Examining Edge Congestion

M.S.T. Carpendale, X. Rong
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada, T2N 1N4
{sheelagh,xingr}@cpsc.uclagary.ca

ABSTRACT
Many applications have emerged that attempt to visualize various aspects of web structure, links, and usage history. These visualizations often rely on graphs to provide a general view and reveal individual relationships. However, these relationships which are explicit in the graph structure are often not visually accessible due to edge congestion problems. In this paper we discuss edge congestion issues and present an interactive approach to handling them. Our edge-displacement algorithm temporarily adjusts the graph layout without distorting the user's mental map. Our algorithm opens up sufficient space to clarify relationship details without moving nodes at all; only edges are shifted.

Keywords
Navigation, web visualization, graph layout, distortion viewing

INTRODUCTION
The graph is a powerful tool enabling us to explore relationships between entities both mathematically and visually. As a result it is seeing increasingly common use in interfaces and visualizations in particular web and network visualizations. However graph layout is difficult and while much emphasis has been placed on creating layouts without edge crossings this is not always possible [4]. This is even more difficult when nodes have spatial meaning (e.g., cities in a map) and when the number of edges comes from real world situations. The fact is that while we gain a lot from the use of graph representations, edge congestion can make graphs difficult to interpret.

There are several problems related directly to edge congestion in graph layouts. One, edges can cut directly across a node interfering with the legibility of its contents (Figure 1(a)). Two, edges can be sufficiently dense to obscure whole sections of the layout (Figure 1(b)). And three, edges can pass closely by a given node causing confusion as to whether they are incident to that node or not (Figure 2(a)).

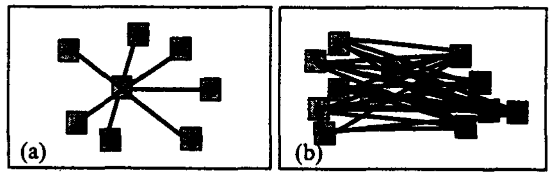

Figure 1: (a) edges crossing the face of a node, (b) edges obscuring a region of the graph

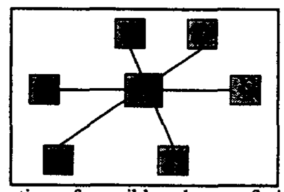
Figure 2: a situation of possible edge confusion, are these edges passing by the central node or incident to it?

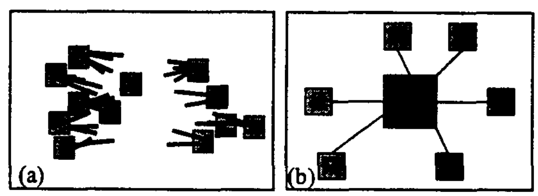

Figure 3: Previous approaches: (a) edges are shortened leaving 'edge pointers', (b) magnifying the node does not help

The legibility of the node in the centre of Figure 1(a) can be addressed simply by drawing that node in front of the edges that are crossing it (Figure 2(a)). However, this causes a situation three, one can no longer tell those edges that are connected to the node from those edges that are just passing by. At this point a graph's elegant ability to reveal relationships is compromised.

One previous approach interactively removed the centre section of the edges to clear some of the clutter and leave short 'edge pointers' that point at the node they would connect to [1]. While this does alleviate the congestion, interpreting graph relationships with these pointers is difficult. Another previous approach is to magnify the region in question with a detail-in-context variation [5,6]. This makes the central node larger but does nothing to relieve the edge confusion.

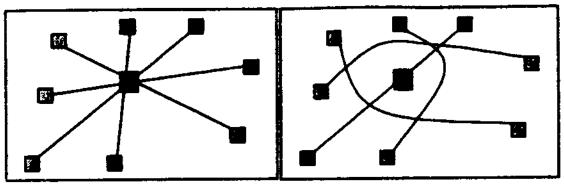

Figure 4: (a) edge confusion (b) relationships revealed through edge-displacement.

EDGE-DISPLACEMENT

Our approach introduces edge-displacement. While we use a distortion algorithm two important distinctions are made. No magnification is used. This is a displacement-only distortion algorithm [3,6]. Also some aspects of the data are not affected by the distortion. No nodes are displaced and the edges incident to a selected node are also excluded from the distortion. This leaves the basic structure of the graph layout entirely intact. Only edges that are passing by the region of interest are curved away from it. This allows the user to interactively clarify node and edge relationships without disturbing the basic graph layout (Figure 4).

Many different functions can be used to create the edge-displacement [3,6]. The illustrations here (Figures 5 and 6) use this formula: $rd = mag * (1 - (dp / lr))$ where: dp is the distance between a point on the edge that will be adjusted and the lens center (lc), lr is the lens radius, mag affects the magnitude of the displacement and rd is the resulting displacement.

This edge-displacement algorithm uses a distortion algorithm to create an effect not unlike a magic lens [2]. However, the use of a constrained distortion allows us to curve the edges away, creating no discontinuities at the boundary of the 'lens'.

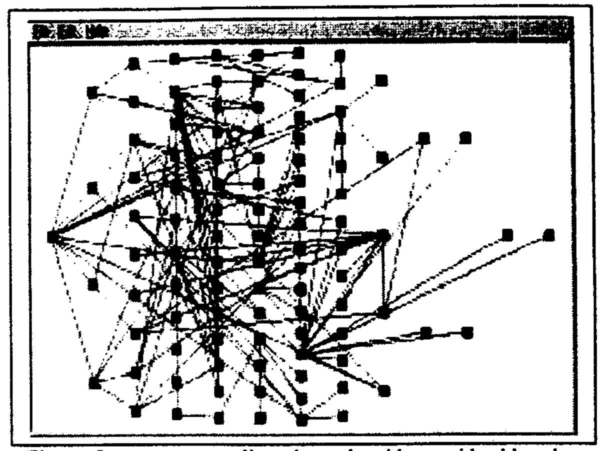

Figure 5: a more complicated graph with considerable edge clutter

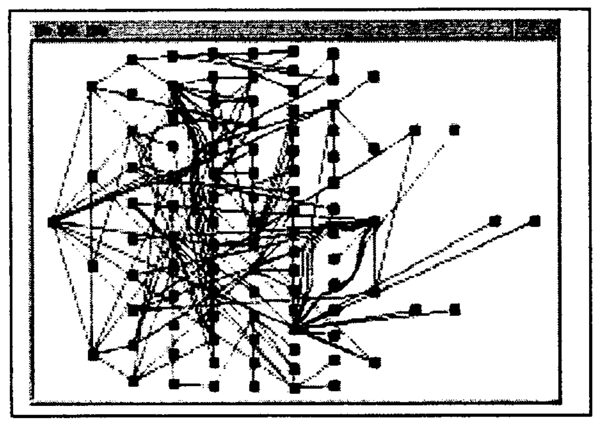

Figure 6: Two edge-displacement lenses are used to clarify selected regions

FUTURE DIRECTIONS AND CONCLUSIONS

We are studying various methods of interacting with edge displacement lenses. For instance, should a lens interactively follow the mouse or should the lens be placed and then removed? Should the amount of edge-displacement be in response to length of time a mouse button is depressed or to a pre-selected value? We are exploring various colour coding schemes; such as differently highlighting both the incident edges and the displaced edges and also fading the highlight colours with time.

We present a novel interactive method for exploring graphs. Our edge-displacement algorithm uses distortion to clarify chosen regions of the graph without altering the basic layout.

REFERENCES

1. R. A. Becker and S. G. Eick. Graphical methods to analyze network data. In *IEEE International Conference on Communications ICC'93*, 946–951, 1993.
2. A. Bier, M. C. Stone, K. Fishkin, W. Buxton, and T. Baudel. A taxonomy of see-through tools. In *Proceedings of the ACM Conference on Human-Computer Interaction: CHI'94*, 358–364, 1994.
3. M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Extending distortion viewing from 2D to 3D. *IEEE Computer Graphics and Applications*, 17(4):42–51, July/August 1997.
4. G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis. Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry: Theory and Applications*, 4:235–282, June 1994.
5. G. W. Furnas. Generalized fisheye views. In *Proceedings of the ACM Conference on Human-Computer Interaction: CHI'86*, 16–23, 1986.
6. Y. K. Leung, M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2): 126–160, 1994.
7. K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, 6(2):183–210, 1995