Efficient User Interest Estimation in Fisheye Views

Jeffrey Heer and Stuart K. Card
Palo Alto Research Center, Inc.
3333 Coyote Hill Road
Palo Alto, CA 94304 USA
{jheer, card}@parc.com

ABSTRACT

We present a new technique for efficiently computing Degree-of-Interest distributions to inform the visualization of graph-structured data. The technique is independent of the interest distribution used, and enables fluid interaction with very large data sets (over 100,000 nodes).

Keywords

Information visualization, focus+context, fisheye view

INTRODUCTION

Information visualization seeks to leverage human visual processing to make sense of abstract information. One particularly rich class of information structures ripe for visualization are those representable as graphs (i.e. nodes and edges), including organization charts, website linkage, and computer networks. One effective technique for the display of such data is the fisheye view [3], a focus+context approach that uses lightweight modeling of user interest to inform the display of information. User interest is modeled using a Degree-of-Interest (DOI) function, which assigns a single number representing the estimated relative interest of the user to each node in the structure. These numbers are used to appropriately layout and render the structure, for example by controlling which nodes are visible and which are elided (hidden). Figure 1 presents a segment of a fisheye visualization of a large directory of websites.

Both Furnas [3] and Card [2] describe the use of a DOI function. By convention, DOI values of 0 indicate maximal interest, with decreasing negative numbers representing correspondingly lower interest levels. The actual DOI distribution can be computed in any number of ways, though for standard browsing tasks on tree structures, the fisheye distribution [3] has proven quite effective. In tree structures, this value is a combination of the distances of a node from the tree root and any (often user selected) focal nodes.

To support natural and fluid interaction with human users, DOI-based visualizations must ensure that the calculation of each node's DOI value is done efficiently. The goal is to sustain continuous interaction without noticeable delays, leaving a window of at most 100ms [1] for the system to recompute the DOI distribution and node layout.

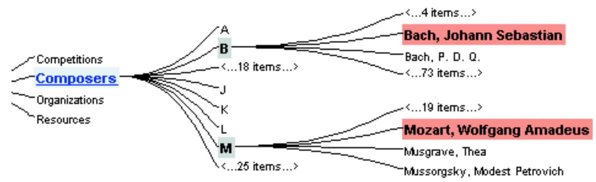

Figure 1 Segment of a multi-focal fisheye visualization of the Open Directory Project (dmoz.org)

Unfortunately, common implementations of this technique do not scale computationally. A straightforward approach to computing DOI is to start at the focal nodes and propagate interest out through the structure. Though simple, this technique runs in time proportional to the size of the structure. For large structures (100,000+ nodes) this introduces unacceptable delays in the user interface, limiting the scale and responsiveness of the visualization.

One can reduce calculation by exploiting the structure of the DOI function used. For example, during a change of focus in a fisheye distribution, the DOI estimates may not change for a large segment of the structure. As noted by Furnas [4], the nodes in need of updating in a tree structure lie within the subtree rooted at the nearest common ancestor of the previous and current foci. Unfortunately, the subtrees in need of updating may still be arbitrarily large, and in the worst case the whole tree may need to be evaluated. Furthermore, this scheme exploits knowledge of the DOI function being used, and so does not generalize to other possible interest distributions or more complicated topologies.

We describe a new general interest estimation approach, named *disinterest thresholding*, that instead scales with the number of visualized items and requires no information about the interest distribution used.

METHOD

Our approach is based on the realization that, from the perspective of the visualization, all elided nodes are equally uninteresting. If all elided nodes have the same computed DOI and this DOI is below the visualization threshold, the visualization will remain unchanged. This suggests the technique of disinterest thresholding: compute the DOI only for those items that will be visible. The DOI distribution is altered such that it saturates to a specified minimum DOI value below the visible threshold. This number is used as the default for all other, non-visible nodes. Figure 2 illustrates the difference between a normal fisheye DOI distribution on a tree compared to one generated using disinterest thresholding.

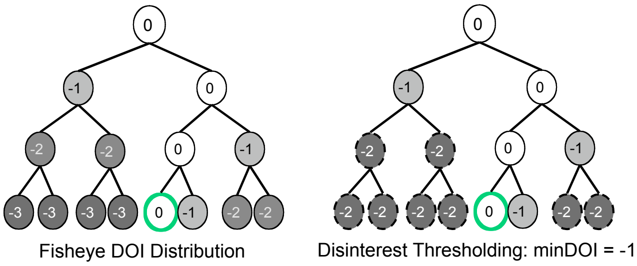

Figure 2 Fisheyes with and without disinterest thresholding

We have implemented disinterest thresholding using a backing array, reminiscent of a virtual memory page table, called the node attribute registry. This table serves as an attribute cache for the set of nodes considered sufficiently "interesting" for display. Each row of the table contains the non-structural attributes necessary for the display of a visible node, including the DOI values, (x, y) coordinates, node size, and node color. When an attribute is needed, the registry is consulted. If the node is in the table, the desired attribute is simply returned. If the node is not in the registry, a suitable default is supplied. In such cases, the DOI value returned is the saturated disinterest threshold, and the coordinates returned are those of the node's first visible ancestor, allowing newly visible nodes to naturally flow out from their parents in animated transitions.

index	dirty	DOI	x	y	size	color	etc…
0	1	0	213	12	5	…	…..
1	1	-1	134	58	4	…	…..

Table 1 Example Node Attribute Registry segment

Nodes are added to the registry, if not already present, when their DOI is set. Upon insertion, a node is assigned its index into the registry. Anytime a node's DOI is assigned, a dirty bit for the corresponding table row is also set. Upon completion of DOI computation, all dirty entries have their dirty bit cleared, and all non-dirty registry entries are marked empty, clearing unvisited entries. This process preserves the state of nodes that remain visible across transitions while freeing up space as nodes become elided.

EVALUATION

To evaluate the scalability of our technique, we ran a series of automated benchmarks using algorithmically-generated DOITrees [2]. We generated uniform trees of branching factor 8, varying the depth from 1 to 6. At each depth level we performed a series of simulated walks through the tree, each walk moving a single focus from the root down to a leaf node and back. A disinterest threshold of DOI = -2 was used. These experiments were performed on a 1GHz PIII machine with 256MB RAM and 16MB S3 video card. The generated trees were stored solely in physical memory.

Figure 3 graphs the time required for DOI calculation (in milliseconds) against the number of nodes, comparing the naïve, least common ancestor [4], and disinterest thresholding techniques.

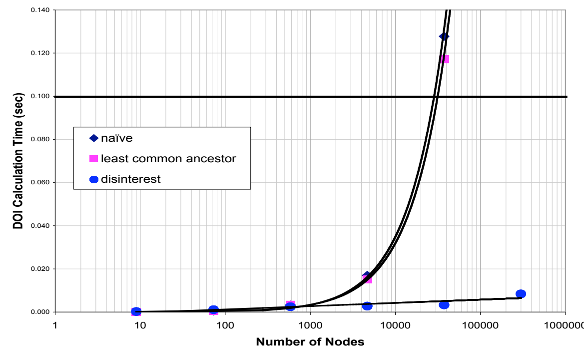

Figure 3 DOI Calculation Times

The plot shows that calculation time scales logarithmically for disinterest thresholding and linearly for the other techniques. Furthermore, disinterest thresholding completes DOI calculation under 20ms even at ~300,000 nodes. The other techniques cross the 100ms threshold at under 10,000 nodes. The key statistic governing the performance of disinterest thresholding is the number of visible nodes, which here is logarithmically related to the total node count.

CONCLUSION

In our experience, the disinterest thresholding technique has enabled fluid, lag-less interaction with structures on the order of half a million nodes. A few observations should be noted, however. First, as the technique scales with the number of visualized nodes, it is not appropriate for visualizations in which large numbers of nodes (i.e. over 10,000) are visible at once. Such cases can easily overwhelm the user's perceptual abilities, so the use of other visualization techniques may be advised. Second, responsive performance is also dependent on an efficient layout algorithm. We have used a tree layout algorithm that is linear in the number of visible nodes [2], but other useful layouts may carry higher computational costs. Finally, disinterest thresholding only approximates an interest distribution, and so should be used carefully if DOI values are to be used for purposes other than visualization.

The abundance of very large data sets requires efficient and scalable techniques for data exploration. Here we have described a technique for scaling common visualization approaches to large amounts of data. Future work will attempt to do even better, with the goal of supporting fluid interaction with millions and even billions of nodes.

REFERENCES

1. S. K. Card, T. P. Moran, and A. Newell, The Psychology of Human-Computer Interaction. Lawrence Erlbaum, 1983.
2. S. K. Card and D. Nation, "Degree-of-Interest Trees: A Component of an attention-reactive user interface," in Advanced Visual Interfaces '02, Trento, Italy, 2002.
3. G. W. Furnas, "The FISHEYE view: a new look at structured files," in Readings in Information Visualization: Using Vision to Think, S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds. Morgan Kaufmann Publishers, Inc., 1981.
4. G. W. Furnas, "Generalized fisheye views," in CHI'86