Tracking User Interactions Within Visualizations

Dennis P. Groth* Benjamin W. Murphy†

Indiana University School of Informatics

Abstract

We present a model and prototype system for tracking user interactions within a visualization. The history of the interactions are exposed to the user in a way that supports non-linear navigation of the visualization space. The interactions can be augmented with annotations, which, together with the interactions, can be shared with other users and applied to other data in a seamless way. The techniques constitute a novel approach for documenting *information provenance*.

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction Styles; H.5.4 [Information Interfaces and Presentation]: Hypertext/Hypermedia—Navigation;

Keywords: interaction, visualization, user tracking, provenance

1 Introduction

Visualization research is frequently presented in terms of a graphic image of a visual representation, along with a verbal description of what the observer should recognize. In addition to the visual results, the systems and operations performed against the data are reported. However, the process-related details and, in particular, the specific path through the visualization that lead to the discovery of interesting information is rarely reported.

We present a method for tracking the process of discovering information from visualizations. The method is based on a model of interaction that is based on directed graphs. The corresponding *interaction graph* articulates with user annotations and is implemented as an XML document that facilitates the sharing of what (and how) the user discovers from their data.[7]

This research is focused on process related activities of the knowledge discovery process [6] and the data visualization pipeline [12, 11, 3]. Research that specifically looks at the tracking of user interactions within a visualization environment include [9] and, more recently, [8]. These previous efforts are particularly relevant to our work at the conceptual level. Where we differ from them is in the capturing of meta-information, such as annotations, along with the interactions.

In a broader context, we consider this research to be a fundamental contribution to developing solutions for an emergent research area: *information provenance*. Problems in knowledge and data provenance[2, 1] are gaining interest, with broad applications to the advancement of scientific discovery [10].

Provenance is a term that refers to the lineage of an item. While some people associate the term with artwork, and the lineage of who owned, or possessed the piece, we use it in the context of the information discovery process. The model that we are presenting supports provenance by fully documenting the discovery process. The prototype demonstrates how users can interact with the history of interactions and capture annotations in the same context. Conceptually, the model separates the interaction from the data. This allows for the exchange of not only the result of a visualization, but precisely how the result was achieved. Another user may take the interaction data and use it against a different dataset, to see how general the technique may be.

*e-mail: dgroth@indiana.edu
†e-mail:bmurphy@indiana.edu

2 Interaction Tracking

The tracking model is based on directed graphs, with nodes signifying measurable states of the visualization system, and edges denoting transitions between the states. The states of the system are generically captured in the model, leaving it up to the implementation to define the specific contents of the state and transition information. For example, as described in [8], the transitions might contain discrete interactions, such as zoom, rotate, translate, or other interactions as described in [4]. Pictorially, the graph can be depicted as shown in the example in Figure 1.

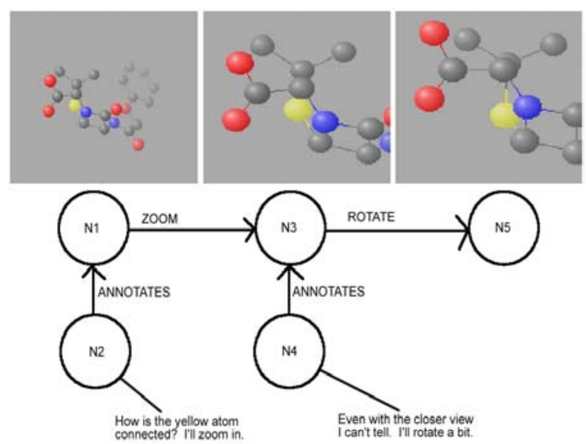

Figure 1: An example instance of an interaction model. In this exploration, the user zoomed and then rotated the data. Annotations reside in the graph as nodes with outgoing edges pointing to the state of the visualization system at the time of the annotation.

The interaction graph is exposed to the user as an additional panel, as shown in Figure 2. A closeup view of the interaction graph display is shown in Figure 3.

Each node contains the information necessary to set the state of the visualization, as well as user annotations. Note that the contents of the nodes may vary based on the visualization system, while the structure of the interaction graph is independent of the system. In the case of our prototype system, we record the 4×4 transformation matrix of the 3D space. This allows the system to be set instantaneously to any previous view in the interaction history.

It is worth pointing out that there is no restriction in the model for branching, or non-linear behavior represented in the interaction graph. However, in order for the model to support non-linear behavior tracking the model manager must keep track of where the

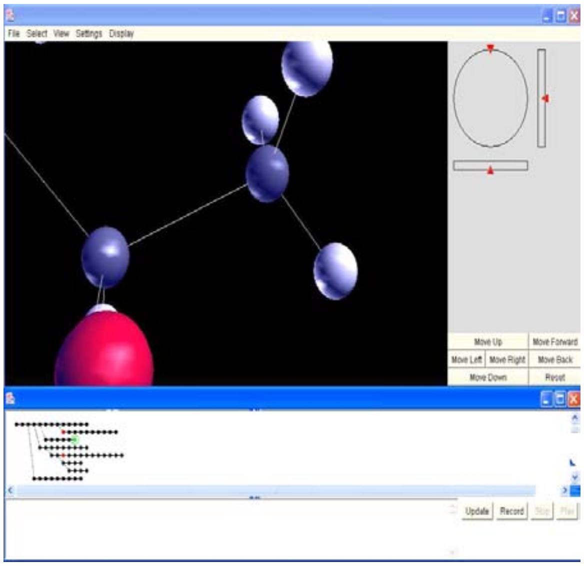

Figure 2: The prototype that shows the interaction graph to the user.

user is relative to the interaction graph. The prototype supports this feature by creating branches in the graph if the current state is not a leaf node.

Annotations are created through the interaction graph interface - not through the visualization system. This separation is further exploited in [5], in which distributed annotation occurs through tablet PC's allowing users to capture annotations in digital ink. At this time, we support three modes of annotation: typed text, voice, or digital ink.

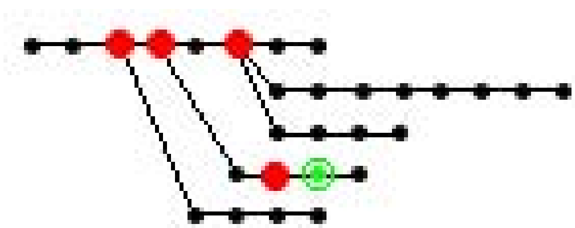

Figure 3: The user can select a node with the mouse to directly set the state of the visualization.

In addition to the direct manipulation features of the interaction graph user interface, we support a variety of data management capabilities. FOr example, users can save the graph to an XML file. Naturally, the user can load a saved interaction graph. What is most interesting about this capability is that the user can load graphs from previous explorations and replay them against different datasets. The files can be shared with other users in support of collaborative projects.

Because our prototype stores the transformation matrix we support interpolated tours of the space, in which the user selects a starting and ending location. Additional capabilities include user pruning of the graph. For example, the user can delete single nodes, prune entire branches, or collapse the the complete graph into a graph containing only those nodes with annotations.

3 Conclusion

Due to the highly interactive nature of visualization systems, a user cannot be expected to fully document each of their interactions. Consequently, there is a need for visualization systems to maintain user trace data in a way that enhances a user's ability to communicate what they found to be interesting, as well as how they found it. We have presented a prototype system that demonstrates how the history of interactions can be recorded automatically, and subsequently navigated. The approach improves the knowledge discovery process, and provides a capability for recording the provenance, or lineage, of information garnered from interactive visualizations.

Our future efforts include the development of collaborative interaction tools that track multiple users' explorations of visualizations. Such systems will support comparison of explorations, which will yield better understanding of how users search through visual representations. We are also working with chemists to focus on domain-specific implementations.

References

[1] Buneman, Peter and Khanna, Sanjeev and Tajima, Keishi and Tan, Wang-Chiew. Archiving Scientific Data. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 1–12, June 2002.

[2] Buneman, Peter and Khanna, Sanjeev and Tan, Wang-Chiew. Why and Where: A Characterization of Data Provenance. In *International Conference on Database Theory (ICDT)*, pages 316–330, 2001.

[3] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in Information Visualization:Using Vision to Think*. Morgan Kaufmann Publishers, Inc., 1999.

[4] Ed Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of InfoVis 2000 (Salt Lake City UT, October 2000)*, pages 69–75, 2000.

[5] Sean E. Ellis and Dennis P. Groth. A collaborative annotation system for data visualization. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*, pages 411–414, 2004.

[6] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. 1996.

[7] Dennis P. Groth. Information provenance and the knowledge rediscovery problem. In *Proceedings of the Internation Conference on Information Visualization*, 2004.

[8] Jankun-Kelly, T. J., Kwan-Liu Ma, and Michael Gertz. A model for the visualization exploration process. In *Proceedings of the IEEE Conference on Visualization*, pages 323–330, 2002.

[9] J. Lee and G. Grinstein. An architecture for retaining and analyzing visual explorations of databases. In *Proceedings of IEEE Visualization '95, Atlanta GA*, pages 101–109, October 1995.

[10] Chappell Alan R. Myers, James D., Matthew Elder, Al Geist, and Jens Schwidder. Re-integrating the research record. *Computing in Science and Engineering*, pages 44–50, May/June 2003.

[11] Chris North and Ben Shneiderman. Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In *Advanced Visual Interfaces*, pages 128–135, 2000.

[12] Helen Wright, Ken Brodlie, and Martin Brown. The dataflow visualization pipeline as a problem solving environment. In M. Göbel, J. David, P. Slavik, and J. J. van Wijk, editors, *Virtual Environments and Scientific Visualization '96*, pages 267–276. Springer-Verlag Wien, 1996.