MoireGraphs: Radial Focus+Context Visualization and Interaction for Graphs with Visual Nodes

T.J. Jankun-Kelly*
Mississippi State University

Kwan-Liu Ma†
University of California, Davis

Abstract

Graph and tree visualization techniques enable interactive exploration of complex relations while communicating topology. However, most existing techniques have not been designed for situations where visual information such as images is also present at each node and must be displayed. This paper presents MoireGraphs to address this need. MoireGraphs combine a new focus+context radial graph layout with a suite of interaction techniques (focus strength changing, radial rotation, level highlighting, secondary foci, animated transitions and node information) to assist in the exploration of graphs with visual nodes. The method is scalable to hundreds of displayed visual nodes.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing Algorithms; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

Keywords: information visualization, focus+context, radial graph layout, graph drawing

1 Introduction

Graphs can be used to represent various types of data and information. Nodes in such graphs often posses information in addition to the topology. Numerous graph visualization methods have been developed to communicate this information to users quickly and effectively. While these methods have been successful, they have neglected a subset of graphs—graphs that posses nodes with visual elements such as images or renderable geometry. These *visual node graphs* require visualization methods which simultaneously display nodes and topology without losing the visual information at the nodes. *MoireGraphs*, named after the circular moire-like patterns formed from the layout algorithm, fulfill this purpose.

A MoireGraph displays a spanning tree induced upon a visual node graph using a radial focus+context graph layout introduced here. A focus+context method was chosen in order to provide a sufficient overview of the visual node graph without significant occlusion of the visual nodes or increase in required screen real-estate. In this case, the distortion applies to both the tree levels and the sizes of the visual nodes—the center focus visual node has the largest display area while outer levels and their contents have subsequently smaller areas. MoireGraphs are interactive and care has been taken to ensure this interactivity for visual node graphs of over a hundred nodes. The interaction methods allow quick exploration of the graph via highlighting of levels and secondary foci in addition to animated graph navigation. The new layout and interaction methods contributed by this research are discussed in more detail after previous work is examined.

*Department of Computer Science and Engineering, Mississippi State University, MS 39762. E-mail: `tjk@cse.msstate.edu`. This work performed while the first author was at UCDavis.

†Visualization and Graphics Research Group, Department of Computer Science, University of California, Davis, CA 95616. E-mail: `ma@cs.ucdavis.edu`

2 Related Work

2.1 Focus+Context Graph Visualization

Graph visualization has been extensively studied in information visualization (see Herman [Herman et al. 2000] for a survey). Of interest to this work are approaches which display both focus and context for graphs. There have been three main approaches in this area. The first approach distorts the graph after it has been laid out—"fish-eye" techniques fall into this category [Bartram et al. 1995; Carpendale et al. 1995b; Formella and Keller 1995; Keahey and Robertson 1996; Sarkar and Brown 1994; Storey et al. 1999]. The second method maps the graph onto a higher dimensional surface and then re-projects onto two dimensions [Carpendale et al. 1995b; Carpendale et al. 1995a; Kreuseler and Schumann 1999]. The final approach uses non-Euclidean geometry (such as hyperbolic geometry) during layout to "open up" the space [Lamping et al. 1995; Lamping and Rao 1996; Munzner 1997; Munzner 1998a; Munzner 1998b]. More rare are methods that use Euclidean geometry but have distortion effects in the layout of the graph or tree [Melançon and Herman 1998; Munzner et al. 2003; Teoh and Ma 2002] or only consider the topology of the graph to perform distortion during layout [Noik 1993]. An in-depth analysis of the different properties of these techniques can be found in Carpendale [Carpendale et al. 1997].

Each of the above approaches has its advantages and disadvantages. Fish-eye techniques are a post-process, and thus can be easily added to existing systems; unfortunately, rendering a full fish-eye distortion of the nodes *and* edges can be costly. The higher dimensional and non-Euclidean projection approaches are effective at the cost of implementation complexity. Distorted, Euclidean layout techniques, however, offer focus+context effects with less computational cost than fish-eye and lower implementation complexity than non-Euclidean techniques. For this reason, the work presented here explores a new radial layout technique with distortion. In addition, the focus+context technique is specifically tailored for displaying visual node graphs unlike most previous approaches. For example, though the Hyperbolic Browser [Lamping et al. 1995] can allocate extra room for nodes in which a visual element can be displayed, the area allocated does not necessarily remain consistent for nodes at the same tree level. This variation in size, coupled with the fact that nodes at the same level are spread throughout the display, makes comparison of visual elements at the same tree level difficult. This is not the case in a MoireGraph, since nodes at the same tree level share the same radial distance in the layout. The Hyperbolic Browser also does not support the wide range of navigation techniques that MoireGraphs support.

2.2 Radial Graph Visualization

Radial graph layouts were introduced by Eades [Eades 1992]; di Battista et al. [di Battista et al. 1999] also detail radial graph layouts. Graphs using a radial layout have two features: The focus node is at the center of the layout and nodes connected to the focus node radiate outward on uniformly separated rings—one ring for each level in the tree. The distance of a node from the focus determines the ring to which that node belongs. Construction of the layout proceeds from the center node in a breadth-first manner. In the traditional algorithm, each node is assigned a sector of its ring depending on the node's angular size and the size of all of its children. Children with larger subtrees are given more area than children with smaller subtrees. Though the original layout algorithm ensures the placement of children is concave, more recent applications for graph visualization (including this one) relax this constraint [Herman et al. 1999b; Wills 1999]. Variable size nodes and animation in dynamically changing radial graphs has also been discussed [Yee et al. 2001].

The layout algorithm described here extends previous radial graph work by including a deformation of the levels. Unlike previous approaches, the spacing between the levels is no longer uniform. Like Yee's work, graph nodes can have varying size. In this case, both radial spacing and node size decrease as the distance from the focus node increases. Using this technique, the focused element is highlighted while the rest of the graph remains in context.

2.3 Image and Document Browsing

Though the main focus of this work is not on image or document browsing, it is illuminating to compare MoireGraphs with these applications. Most document or image visualization systems either do not display a graph of the space, or do not display the documents/images when they do use a graph. For document browsing, the Document Lens [Robertson and Mackinlay 1993] is of interest because it uses a focus+context technique to display the document of interest in detail while compressing the rest of the document space. Image browsing systems also display the entire space of images either without hierarchies (the default for most file managers) or with some structural information provided by annotations ([Kang and Shneiderman 2000; Shneiderman and Kang 2000]). PhotoMesa [Bederson 2001] utilizes novel treemap algorithms within a zoomable user interface to provide similar structural browsing of photographs. All of these systems are well suited to document or image navigation based upon browsing the document/image space with perhaps some document/image relational information. This relational information, however, is not as apparent as in a MoireGraph—the links between visual nodes explicitly encodes connectivity. In conclusion, the purpose of the MoireGraph and these browsing systems can be seen as complimentary. The browsers give an overview of the entire document/image space while MoireGraphs highlight the relations among documents/images.

3 Radial Focus+Context Graph Layout

Given a visual node graph, a MoireGraph is constructed in three steps (Figure 1 depicts the structure of a MoireGraph). First, a focus is chosen. This focus will become the root of a spanning tree created from the visual node graph. Next, this tree is positioned using the new radial focus+context layout technique discussed below. Finally, the tree and its visual node elements are rendered. This last feature is vital—without it, a user would have difficulty building their mental map of the graph since it would require extra effort to associate the visual element of the node (e.g., an image) with a vertex in the graph (a position on the screen).

The radial layout deforms the level radii and node sizes depending on two factors: The depth of the spanning tree and a user controlled focus strength. Initially, the focus strength allocates a quarter of the display radius (half the size of the display area) to the focus node; this becomes the radius of a circle in which the focus node's visual element will be inscribed. The remaining length along the display radius is split among the spanning tree levels geometrically (see Figure 2a). Thus for all the nodes at level i, where $i > 0$ (level 0 is the root of the spanning tree), its corresponding radius r_{n_i} is:

$$r_{n_i} = (1 - f)r_d/2^i$$

where i is the level in the spanning tree, r_d is the display radius, and f is the focus strength (a value between 0 and 1). A geometric progression was chosen because it does not decrease the size the of the visible node too quickly. After all the node radii have been calculated, they are normalized in order to fit within the display radius (this is the source of κ in Figure 2a). For non-circular visual node elements, n_i is used as a bound for the size of the nodes at level i. For simplicity, node sizes smaller than a given threshold are clamped to that threshold. After the node sizes are calculated, the radius of each level is determined (see Figure 2b). The radius of the first level is the sum of the focus node's size and the node size for the first level. Subsequent level radii are the previous level's radius plus the node sizes for the current and previous levels. This radius determines the radial distance from the focus for all nodes on the given level.

Once the node and level radii have been calculated, the layout proceeds according to a radial layout algorithm. The angular position of a node is determined by four factors: The sector of the layout allocated to its parent (its *angular spread*), the node's angular spread, the total angular spread of its children, and the angular spread of its siblings (see Figure 2c). The larger of the node's angular spread (which is $2\arctan(r_n/r_{l_n})$ where r_n is the node's radius and r_{l_n} is the level radius of the node) and the total angular spread of its children is used as the node's effective angular spread. The angular spread of the parent is then divided proportionally among its children's effective angular spreads. For example, in Figure 2c, node n_k has more children than n_i and n_j and is thus given a larger share of its parent's angular spread. As described, the angular spread calculation occurs in two phases: First, the spreads are calculated bottom-up from the leaf of the spanning tree to the focus. Secondary, The sector division then occurs top down, starting with a 360° angular spread for the focus node. This layout is a relaxed version of the original radial layout proposed by Eades—it does not force child nodes to be within a concave area bounded by its parent's angular spread.

After the two stages of the radial focus+context layout, each node possesses a polar coordinate with respect to the focus node. The final rendering of the visual node graph then proceeds in level-order along the induced spanning tree. The focus node is always drawn on-top of other nodes (except secondary foci, see next section). If time-critical rendering is desired, such as during interaction or animation, the rendering can be halted during the level-order traversal. This is similar to the approach taken by Munzner [Munzner 1998b], and has the effect of eliding the nodes at the deeper (and less focused) levels of the spanning tree. Even though this may cause "popping" as some visual node elements are drawn, hidden, and drawn again, its effect is only temporary. In addition, the topological information (the edges) remain to provide continuity in the user's mental map of the graph.

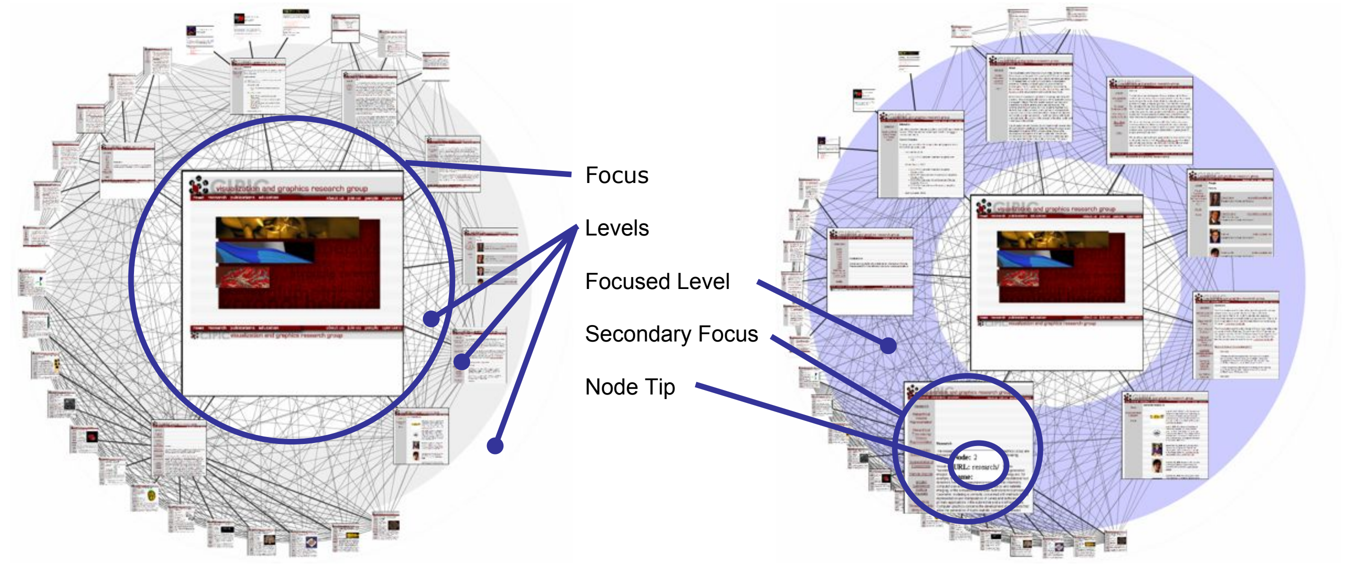
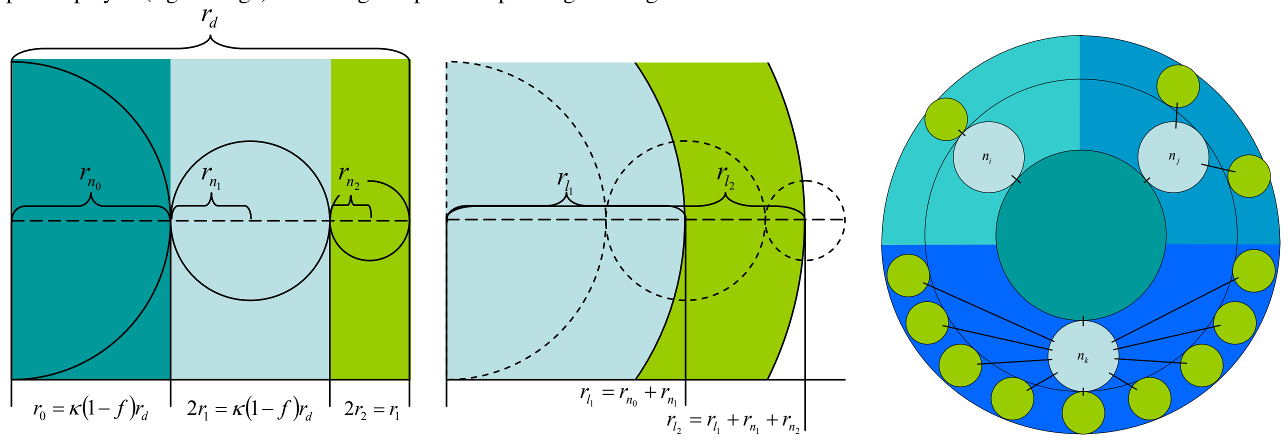


Figure 1: The structure of two MoireGraphs displaying web-site reachability. A MoireGraph radiates outward from the focus node, with each level in the underlying tree corresponding to a circular level in the display (left image). Each node in the tree has corresponding visual element (a rendering of the node's web page). As a user interacts with a MoireGraph, different levels or nodes in the display are focused upon, causing more screen space to be allocated to them (right image). If a user highlights a node for a long enough time, a node information tip is displayed (right image). Bold edges represent spanning tree edges.



(a) Node radii (b) Level radii (c) Angular spread

Figure 2: Layout calculations. To calculate the focus node's radius and thus its size, the focus strength (f), the display's radius (r_d), and a normalization factor (κ) is used (left image). Each subsequent node has half that radius. For level radii, which determine the radial distance from the focus of nodes on the level, the node radii from the previous and current levels are summed (center image). Finally, the angular positions of a node is determined by weighted division of its parent's angular spread (right image). The weight is determined by the nodes angular spread and the total angular spread of its children—nodes with more children receive more area than those with fewer.

4 Navigation and Interaction

The radial focus+context layout of a MoireGraph provides a static picture of the underlying visual node graph. To facilitate exploration of the graph, a set of interaction methods has been implemented. These range from highlighting methods for level or nodes to animated graph navigation. These methods are designed to support quick navigation of the graph, drill-down to specific nodes in the graph, and comparisons between nodes in the graph. The navigation techniques fulfill the same purpose as the navigational methods used by radial space-filling hierarchies [Stasko and Zhang 2000; Yang et al. 2003].

4.1 Changing Focus Strength

The default focus strength of a MoireGraph is suitable for a shallow spanning tree. As the number of levels increase, the width of each level proportionally decreases, shrinking the available space for each level's visual nodes. To offset this effect, the focus strength, the amount of the initial display area the focus receives, can be changed (see Figure 3). As the focus strength increases, the rest of the graph (the context) is pushed to the periphery of the layout. Decreasing the focus strength allows more detail to be given to the levels as the node size increases. It is possible at extreme ranges to either hide the focus' children completely or to effectively remove the focus node if so desired.

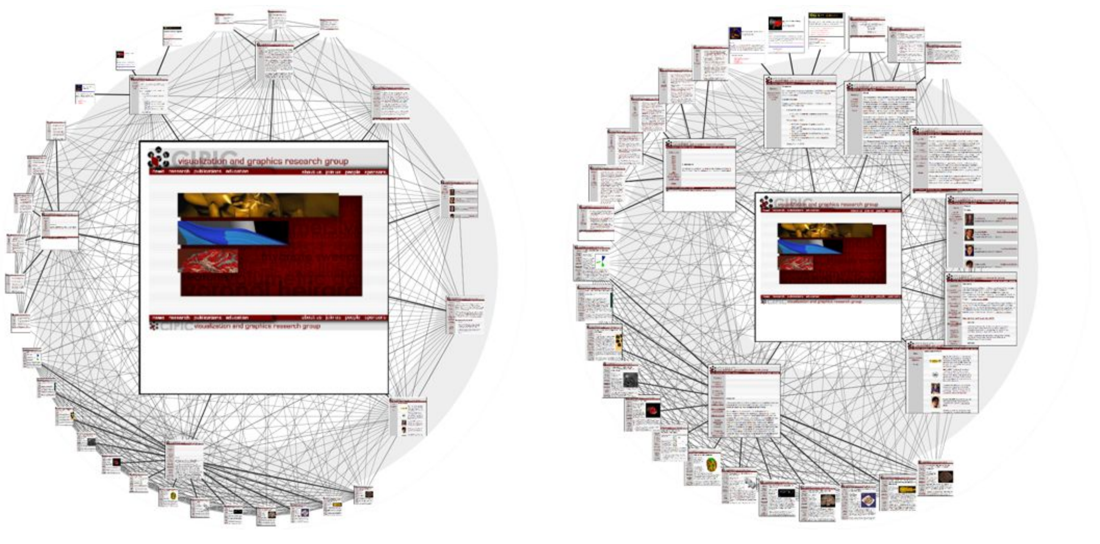
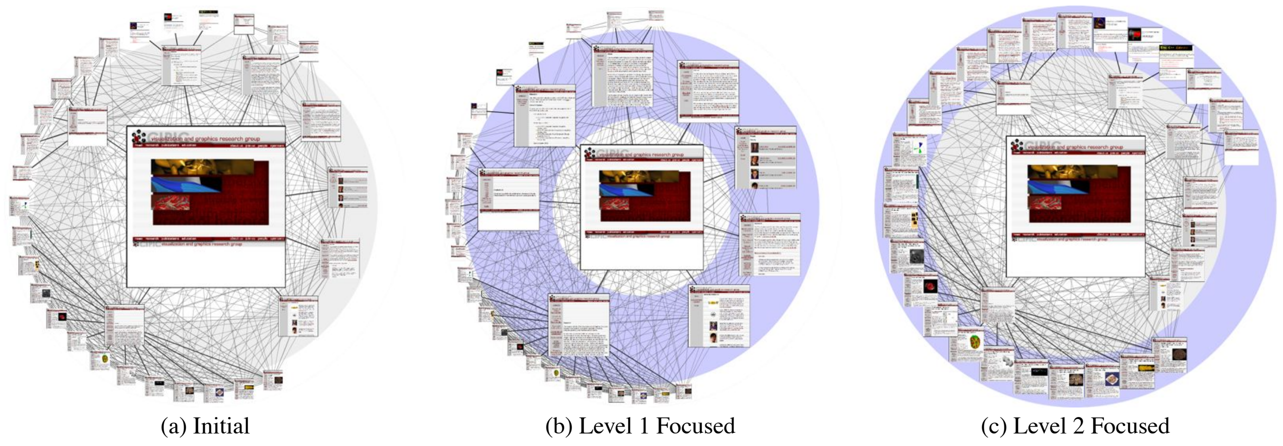
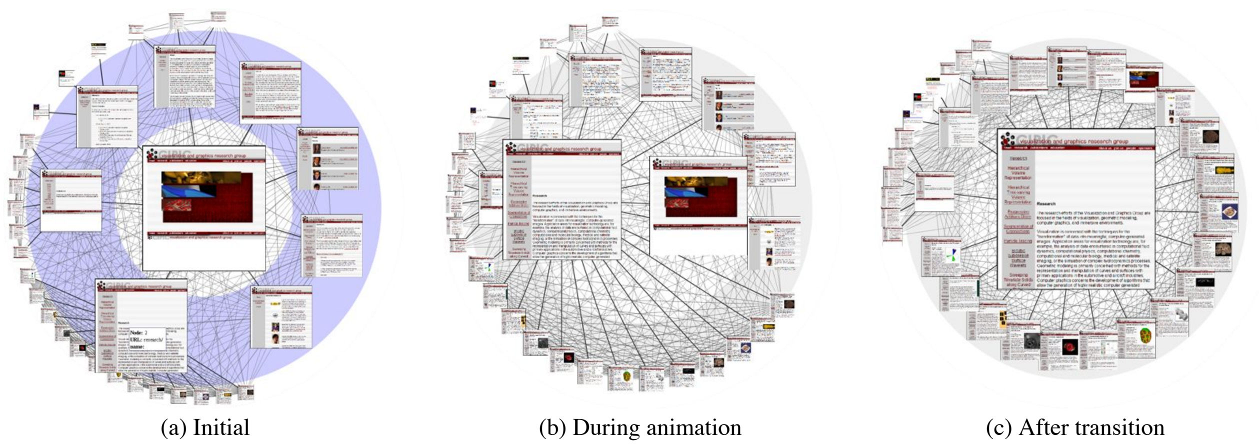


Figure 3: Changing the focus strength. As the focus strength increases, the rest of the graph is pushed to the periphery (left image). Conversely, as the focus strength decreases, more room is allocated to the focus' children (right image).



(a) Initial (b) Level 1 Focused (c) Level 2 Focused

Figure 4: Level highlighting. By highlighting a level in a MoireGraph, the space allocated to the level is increased to provide a more detailed look at the level's visual content.



(a) Initial (b) During animation (c) After transition

Figure 5: Animated Navigation. Selecting a node in a MoireGraph changes the focus. The angular coordinates of a node and the node's size are interpolated during the animation.

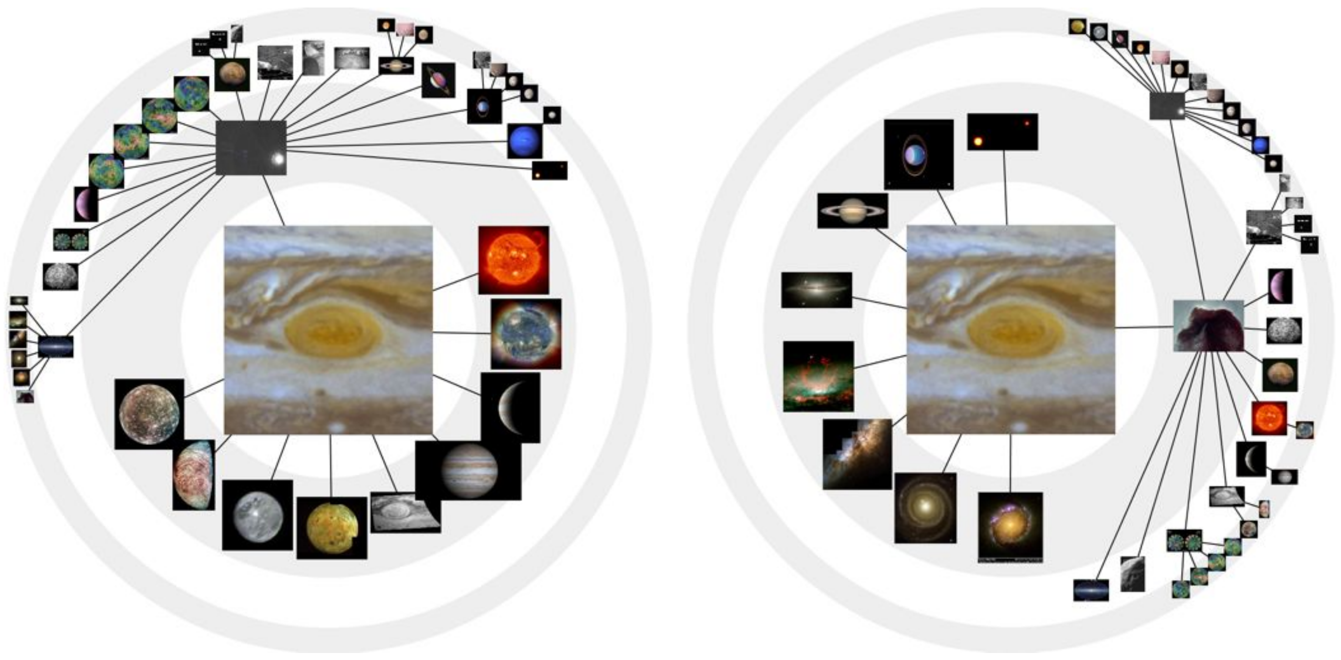

(a) Grouped by stellar hierarchy (b) Grouped by mission of origin

Figure 6: Image databases such as the NASA Planetary Photojournal posses several metadata attributes that group the same images differently depending on the metadata used. A MoireGraph can animate between displays of the different visual node graphs. In this image, only spanning tree edges are shown.

4.2 Radial Rotation

Besides controlling the focus strength, the user can also change the orientation of the graph by rotating the levels about the focus. This has the effect of offsetting the angular portion of each node's polar coordinates. This capability is useful in applications where the direction of an edge has significance. For example, in a visual node graph representing temporal information, the nodes could be rotated so that nodes to the left of the focus were from a previous time while those on the right were from a later time.

4.3 Level Highlighting

As the level depth of the MoireGraph increases, the corresponding available space for visual node rendering decreases. To facilitate the comparison of the focus node with nodes at different levels, levels can be highlighted during visualization (Figure 4). Highlighting occurs when the mouse pointer moves over a level other than the focus level; the level is then allocated twice the space it possessed originally. Consequently, the visual nodes of that level are displayed larger while those at other levels become smaller. As a visual cue, the highlighted level is also colored differently than the other levels. To avoid abrupt transitions, transitions from the non-highlighted to highlighted states (and vice-versa) are animated.

4.4 Secondary Foci

Level highlighting provides one method of comparison within an interactive MoireGraph. In addition, secondary foci can be chosen. A secondary focus is a visual node that has been selected for emphasis. For example, a secondary focus could be the result of a drill-down process searching for a specific node to examine. Another use of secondary focusing is to unocclude a specific visual node which has many siblings—secondary foci are considered "top-most" on any given level. Secondary foci are emphasized by increasing their size relative to their siblings on the same level (see Figure 5a for an example). Unlike level highlighting, choosing a secondary focus does not change the graph layout—the size of the node increases "in-place." Thus, secondary foci may temporarily occlude non-focused siblings; in our experience, this is acceptable for comparisons between the main focus and the secondary focus. In a MoireGraph, a secondary focus is chosen by moving the mouse over the visual node. Like level highlighting, transitions between different secondary foci is animated.

4.5 Animated Graph Navigation

Visual node graphs may contain numerous nodes. Thus, it is necessary to support graph navigation. Navigation is supported by the ability to choose new focus nodes. When a new focus is chosen, the display animates the transition between the old layout and the new layout. The animation uses the radial graph animation method suggested by Yee [Yee et al. 2001] (Figure 5): The polar coordinate of each node is interpolated between the starting and ending positions during animation. Like Yee's method, the orientation between the new focus and its parent is preserved to maintain the user's mental map of the graph. Since visual nodes change size during the transition if they change levels, the node size transition is also animated.

In some applications, there may be several visual node graphs associated with the same collection of visual nodes (Figure 6). For example, in the NASA Planetary Photojournal image database [NASA/JPL 2003], images may be associated by the place of their target in the galaxy (e.g., images of Jupiter would be children of images of the Sun) or images may be linked via what mission generated the image (e.g., images from the Hubble Space Telescope would be clustered together). A MoireGraph can animate between these two visual graphs using the same method for changing the focus node. In this case, the focus node would remain the same while the configuration of the rest of the visual node graph changes. In some cases, new nodes could be introduced or removed during graph changes. This is handled by introducing "phantom nodes" that are faded in or out as appropriate.

4.6 Node Information Tips

A visual node often contains more information than just the visual element associated with the node. Image databases with metadata attributes are a good example. To extract this information, node tips are supported in a MoireGraph. A node tip is a pop-up window that appears over a visual node that has been highlighted for a given period of time—they are essentially "tool-tips" for graph nodes. The node tip displays node metadata in a manner consistent with the given application. Thus, node tips act as a type of magic-lens for the visual nodes.

4.7 Display Properties

Currently, non-spanning tree edges in the visual node graph are not displayed by default; they may be drawn upon user request. To distinguish them from tree edges, non-tree edges are drawn thinner than other edges. In the future, it may be interesting to look at changing the rendering of the edges depending on the visual node graph's metadata or providing edge tips for edge information that are analogous to node tips for nodes.

5 Performance Considerations

To be effective, a visualization system must be responsive to user interaction. Thus, performance must be considered. Three factors affect the responsiveness of interactive MoireGraphs: The size of the original visual node elements, the quality of rendering, and the depth of the spanning tree during rendering. In the current implementation, no hardware acceleration is used in order to accurately gauge the performance impact of each factor.

The original size of the visual elements can have serious performance consequences. Without hardware accelerated mipmapping, large images can cause a performance hit during initialization (due to load times) and during interaction (due to the large amount of scaling being performed). Currently, thumbnail images are used in the display; high-resolution images can then be accessed in a separate window at the user's request.

Related to the size of the visual node elements is the quality of the visual element rendering, especially during interaction. While an animation is active, images do not have to be displayed at their full fidelity to ensure system responsiveness. This approach is standard in interactive scientific visualization systems. Currently, visual node elements are rendered using quick algorithms during animation. When the animation ceases, high quality rendering is used.

As the number of nodes in the spanning tree increases, so does the rendering time for that spanning tree. As previously mentioned, rendering occurs in level order in the spanning tree. To maintain responsiveness, this traversal is interrupted in an attempt to guarantee 10 frames-per-second performance during animation. In a more aggressive setting, the spanning tree calculation can also be terminated prematurely at a given depth or number of discovered nodes.

With the current implementation, a MoireGraph remains reasonably interactive for visual node graphs with a few hundred displayed nodes. For example, for the protein folding example with almost 300 nodes discussed later, performance ranged from 11 frames-per-second during animation with few nodes elided at 500x500 pixels to 7 frames-per-second with nodes in the outer two levels elided at 1024x1024 pixels (measurements were performed on a 2.65 GHz Pentium 4 with 1GB of RAM using no 3D-hardware acceleration). Though larger graphs are interesting to consider, there is a point where the size of the visual elements displayed becomes too small to show any detail. At this point, the node could be rendered traditionally (without the visual element) to provide a user with feedback on the structure of the graph without taxing the rendering system with pixel or sub-pixel sized visual elements.

6 Applications

The MoireGraph was developed to address the need for the visualization of graphs with visual elements. There are a wide class of graphs which posses this property. For the sake of demonstration, three are presented: web-site reachability, an image database, and results from a biochemistry optimization process.

6.1 Web-site Reachability

Figures 3-5 are examples from a web-site reachability visualization. In this case, the data consists of a subset of the first three levels of the UCDavis Visualization and Graphics Research Group's website (http://graphics.cs.ucdavis.edu/). An edge indicates a link from one page to another in the graph. The visual nodes in the example consist of renderings of the corresponding web page. This demonstration shows that many existing graph datasets such as web-site structure or file systems can also be displayed via a MoireGraph. For example, file structure could be displayed in a MoireGraph with the visual elements being the contents of the given file: Text files would contain the first few lines of text while binary documents would be rendered using an appropriate viewer.

6.2 Image Databases

Image databases are a natural application for MoireGraph visualizations. As an example, Figure 6 displays two MoireGraphs for 47 images from the NASA Planetary Photojournal. Each image has five pieces of metadata associated with it: The image's target, which stellar body is it orbits (if any), the mission which generated the image, the image's creation date, and the image credits. Two graphs, one for stellar structure and the other for source mission, were created from this data. Node tips in this graph display all the relevant image metadata. This MoireGraph balances the display of the images with the connectivity of the images. A real image database application could combine a MoireGraph with a traditional image browsing approach by animating between the two displays. This would combine the strengths of both approaches.

6.3 Protein Optimizer Steering

In a computational steering environment, a running simulation can be viewed and controlled from a remote workstation. ProtoShop [Crivelli et al. 2002] is an example of a CSE for biochemistry. ProtoShop provides a front end to visualize and manipulate proteins. These proteins are then submitted to an optimizing code running on a remote server. The optimizer tries to minimize the energy of the protein configuration by iteratively modifying the protein. At any stage in the process, a protein configuration may be waiting to be considered, being actively considered, retired after consideration, or culled if its energy is determined to be too high. Since new configurations are generated from old configurations, there is a natural tree structure of configurations. This structure can be visualized via a MoireGraph to gain insight into the process (Figure 7).

The simulation run profiled contains 284 configurations for the given protein. Initially, the optimizer started considering several configurations in parallel; their children formed a forest of configurations. To create a tree, a "dummy" node without any image was

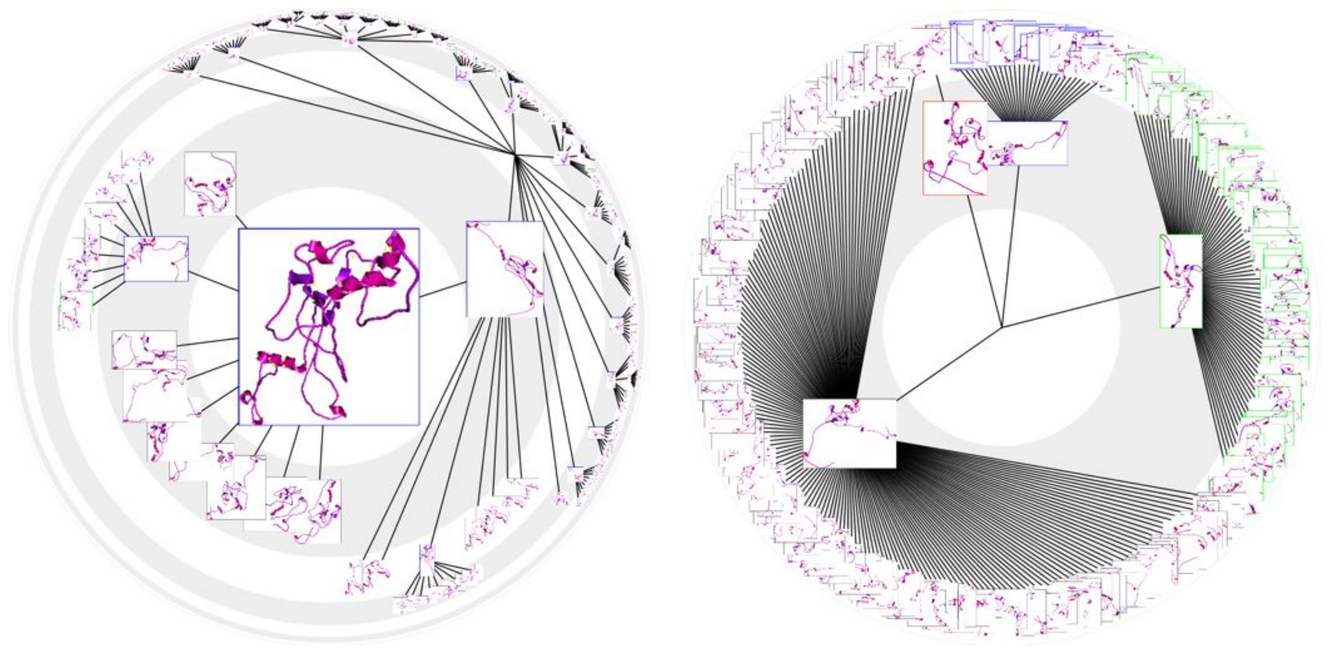


Figure 7: MoireGraphs for 284 configurations of proteins from an optimization process; the node without an image is a dummy node connecting the forest of configuration trees. As the optimizer progresses, more configurations are generated from a candidate configuration forming the configuration tree on the left. The right tree partitions the proteins into the current optimization status which is also encoded by their colored outline (right image): waiting (green), active (red), retired (blue), or culled (black).

inserted to connect these initial configurations. To indicate their status in the optimization process, nodes are outlined in green if waiting, red if active, blue if retired, and black if culled. If desired, the graph can also be clustered according to status. When highlighted, the node tip displays a configuration's unique identifier, its energy, and its status. This identifier can then be used in the CSE front end to access and manipulate that configuration. This example demonstrates how a MoireGraph can be used to augment another system, in this case by providing feedback on the progress of the optimization.

7 Conclusions

Graph visualization systems benefit from displaying visual nodes where appropriate—it lessens the user's cognitive load in associating the topology with the node's visual content. Applications using visual nodes graphs run the gamut from traditional file and web hierarchies to image databases. MoireGraphs provides an effective means of visualizing and exploring these kinds of graphs.

This work has made several research contributions. A new focus+context method for radial graphs has been presented. Though designed for visual node graphs, it could be applied in other domains. Several interaction techniques with radial graphs have also been described. These techniques could be extended to other graph visualization systems, be they focus+context or not. For example, secondary foci could be implemented in other focus+context systems for improved graph exploration.

7.1 Future Work

Extending previous focus+context graphs to perform layouts in the presence of visual nodes is one avenue of potential research. Traditional graph layout algorithms could also be extended to include space for visual nodes. Once complete, a user study would be useful in determining which layouts are better suited for different tasks.

Two modifications to the MoireGraph being considered are layout modifications and topology schematization. The layout currently uses a geometric progression of the levels. Others schemes, such as a Gaussian progression, could be interesting to implement and compare. Finally, as the graph size increases, the density of edges becomes very thick. Thus, it may be beneficial to add topology schematization in order to reduce visual clutter (such as in [Herman et al. 1999a]). The schematization would be introduced when edge density passes a threshold and would be removed when the density drops (such as during level highlighting). Similarly, instead of rendering a visual node element the same way as the distance from the focus increases, different representations could be used in a manner similar to semantic zooming [Bederson and Hollan 1994].

Acknowledgments

This work was funded by the National Science Foundation grants under contracts ACI-9983641 and ACI-0222991. The authors would like to thank Soon Tee Teoh for his input and Oliver Kreylos for providing the protein configurations dataset. We thank Prof. Head-Gordon, Prof. Byrd, Prof. Schnabel, Elizabeth Eskow, and Silvia Crivelli for supplying the global optimization code used to generate the protein optimization tree. The planetary images were taken from the NASA Planetary Photojournal [NASA/JPL 2003].

References

BARTRAM, L., HO, A., DILL, J., AND HENIGMAN, F. 1995. The continuous zoom: A constrained fisheye technique for viewing and navigating large information spaces. In *Proceedings of the ACM Symposium on*

User Interface Software and Technology, Information Navigation, 207–215.

BEDERSON, B. B., AND HOLLAN, J. D. 1994. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proceedings of the 7th annual ACM symposium on User Interface Software and Technology (UIST'94)*, ACM Press, P. Szekely, Ed., ACM SIGGRAPH/SIGCHI/SIGSOFT, 17–26.

BEDERSON, B. B. 2001. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, ACM Press, J. Marks, Ed., ACM SIGGRAPH/SIGCHI, 71–80.

CARPENDALE, M. S. T., COWPERTHWAITE, D. J., AND FRACCHIA, F. D. 1995. 3-dimensional pliable surfaces: For the effective presentation of visual information. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Information Navigation, 217–226.

CARPENDALE, M. S. T., COWPERTHWAITE, D. J., FRACCHIA, F. D., AND SHERMER, T. C. 1995. Graph folding: Extending detail and context viewing into a tool for subgraph comparisons. In *Proc. 3rd Int. Symp. Graph Drawing, GD*, Springer-Verlag, Berlin, Germany, F. J. Brandenburg, Ed., no. 1027, 127–139.

CARPENDALE, M. S. T., COWPERTHWAITE, D. J., STOREY, M.-A. D., AND FRACCHIA, F. D. 1997. Exploring distinct aspects of the distortion viewing paradigm. Tech. Rep. 97-02, School of Computing Science, Simon Fraser University.

CRIVELLI, S., ESKOW, E., BADER, B., LAMBERTI, V., BYRD, R., SCHNABEL, R., AND HEAD-GORDON, T. 2002. A physical approach to protein structure prediction. *Biophysical Journal 82*, 36–49.

DI BATTISTA, G., EADES, P., TAMASSIA, R., AND TOLLIS, I. G. 1999. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall.

EADES, P. 1992. Drawing free trees. *Bulletin for the Institute for Combinatorics and its Applications 5*, 10–36.

FORMELLA, A., AND KELLER, J. 1995. Generalized fisheye views of graphs. In *Proc. 3rd Int. Symp. Graph Drawing, GD*, Springer-Verlag, Berlin, Germany, F. J. Brandenburg, Ed., no. 1027, 242–253.

HERMAN, I., MARSHALL, M., MELANÇON, G., DUKE, D., DELEST, M., AND DOMENGER, J.-P. 1999. Skeletal images as visual cues in graph visualisation. In *Data Visualisation '99, Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, Springer-Verlag, E. Grller, H. Lffelmann, and W. Ribarsky, Eds., 13–22.

HERMAN, I., MELANÇON, G., DE RUITER, M. M., AND DELEST, M. 1999. Latour – A tree visualisation system. In *Proceedings of the Symposium on Graph Drawing '99*, 392–399.

HERMAN, I., MELANÇON, G., AND MARSHALL, M. S. 2000. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics 6*, 1 (January-March), 24–43.

KANG, H., AND SHNEIDERMAN, B. 2000. Visualization methods for personal photo collections: Browsing and searching in the photofinder. In *IEEE International Conference on Multimedia and Expo (III)*, 1539–1542.

KEAHEY, T., AND ROBERTSON, E. 1996. Techniques for non-linear magnification transformations. In *Proceedings of the IEEE Symposium on Information Visualization '96*, 38–45.

KREUSELER, M., AND SCHUMANN, H. 1999. Information visualization using a new focus+context technique in combination with dynamic clustering of information space. In *Workshop on New Paradigms in Information Visualization and Manipulation*, 1–5.

LAMPING, J., AND RAO, R. 1996. The hyperbolic browser: A focus+context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing 7*, 1 (Mar.), 33–55.

LAMPING, J., RAO, R., AND PIROLLI, P. 1995. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. ACM Conf. Human Factors in Computing Systems, CHI*, ACM, 401–408.

MELANÇON, G., AND HERMAN, I. 1998. Circular drawing of rooted trees. Tech. rep., Centre for Mathematics and Computer Sciences. http://www.cwi.nl/InfoVisu/papers/circular.pdf.

MUNZNER, T., GUIMBRETIÈRE, F., TASIRAN, S., ZHANG, L., AND ZHOU, Y. 2003. TreeJuxtaposer: Scalable tree comparison using focus+conext with guaranteed visibility. *ACM Transactions on Graphics 22*, 3.

MUNZNER, T. 1997. H3: Laying out large directed graphs in 3D hyperbolic space. In *Proc. IEEE Symp. Information Visualization*, L. Lavagno and W. Reisig, Eds., 2–10.

MUNZNER, T. 1998. Drawing large graphs with H3Viewer and Site Manager. In *Graph Drawing*, 384–393.

MUNZNER, T. 1998. Exploring large graphs in 3D hyperbolic space. *IEEE Computer Graphics and Applications 18*, 4 (July/Aug.), 18–23.

NASA/JPL, 2003. NASA Planetary Photojournal. http://photojournal.jpl.nasa.gov/.

NOIK, E. G. 1993. Layout-independent fisheye views of nested graphs. In *Proceedings fo the 1993 IEEE Symposium on Visual Languages*, 336–341.

ROBERTSON, G. G., AND MACKINLAY, J. D. 1993. The document lens. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST93)*, Visualizing Information, 101–108.

SARKAR, M., AND BROWN, M. H. 1994. Graphical fisheye views. *Communications of the ACM 37*, 12, 73–84.

SHNEIDERMAN, B., AND KANG, H. 2000. Direct annotation: A drag-and-drop strategy for labeling photos. In *Proceedings of the IEEE International Conference on Information Visualisation (IV'00)*, E. Banissi, M. Bannatyne, C. Chen, F. Khosrowshahi, M. Sarfraz, and A. Ursyn, Eds., 88–95.

STASKO, J. T., AND ZHANG, E. 2000. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Visualization 2000*, 57–64.

STOREY, M.-A. D., FRACCHIA, F. D., AND MULLER, H. A. 1999. Customizing a fisheye view algorithm to preserve the mental map. *Journal of Visual Languages and Computing 10*, 3, 245–267.

TEOH, S. T., AND MA, K.-L. 2002. Rings: A technique for visualization of large hierarchies. In *Proceedings of Graph Drawing 2002*, Springer-Verlag, M. Goodrich and S. Kobourov, Eds.

WILLS, G. J. 1999. NicheWorks — interactive visualization of very large graphs. *Journal of Computational and Graphical Statistics 8*, 2, 190–212.

YANG, J., WARD, M. O., AND RUNDESTEINER, E. A. 2003. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceedings of the IEEE Symposium on Information Visualization 2002*, 77–84.

YEE, K.-P., FISHER, D., DHAMIJA, R., AND HEARST, M. 2001. Animated exploration of dynamic graphs with radial layout. In *Proceedings of IEEE Symposium on Information Visualization 2001*, 43–50.