Online Animated Graph Drawing for Web Navigation

Peter Eades, Robert F. Cohen and Mao Lin Huang

Department of Computer Science and Software Engineering
The University of Newcastle, NSW 2308, Australia
{eades,rfc,mhuang}@cs.newcastle.edu.au

Abstract. This paper describes an online animated graph drawing method. The method deals with huge graphs which are partially unknown. It is instantiated in OFDAV, a web diagram visualizer.

1 Introduction

Traditional graph drawing techniques assume that the complete graph can reasonably be represented in a readable and understandable manner on the display medium. However, there are important situations where this assumption does not hold. For example, it is unlikely that there is a readable and understandable technique to visualize the complete WWW on any display medium. Other examples include large file systems, and graphs arising in information retrieval.

This paper presents techniques to visualize very large amounts of relational information. We assume that the amount of data that can be effectively displayed at one time is only a tiny subset of the graph. Our aim is to provide tools to help humans understand large graphs while only being able to view a small portion of the graph at a time.

Most graph drawing systems to date approach this problem in the following manner:

1. Layout the graph on a virtual (very large) page.
2. Provide a small window and scroll bars to allow the user to navigate the visualization.

However, this technique has a number of major problems that impinge upon human understandability:

- The whole graph may not be known. In some cases, the local node in a distributed system may know only a part of the graph; in other cases, at the time of viewing only a small subgraph is known.
- The layout is predefined and views are extracted of this layout. This means that changing views is a geometrical operation and not a logical operation. The user naturally thinks in terms of the logical relations in the application domain (for example, hypertext links), not in terms of the synthesized geometry of the layout; thus navigation by scroll-bars is difficult.

- The approach is *structure orientated* in the sense that the information structure drives most aspects of the display. Note that in a Geographical Information System, the topology on the screen is closely related to the physical topology in the application domain. However, in graph drawing applications, the topology is synthetic and created by an algorithm with fixed aesthetic criteria. This problem is severe for large graphs because the user cannot see the whole graph. A user orientated approach is more suitable. The user should be able to control logical content of the display.

Further, a number of minor technical problems arise:

- long edges on the large virtual screen are hard to follow,
- it is difficult to navigate out of large empty areas, and
- it costs huge memory to store and display the large virtual screen.

Some alternative techniques have been proposed (see [12, 3, 8, 7, 10, 11, 5]). For example, *fish-eye* [12] views can keep a detailed picture of a part of a graph as well as the global context of the graph. Three dimensional methods, such as *cone trees* [10], increase the density of information on the screen. While these techniques effectively deal with graphs of moderately large size, they do not help where the graph (such as a WWW graph) is not completely known. Further, these techniques still predefine the geometry.

Our *Online Force-Directed Animated Visualization (OFDAV)* technique provides a major departure from traditional methods. Our aim is to seek a new visualization technique that can provide effective navigational views. Our technique does not need the whole graph to be known, it does not predefine the geometry (the user can navigate logically), and it is user-oriented.

In **OFDAV**, the view of the user is focused on a small subgraph of a large graph G at any point in time. This subgraph is defined by a *focus node* v. We use variations of traditional graph drawing algorithms to draw the subgraph of G and a logical "neighborhood" around this subgraph. We then allow the user to change focus node by selecting another node of G. We use multiple animations to guide the user between views and preserve the mental map [3]. We also adopt a linear "history" that traces the subgraphs that the user has visited. This assists in backtracking through the graph.

The technique is implemented as a system for navigating the WWW. **OFDAV** is a Java program. It uses a force-directed layout method [2].

A number of researchers have noted that "overview diagrams" [6, 9, 8] provide a reasonable solution to the famous "Lost in hyperspace" problem, where users become disoriented with respect to a complex system of hypertext links. Our system provides such overview diagrams. Some existing overview diagram systems have developed [9]; however, these systems all predefine the layout or geometry, and they only visualize the *history* within very limited context levels. In contrast of this, **OFDAV** provides an on-line browsing environment in which we can navigate through unlimited context levels.

2 The online graph model

In this section we describe the graph model and the transitions on which our system is based.

Our system aims to explore a huge graph $G = (V, E)$[1].

The exploration of the graph G uses a sequence of subgraphs $F_1, F_2, \ldots$; each subgraph in this sequence is a *logical key frame*. Intuitively, the logical key frame is the subgraph which is currently being viewed. For each i, there is a FIFO queue Q_i of the *focus* nodes of F_i. Intuitively, the queue Q_i consists of the nodes that have been visited during the exploration of G and the logical key frame F_i is the graph induced by nodes near Q_i (in graph theoretic distance).

To change from one logical key frame F_i to the next logical key frame F_{i+1}, the user selects a node $v_{i+1} \in F_i$ with a mouse click. The node v_{i+1} is appended to the queue, and (in a FIFO manner) a node is deleted from the queue. We have experimented with queue lengths between 7 and 10.

3 The graph drawings

For each logical key frame F there is an *animated graph drawing*. This is a continuous sequence of drawings called the *screens* of F; it plays the same role as "in-betweening" in traditional computer animation.

The change in logical key frames from F_i to F_{i+1} is triggered by a mouse click on the new focus node. The first screen of the first logical key frame is a graph drawing of F_1 computed by a modified force directed algorithm. The first screen for F_i is defined as follows. The nodes common to F_i and F_{i-1} stay on the screen as they were in the final screen of F_{i-1}. The new nodes appear radially around and very close to their neighbors in F_{i-1}.

The following screens of the animated graph drawing of F_i are computed using an animated force directed approach. In our implementation, the force directed algorithm is based on Hooke's law springs, and the strength of the springs varies. It tends to arrange the focus nodes in a line in the order that they appear in the queue. Details of the force model and the methods used to compute each screen are given in [4]. The algorithm is tuned to web graphs, which have a tree-like structure. However, in other applications, an application specific force directed method may be suitable.

4 Remarks

The **OFDAV** system is an effective tool for navigating through web documents. The layout of each logical key frame is a "spring" drawing and has the advantages and disadvantages of spring drawings (see [1]). However, it has other features which are particularly meaningful for visualizing large graphs which are partially unknown. Some of these features are listed below.

[1] In our implementation, the graph is directed, connected, and rather tree-like; but our methods apply in general to undirected graphs.

The mental map: A major problem in moving from one logical key frame to the next is preserving the user's mental map of the graph (see [3] for a full study of this issue). Our goal is to preserve the users mental map, while taking best advantage of the view screen. In **OFDAV**, we use animation to assist the user in understanding the change in view. We use three types of animation:
  - *Fade Animation*: We use shrinking/growing to help the user identify nodes that are disappearing/appearing.
  - *Camera Animation*: this moves the drawing so that the new focus node moves toward the center of the screen.
  - *Layout Animation*: We use a complex system of forces based on a Hooke's law springs.

Visualizing the history: The queue of focus nodes shows the recent history of the exploration of the graph. The force model that we use tends to keep this history in a line; the new nodes are added to one end of the line and the nodes which disappear are at the other end. This aids the user in keeping track of where they are and where they have come from. As well as the queue of focus nodes, **OFDAV** keeps a track of *past history*: a list of all previous focus nodes. An option in **OFDAV** is to show all the history.

Text with graphics: **OFDAV** optionally works in *Show-Document mode*: by clicking on *ShowDocument* button, the animation freezes and a mouse click on a particular node brings up the associated HTML document to the screen.

We believe that the principles of **OFDAV** can be used for any large relational system, and it is particularly useful for systems which are partially unknown.

5 Examples

The following three figures are screen dumps from **OFDAV**. All are from a visualization of the web context of the Department of Computer Science and Software Engineering at University of Newcastle. Figure 1 shows the initial key frame F_1, consisting of the focus node v_1 = 'Dept' and a set of nodes surrounding it with maximum distant $d_{init} = 2$ from node v_1. Figure 2 shows the following key frame, formed after a user mouse click on the 'PostStudents' node. Three frames further on we have Figure 3, formed by clicking on the nodes 'Mao', 'links' and then on the 'News' node.

References

1. G. Di Battista, P. Eades, R. Tamassia, and I. Tollis. Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry Theory and Applications*, 4(5):235–282, 1994.
2. P. Eades. A heuristic for graph drawing. *Congr. Numer.*, 42:149–160, 1984.
3. P. Eades, W. Lai, K. Misue, and K. Sugiyama. Preserving the mental map of a diagram. In *Proceedings of Compugraphics 91*, pages 24–33, 1991.

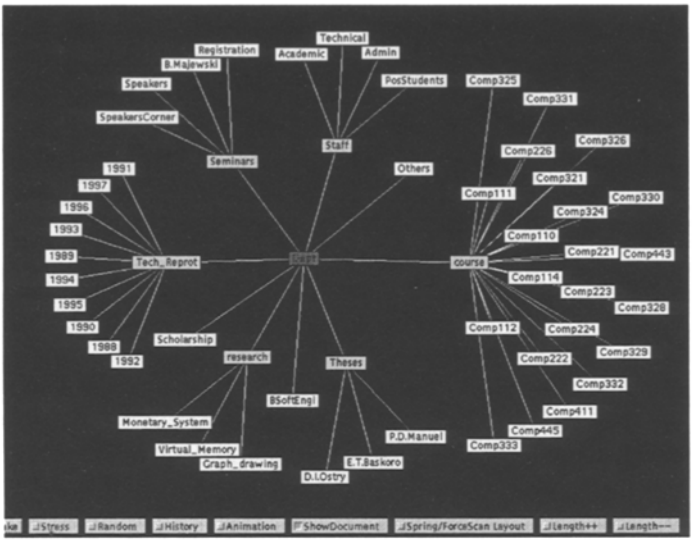

Fig. 1. A visualization of the web context of the Department of Computer Science and Software Engineering at University of Newcastle. This is the initial key frame F_1, consisting of the focus node v_1 = 'Dept' and a set of nodes surrounding it with maximum distant $d_{init} = 2$ from node v_1.

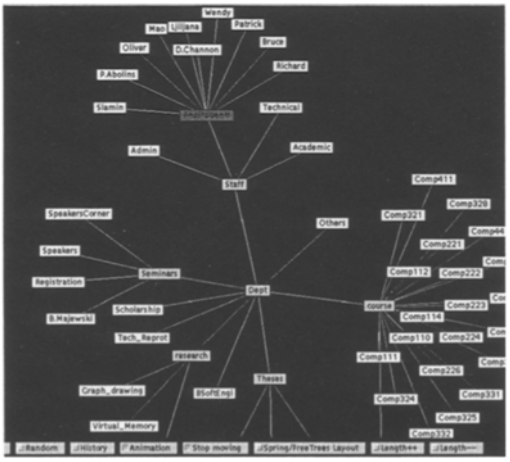

Fig. 2. Navigating the web context of the Department of Computer Science and Software Engineering at University of Newcastle. This is a frame F_2 following the graph in Figure 1. It has been formed after a user mouse click on the 'PostStudents' node, Here, v_2 = 'PostStudents' and the *recent history* consists of the nodes 'Dept' and 'PostStudents'.

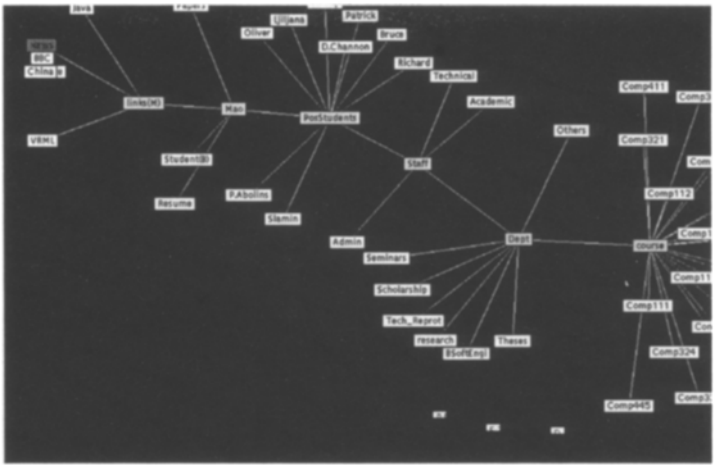

Fig. 3. Three frames further from Figure 2, formed by clicking on the nodes 'Mao', 'links' and then on the 'News'. Note that 3 old nodes are smoothly disappearing and 3 new nodes are smoothly appearing. Here, v_5 = 'News' and the *recent history* consists of the nodes 'Dept', 'PostStudents', 'Mao', 'links' and 'News'.

4. Peter Eades, Mao Lin Huang, and Junhu Wang. Online animated graph drawing using a Modified Spring algorithm. Technical Report 97-05, Dept. of Computer Science and Software Engineering, Uni. of Newcastle, 1997.
5. P. Kahn. Visual clues for local and global coherence in the WWW. *Communications of the ACM*, 38(8):67–69, 1995.
6. Sougata Mukherjea and J. Foley. Visualizing the World-Wide Web with the navigational view builder. *Computer Networks and ISDN Systems. Special Issue on the Third International World Wide Web Conference*, 27(6):1075–1087, 1995.
7. Sougata Mukherjea, J. Foley, and S. Hudson. Interactive clustering for navigating in hypermedia systems. In *Procceedings of the ACM European Conference on Hypermedia Technology*, 1994.
8. J. Nielsen. The art of navigating through hypertext. *Communications of the ACM*, 33(3):296–310, 1990.
9. Chris Pilgrim and Ying Leung. Applying bifocal displays to enhence WWW navigation. In *Procceedings of the Second Austrlian World Wide Web Conference*, 1996.
10. G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proc. ACM Conf. on Human Factors in Computing Systems*, pages 189–193, 1991.
11. George G. Robertson, Stuart K. Card, and Jock D. Mackinlay. Information visualization using 3D interactive animation. *Communications of the ACM*, 36(4):57–71, 1993.
12. M. Sarkar and M. H. Brown. Graphical fisheye views. *Commun. ACM*, 37(12):73–84, 1994.