Kawai algorithm as implemented in the *GraphVis* toolkit [Ganser and North 1993]. For larger datasets we will use the *GraphVis* version of the Fruchterman-Reingold algorithm [Fruchterman and Reingold 1991].

3 The Extrusion Technique

One very common phenomenon in the visualization of networks is the overwhelming complexity when one first encounters a new dataset, resulting in an inability to understand the underlying structure. This problem is especially pronounced in data-sets in which either the higher-dimensional structure does not map well onto two or even three dimensions or when the number of connections is so dense that visualizing the networks leads to a “spaghetti-like” structure that confuses, rather than clarifies, the understanding of the user. While algorithms like Kamada-Kawai can be extended to three dimensions, there is the more insidious problem of determining where precisely the user should be placed in the 3-D visualization [Eades et al. 1997]. Visualizing the network in three-dimensions may or may not help, since placing the user’s view

¹CAVE is a registered trademark of the Board of Trustees of the University of Illinois.

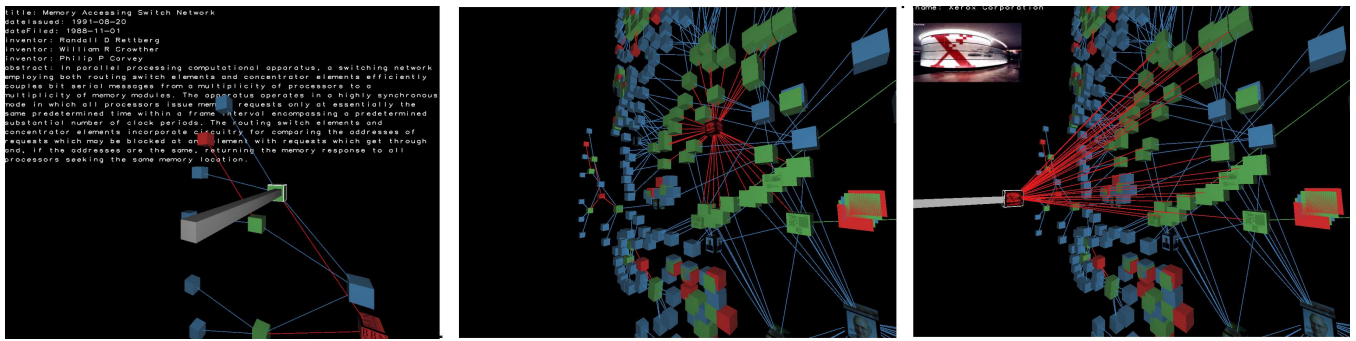


Figure 1: *Redgraph Snapshots: Left - Node Metadata, Center - 2D Network Layout, Right - Interactively extruding a Node into the 3-D*

point “in the middle” of a three-dimensional visualization may hinder rather help the user. For example, the user’s viewpoint may hide parts of the network structure from view. One effective way to visualize networks in three-dimensions that cuts this Gordian knot is the use of “extrusion.”

Extrusion lets the user be in charge by having her select the nodes of interest from a two-dimensional structure, and then “pull” them out into a three dimensional space, which literally “extrudes” the two-dimensional data into three dimensions. In detail, the user first sees the data network in 2-D on a virtual plane of the virtual environment, as shown in the center picture in Figure 1. The user then uses the wand to rapidly browse the nodes. Upon intersection between the wand and a node, the visualization program displays all information contained in the RDF data model as shown in the leftmost picture of Figure 1. When the user discovers a node that they are interested in, they then continue holding down a button on the wand, and pull or push the node into the third dimension, as shown in rightmost picture in Figure 1. As a result of this movement of the node, the user stretches the edges between the nodes into three dimensions as well.

This method has a number of advantages. First, many users can quickly identify clusters and other interesting phenomenon in two-dimensional data-structures. By beginning the visualization in two-dimensions, we take advantage of the user’s ability to conceptualize the data in two-dimensions. Extrusion avoids the problems of the three-dimensional layout of data by using well-understood and optimized two dimensional visualization algorithms for the initial layout. Most importantly, extrusion allows the user to interact with the data using both their visual and proprioceptive-motor abilities. Human interaction with the network through extrusion allows them to dynamically re-cluster data using parameters that may be difficult or impossible for the computer by itself detect, and so provides the third-dimension as an abstract “sketchpad” for the placement of nodes. This solves the issue of the placement of the user viewpoint in the a three-dimensional data-set [Eades et al. 1997], by letting the user interactively create their own viewpoint through the movement of the network nodes using extrusion.

4 Resource Description Framework

The software we have created is aimed to be a general-purpose tool allowing any sort of network to be visualized, ranging from social networks in Chinese villages to the citation-networks present in the United States Patent Office. The mathematical definition of networks, while adequate to define a network, does not define what types of structured data each vertex or edge represents. Some data may be numeric, some may be textual or imagery, and to integrate this into a network one must have some way of “attaching” this in-

formation to particular nodes and edges. Furthermore, the network may have different *types* of nodes and edges, with different properties that may impact the visualization. One type of link might be citation with a relation between a paper and its author, and the other might be a link between the author and the institution that employs them.

A standard data-model, called the Resource Description Framework (RDF) has been created by the W3C (World Wide Web Consortium) to describe network data, being originally conceived by the W3C as an abstraction of the “link and document” structure of the Web, where nodes are documents and links are directed edges. URIs (Uniform Resource Identifiers) are used to identify both nodes and vertices. This formalism is a descendant of semantic networks, which inspired some of the first work in three-dimensional network visualization [Fairchild et al. 1988]. In this logical formalism, the origin of the directed link is the *subject*, the link itself a *predicate*, and the endpoint of the directed link the *object*, as in Figure 2.

The primary advantage of the RDF data model is that it allows seamless merging of different networks. Graphs that have been serialized differently can be compared due to the use of URIs as unique identifiers and the straightforward formal semantics used by RDF (difficult to do in the generic case with tree-structured data-formats without formal semantics such as XML). RDF also features a rich number of tools ranging from APIs to optimized data-stores, and the current normative standard that should allow this toolset to be used over many different applications. Currently, there are over 9 billion “triples” (or node-link structures) described using RDF, ranging from data in the biomedical sciences such as the Gene Ontology Annotations (over 6 million node-links, available at <http://www.geneontology.org>) to a conversion of the structured data in Wikipedia known as Dbpedia (over 91 million node-links, available at <http://dbpedia.org>). Furthermore, RDF can be extended to deal with class-based subsumption using schemas and constraints that build upon further W3C standards. A number of tools exist that can serialize custom tab-delimited forms to RDF, allowing many existing data formats to be visualized using RDF. These qualities allow our network visualizer to qualify as fully abstract and generic.

5 Redgraph Implementation

One of the guiding principles of RDF is that it is self-describing. One can follow the edge of an RDF data source to discover more data about any node. *Redgraph* exploits this feature to discover what types of data are attached to each data node. The name of each RDF edge is used to qualify the data (e.g. image, abstract, assignedto). *Redgraph* then uses the data-types specified by RDF in combination with the node-type to appropriately render different data attached to each node (e.g. text or 2-D images).

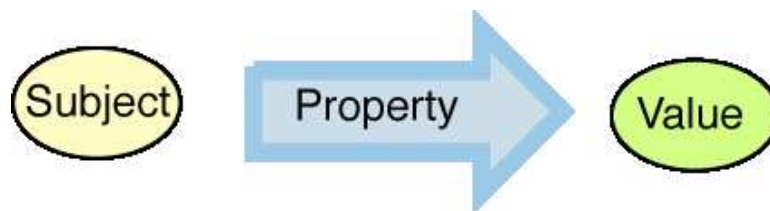


Figure 2: The RDF “triple” data model

Redgraph uses the generic C-based Redland API (<http://librdf.org>) to load the data file into a virtualized data structure. A set of parameters is used to determine which nodes and edges are rendered as part of the network visualization, and which nodes and edges are considered meta-data that is only shown as “details on demand” [Schneiderman 1996] when a user “touches” the node with the wand. The network visualization and extrusion technique is implemented in the *Syzygy* [Schaeffer and Goudeseune 2003] virtual reality library. *GraphViz* [Ganser and North 1993] is used for the initial 2-D network layout using the Kamada-Kawai algorithm [Kamada and Kawai 1989].

6 Experiment: The PARC USPTO Dataset

Our experiment took place in the local CAVE-like system, a 6-sided CAVE-like system. The structure is 3m x 3m x 3m with a rigid ceiling and floor, flexible walls, and a sliding door. The local CAVE-like system uses a X PC workstation cluster running X with XXXX graphics cards as the graphics rendering engine. XXXX projectors connect with XXXX active stereo glasses. Tracking is supported by the Intersense XXXX system. The Audio system consists of an XXXX running Supercollider and X free-field XXXX Active studio speakers.

For our initial data-set, we used freely-available data from the United States Patent and Trademark Office (USPTO) in order to explore the social networks of innovation in computer science. Since the USPTO patent database is massive (totalling over seven million patents records in 2006), we selected a subset of patents relating to computing history that related the work in personal computing at Xerox PARC by gathering a list of employees who worked at Xerox PARC and then searching the entire patent database for any patents filed under their names. This resulted in a data-set of patents, resulting in 7667 nodes and links in RDF.

For our network visualization, the patents (green), inventors (blue), and institutions (red) were each nodes, and the edges were the relationship between a patent and an inventor, and a patent and the institutions it was assigned to. Auxiliary information for each node was displayed when the user intersected that node with the wand. Patent nodes contained the patent abstract and any images associated with the abstract. Inventor nodes contained their name, location(s), and occasionally a picture of the individual. Institution nodes only contained the institutional name and sometimes an image of the building. Using the network visualization, one could trace a number of obvious connections, such as what institutions received what number of patents or which individual created the most patents. But one could use the visualization to trace less simple relationships such as the flow of employees from institution to institution over time, exemplified by the movement of a vast number of employees from Xerox PARC to DEC and later to Microsoft. The visualization also let one see the influence of institutions and people as determined by the relationships between their patents. This allows one to discover how a single patent, such as the patent for Douglas Engelbart of the “mouse” at SRI had a tremendous influence at Xerox PARC

through it was rarely directly cited (instead being indirectly cited through the patent filed by Charles Thacker, a Xerox PARC employee).

A number of users, mostly from a background in the humanities and the social sciences, were shown the use of extrusion with the PARC USPTO dataset. Afterwards, they were asked to give qualitative feedback. Users felt significant changes in their understanding of PARC were made by exploring the network: “Actually, I would say that the biggest surprise for me was the number of single patents and solo inventors associated with PARC. Like a lot of people who are not well versed in this particular piece of history, I went into the model with a lingering image of some sort of cabal comprised of two or three people. Put another way, the diversity of the field was very informative.” The users were not without criticism, as another user insisted that “because the 3-D part worked so well, I found myself very effectively immersed in that aspect of the interface,” which actually led to problems when the patent abstracts were displayed in the local CAVE-like system, since “the first time the textual information appeared on the rear screen, the immersiveness of the experience fell away and I was back to a 2-D world.”

We performed a pilot study which tested the effectiveness of extrusion into the third dimension. The study consisted of 8 subjects, ranging in ages 21 to 53, with 3 females and 5 males. In this experiment, the users were given four tasks:

1. Find the name of the inventor who filed patents for both BBN and Xerox.
2. Find the name of the patent by Douglas Engelbart of Stanford Research Institute that was cited by someone working at Xerox
3. What two companies did a large amount of people who filed patents for Xerox also file patents for?
4. How many patents does John Seely Brown of Xerox have?

The first two questions were meant to test fine-grained navigation of links in the network. The third question required a global view of the network structure, and the last question tested the ability to find a node without making complex connections. The first experimental condition (**2-D**) let the user reposition nodes in the network in two-dimensions, but did not let the user extrude the data in the third dimension. The other experimental condition (**3-D**) let the user extrude the data into the third dimension. Our hypothesis was that for a given task, the use of the third-dimension would result in faster answers to the questions. The users were given each question and then timed until they got the correct answer. The task order was balanced to prevent ordering and learning effects. The results in seconds are displayed in Table 1. Table 1 also shows results of unpaired t-tests (with 6 degrees of freedom) comparing the means of the 2-D and 3-D conditions.

Only the results for Task 1 and Task 2 were statistically significant ($p < .05$). In both tasks the subjects answered the question more quickly using extrusion in the third dimension. Therefore, it can

Task	3-D (m ± st.dev) s.	2-D (m ± st.dev) s.
1 (t=3.0201, p= 0.0392)	82.96 ± 30.83	174.42 ± 52.14
2 (t= 3.2222, p=0.0485)	96.15 ± 56.85	362.10 ± 154.98
3 (t=0.4403, p=0.6751)	104.55 ± 43.80	126.24 ± 88.27
4 (t= 1.2847, p=0.2463)	41.95 ± 18.97	64.59 ± 29.72

Table 1: Mean Time to Solve Tasks in 2-D and 3-D

be concluded that in tasks involving navigating dense networks for fine-grained results that involve tracing connections between nodes, extrusion into the third-dimension is useful, while it may not be useful in making broad observations about the network or finding particular nodes, unless they are hidden in dense clusters. Qualitative feedback from the subjects showed that users found “that third dimension allows you to basically separate your data and pull out points of interest and file them away” as well as making it “easier to discern the edges and connectivity.”

7 Conclusion

In conclusion, our network visualization algorithm allows a user to intuitively and interactively navigate a network by transforming it themselves from a 2-D visualization to a 3-D visualization. Instead of automatically attempting to layout the network in three dimensions, we layout the network in two dimensions and then let the user, by selecting nodes, “extrude” various nodes of the network into three-dimensional space. This allows the user to organize the network intuitively by the structure, not just of the network, but of their own interests in the network.

Future work involves more detailed user-studies that compare both the user efficiency at discovering types of networks as well as a more thorough comparison of various initial conditions in the program, such as the initial algorithm used to layout the network in two-dimensions. A number of features are currently still under-development, such as the integration of the network visualization tool with a real-time search engine to allow the user to search for nodes using RDF, as well integrating a number of network-analytic tools into the user interface to help the user identify central nodes and trace in three-dimensions the strength of connections between nodes in the network. *Redgraph* is being modified in response to the qualitative feedback given by users, such as the use of “pop-up” menus to display two-dimensional data (text and images) and the ability to put this two-dimensional data in a “scratchpad” in one part of the immersive environment. Given the ease which ordinary users worked in three-dimensions using extrusion and the increasing amount of complex network-based Semantic Web data available on the Web, the utility of exploring of these networks in three dimensions bodes well for virtual reality.

Acknowledgements

To XXXX (<http://www.XXXX.org>) for allowing us to demonstrate at the first XXXX conference. Thanks to Kevin Webb for helping provide the patent data-set used in this experiment. The local CAVE-like system was funded by the National Science Foundation (XXXX).

References

BARABASI, A., AND ALBERT, R. 1999. Emergence of scaling in random networks. *Science* 286.

BOLLOBAS, B. 1985. *Random Graphs*. Academic Press, London, England.

CRUTCHER, L. A., LAZAR, A. A., FEINER, S. K., AND ZHOU, M. X. 1995. Managing networks through a virtual world. *IEEE Parallel Distrib. Technol.* 3, 2, 4–13.

CRUZ-NEIRA, C., SANDIN, D., AND DEFANTI, T. 1993. Surround-screen Projection-Based Virtual Reality: the design and implementation of the CAVE. In *Proceedings of SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH, Anaheim, CA, ACM.

EADES, P., HOULSE, M., AND WEBBER, R. 1997. Finding the best viewpoints for three-dimensional graph drawings. In *Proceedings of Symposium on Graph Drawing 1997*, 87–98.

FAIRCHILD, K., POLTROCK, S., AND FUMAS, G. 1988. Sem-Net: Three-dimensional graphic representations of large knowledge bases. In *Cognitive Science and its Applications for Human Computer Interaction*, R. Guidon, Ed. Lawrence Erlbaum, Hillsdale, NJ, 201–233.

FRUCHTERMAN, T., AND REINGOLD, E. 1991. Graph drawing by force-directed placement. *Software Practice and Experience* 21, 11, 1129–1164.

GANSER, E., AND NORTH, S. 1993. An open graph visualization system and its applications to software engineering. *Software - Practice and Experience* 30, 11.

HAYES, P., 2004. RDF Semantics, W3C Recommendation. <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>.

HENRY, N., AND FEKETE, J.-D. 2006. MatrixExplorer: A dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics* 12, 5.

KAMADA, T., AND KAWAI, S. 1989. An algorithm for drawing general undirected graphs. *Information Processing Letters* 31, 1, 7–15.

NEWMAN, M. 2005. Power laws, Pareto distributions and Zipf’s law. *Contemporary Physics* 46, 323–351.

PELLACINI, F., VIDIMČE, K., LEFOHN, A., MOHR, A., LEONE, M., AND WARREN, J. 2005. Lpics: a hybrid hardware-accelerated relighting engine for computer cinematography. *ACM Transactions on Graphics* 24, 3 (Aug.), 464–470.

SCHAEFFER, B., AND GOUDESEUNE, C. 2003. Syzygy: native PC cluster VR. In *Proceedings of Virtual Reality 2003*, IEEE, 15–22.

SCHNEIDERMAN, B. 1996. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of Symposium on Visual Languages 1996*, IEEE Computer Society Press, Los Alamitos, CA, USA, IEEE, 336–343.

VAN HAM, F., AND KRISHNAN, N. 2006. ASK-GraphView: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics* 12, 5, 669–676. Member-James Abello.