Multi-focus and Multi-window Techniques for Interactive Network Exploration

Priya Krishnan Sundararajan, Ole J. Mengshoel and Ted Selker

Carnegie Mellon University, Silicon Valley, CA, USA

ABSTRACT

Networks analysts often need to compare nodes in different parts of a network. Even a moderately-sized network on a screen shows coarse structure; unfortunately it may make detailed structure and node labels unreadable. Zooming in can be used to study details and read node labels, but in doing the network analyst may lose track of details elsewhere in the network. To address the problems, we present multi-focus and multi-window techniques to support interactive exploration of networks. This work supports the analyst in partitioning and selectively zooming in the network; data associated with each node can be further inspected using aligned or floating windows. Based on a user's selection of focus nodes, the network is enlarged near the focus nodes. The technique allows the user to simultaneously focus on up to 10-20 node neighborhood while retaining the larger network context. We demonstrate our technique and tool by showing how they support interactive debugging of a Bayesian network model of an electrical power system. In addition, we show that it can visually simplify comparisons for different types of networks.

Keywords: Computer Graphics,Methodology and Techniques,Interaction techniques

1. INTRODUCTION

Many datasets can be partly represented as networks. Even when visualizations of networks can fit to a computer screen, we may only see the global structure. The information associated with the nodes and edges are difficult to read with even a few hundred nodes. A Bayesian network provides an elegant method of representing complex relationships among random variables in the form of networks. It is a directed acyclic graph model representing a set of variables, in the form of nodes and their conditional dependencies, in the form of edges. Bayesian networks have become an important statistical machine learning tool and are being used effectively in uncertainty reasoning of many real world problems such as electrical power system diagnosis,[1] medical diagnosis, image recognition[2] etc. Interactively investigating these networks, with hundreds or thousands of nodes and edges, can be quite challenging.[2]

As an example of an application, consider a Bayesian network for medical diagnosis. In this case, nodes may represent diseases and symptoms. The edges represent the causal relationship between diseases and their symptoms. The conditional probability table (CPT) table for each disease node reflect prior probabilities, while a symptom node CPT has probability values for the different combination of states of its parent disease nodes. Given observed symptoms, also known as evidence, the network can be used to compute the posterior probability distribution for the presence of the diseases. Unlike other networks containing millions of nodes and edges, Bayesian networks found in most applications to date contain up to a few thousand nodes. In addition, the in- and out-degrees of Bayesian network nodes are quite small, typically a dozen or less, while some social networks nodes have thousands or millions of neighbors.

While software tools like Hugin[3] and GeNIe/SMILE[4] provide powerful visualization support for nodes, edges, and conditional probability tables (CPTs), many real-world Bayesian networks are becoming so large and complex that more advanced visualization and interaction techniques would be beneficial. Algorithmic Bayesian network analysis can be opaque, and existing visualization tools used for scrolling, zooming or panning the network do not elegantly combine network structure visualization with interactive exploration and understanding of node details. Even when merely moderately sized (say, with a few hundred nodes) Bayesian networks are zoomed to fit the screen, node labels may become unreadable. Consequently, an analyst may need to zoom, pan, and scroll in order to read node labels and thereby better understand the

Further author information:
Priya: E-mail: priya.sundararajan@sv.cmu.edu
Ole: E-mail: ole.mengshoel@sv.cmu.edu
Ted: E-mail: ted.selker@sv.cmu.edu

joint role of different network nodes in Bayesian network computations. Unfortunately, in the process of zooming, panning, and scrolling, analysts may easily lose context. As an example, given a node of interest, it is hard to understand interactions with its child or the parent nodes if some of them are located far-off in the network layout. It is also difficult for an analyst to keep values of the conditional probability tables (CPTs) in mind when studying and understanding zoomed-in details. Our multi-focus and multi-window visualization techniques enable a user to better compare and analyze internal details of nodes in different parts of a network while retaining network context.

The fisheye technique[5] was introduced to address the fundamental visualization problem of finding a *specific* address *node* in a address book *structure* of AT & T employees. The fisheye technique maintains context and lets users study details, but allows focus on only one part of a display. In many cases, however, a user might need to compare multiple things in multiple parts of a network. Remembering the details of a previously studied zoomed node becomes a tedious memory taxing and error prone activity as the number of nodes for comparison increases. This is the limitation of traditional single fisheye techniques.

We formulate a number of design goals (**DG**) to overcome the limitation of a single fisheye technique and to address our requirements to help users to analyze Bayesian networks interactively. For the zooming algorithm, we integrate goals specified by Reinhard et al. in[6] as below and the suggestions from Bayesian experts. We hypothesized that the inherent complexity in Bayesian representations could be overcome with visualization tools that focus on comparing parts of the network and their contents.

1. **DG1** *Multi-focus zooming*: Multiple focus nodes selected at different parts of the network should be zoomed simultaneously in a user controlled manner and they should be readable regardless of their location in the network. The process of zooming in and zooming out should be animated to avoid hard to follow abrupt layout changes.

2. **DG2** *Topology Maintenance*: The continuity of layout adjustment should not challenge the user's mental map of the structure of the network, that is the topology, and the proximity relations of the nodes should be maintained.[7]

3. **DG3** *Focus nodes selection*: The user should be able to determine which nodes to zoom by studying the details of all the nodes from a dataview window and select the nodes that show an interesting behavior. The user should also be able to study a similar set of nodes by using a search option or a particular section of the network by using a group selection option.

4. **DG4** *Scoped zooming*: The zooming should be restricted to a region of interest or a partition, around the focus node to avoid distorting the whole layout. A partition algorithm should help the user to intuitively decide on the surrounding area around the focus node for zooming. Set of nodes in the network should be partitioned. There is one partition per focus node. With no focus node, there is one partition for all the nodes. The partitions around the focus nodes is used to localize the zooming effect within that region.

5. **DG5** *Dynamic partitioning*: The partitioning algorithm should dynamically adjust the partitions as the user adds or removes the focus nodes for zooming.

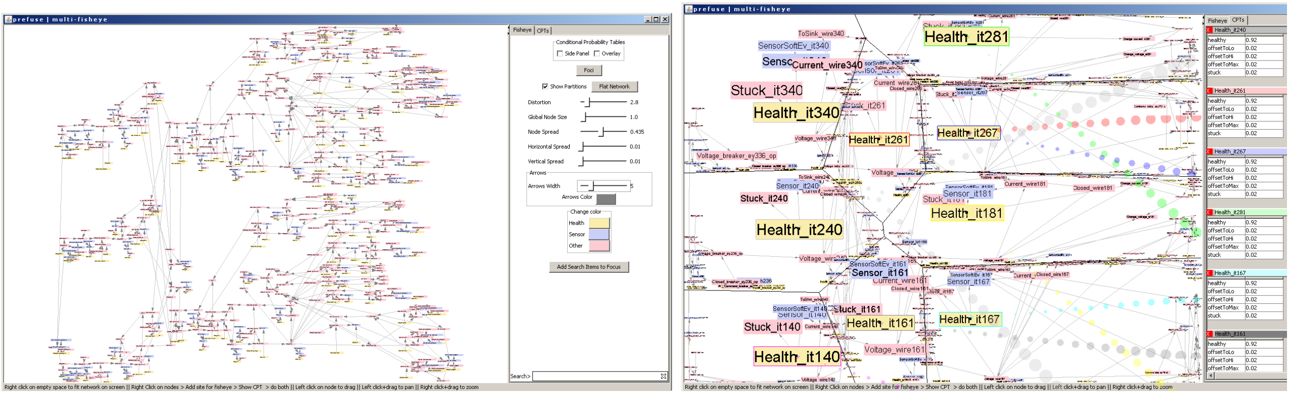
6. **DG6** *Continuous partition regions*: After zooming, discontinuities or 'ghost regions' should not exist between the partitions. In other words, the zooming should be smooth across all the regions.

7. **DG7** *Context visibility*: The visibility of the neighborhood nodes around the focus node should be user controllable.

8. **DG8** *Label zooming*: The node labels of the focused nodes should be bigger and easily readable. The level of detail decreases as we move away from the focus node.

9. **DG9** *Data exploration*: Multiple windows that can simultaneously open, should allow users to study focused nodes in more detail. Depending on the type of network, the data-level windows can show node details such as CPTs, time-series graphs or an individual's bio-data for comparison or data analysis.

Existing visualization techniques do not integrate the above design goals. We introduce a multi-focus technique to help analysts compare different parts of a network simultaneously while retaining network structural context and especially to make the Bayesian analysis more effective in larger networks. Our novel approach satisfies the above goals and supports visualization principles overview first, zoom and filter, then details-on-demand.[8]



Conditional Probability Tables
☐ Side Panel ☐ Overlay
Foci
☑ Show Partitions
Flat Network

Distortion 2.8
Global Node Size 1.0
Node Spread 0.435
Horizontal Spread 0.01
Vertical Spread 0.01

Arrows
Arrows Width 5
Arrows Color

Change color
Health 🟨
Sensor 🟦
Other 🟥

Add Search Items to Focus

Right click on empty space to fit network on screen || Right Click on nodes > Add site for fisheye > Show CPT > do both || Left click on node to drag || Left click+drag to pan || Right click+drag to zoom

Health_it240	
healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it261	
healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it267	
healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it281	
healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it167	
healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it161	
healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02

(a) Before fisheye zooming. Node labels are unreadable in this view. (b) After fisheye zooming. Node labels are readable in this view.

Figure 1: Visualization of multi-variate probability distributions of ADAPT electrical Bayesian network.

Our multi-focus technique starts with a base network as shown in Figure 1a. It allows analysts to select a set of focus nodes to zoom-in n multiple parts of a network. In order to minimize distortions while zooming, we decided to localize a fisheye-like effect to a region surrounding a focus node. The distortion neighborhood is defined by partitioning the display into polygonal regions, using a Voronoi algorithm[9] giving an optimal layout for partitioning. All the nodes inside a polygonal region are closer to the focus node in that region than to the focus nodes in other regions. A localized fisheye zooming is then applied to those partition regions, several such zoomed partitions can be created at the same time allowing users to zoom a portion of the network without losing structural context (see Figure 1b). Our analysis and visualization is targeted on two tasks. The construction task focuses on navigation in which various nodes are zoomed for inspection and the diagnosis task in which causal nodes that lead to a particular outcome are selected for an in-depth comparison and analysis.

The remainder of the paper is structured as follows: Section 2 provides an overview on previous related work, while the multi-focus algorithm is described in details in Section 3. Application and expert evaluation are discussed in Section 4. Finally, Section 5 concludes the report with some hints for future research directions.

2. RELATED WORK

A number of existing techniques that partially fulfill the design goals outlined above. This section discusses how visualization, visual distortion, multi focus approaches and interactive visualization approaches might each improve on each other's ability to understand and solve problems in networks such as Bayesian networks.

2.1 Network based fish-eye techniques

A number of focus+context display techniques have been introduced.[10] A powerful and popular way to retain overview and detail is the fisheye approach.[5] In a fisheye view, the entire node structure is always visible. The user-selected node and its neighboring network nodes are magnified and distant nodes in terms of graph structure are demagnified. The fisheye techniques described by Sarkar and Brown[11] use filtering and distortion, but support focus on a single item and distort the whole layout. Lamping et al.[12] present an improved spatial distortion approach for the fisheye focus+context transformation for hierarchical data structures for shortening the spatial layout with hyperbolic geometry. It places nodes around the root and provides smooth and continuous animation of the tree as users click or drag nodes to read the focus point of the layout. This approach also distorts the whole layout for each selection of focus point. In these techniques, some of our design goals are not satisfied such as multi-focus zooming (DG1), topology maintenance (DG2), focus nodes selection (DG3), scoped zooming (DG4), dynamic partitioning (DG5) and data exploration (DG9). The design goals that are satisfied are continuity in layout (DG6), context visibility (DG7) and label zooming (DG8). However the topology maintenance (DG2) is achieved in the Furnas's fisheye approach[5] but not in the hyperbolic approach.[12]

Sarkar et al.[13] also investigated a two focus approach with orthogonal and polygonal stretching, where the simple orthogonal distortion method maintains orthogonal ordering of points (nodes maintain their left-of, above, etc. relationships), but polygonal method does not. The zooming action is such that the user acts indirectly on the focus nodes through a ‘rubber sheet.’ This technique does not support the continuous zooming (DG6). Formella and Keller[14] distort external node layout to make space to magnify all points within circumscribed polygons. This simple distortion mechanism does not scale to large networks as the user has to manually select the focus area by using a rectangular selection. Both these techniques do not support dynamic partitioning(DG5).

A topological fisheye method[15] precomputes coarsened graphs and renders the level of detail from the combined graphs, depending on the distance from one or more foci. This system supports more than one focus. The drawback, however, is the computation involved in pre-computing the coarsened graphs making it unsuitable for interactive exploration. Dynamic insets[16] uses the connectivity of the graph to bring offscreen neighbours of on-screen nodes and their context into the viewport as insets. However the entire structure of network is not retained in this technique. Feng et.al. recently introduced a multi-focus and context technique[17] for generating spatiotemporal coherent time-varying graphs. This technique utilizes a triangle mesh to partition the graph nodes and leveraged this underlying mesh for constrained multi-focus and context visualization. This was achieved through formulating an energy function for optimized deformation. This approach supports design goals such as the topology maintenance (DG2), continuity in layout (DG6) and context visibility (DG7). But it does not support design goals such as multi-focus zooming (DG1), focus nodes selection (DG3), scoped zooming (DG4), dynamic partitioning (DG5), label zooming (DG8) and data exploration (DG9). Other zoom algorithms[18] [19] provide more scalable multi-focus distortions, but without scoping of distortion, any change in these approaches affects the entire network layout. However, the user has no direct control over the sizes of nodes aside from opening or closing them.

All the above techniques do not support our design goal, the scoped zooming (DG4) as they distort the whole layout when rendering graphs at different levels. While the issue with distorting the whole layout is addressed by Reinhard et. al. in their improved fisheye zoom algorithm,[6] it results in wasted screen spaces called ‘ghost regions,’ thus do not support continuity in the layout (DG6).

2.2 Tree based fish-eye techniques

Several systems demonstrate focus+context techniques for tree visualization, like SpaceTree,[20] which uses extensive zooming animation to help users stay oriented within its focus+context tree presentation. This technique does not support focus nodes selection (DG3), scoped zooming (DG4), dynamic partitioning (DG5), and continuous partition regions (DG6). The TreeJuxtaposer[21] technique uses focus+context methods to support comparisons across hierarchical datasets. The technique supports most of our design goals except DG6, the continuous partition regions. As this technique allows the user to do a rectangular selection which usually gives rise to ‘ghost regions,’ rendering discontinuity in the layout as well. These techniques require aggressive space constraint methods if the nodes of comparison are on the farther ends of the tree structure.

Tu and Shen present the ‘balloon focus,’ a multi-focus context technique for treemaps. Their user study confirms that users prefer the multi-focus treemaps to explore query results.[22] While the treemaps provide good usage of the available space, network structure can be difficult to identify.[23] So this approach does not retain the topology of the network (DG2). Bayesian networks are in general not trees, making tree visualizations too limiting.

2.3 Image based fish-eye techniques

Elmqvist et al.[24] propose a new distortion technique that folds the intervening space to guarantee visibility of multiple focus regions. The folds themselves show contextual information and support unfolding and paging interactions.[24] A drawback of this technique is the lack of user control over the scope of the focused regions. Non-linear magnification,[25] pliable surfaces[26] and CATGraph,[27] when applied to graphs, distort the labels within the zoomed areas. Such distortions make labels difficult to recognize or read. Blending multiple zoomed regions has not been solved in previous works and involves a separate computational step. These techniques support multi-focus zooming (DG1), topology maintenance (DG2), dynamic partitioning (DG5), layout continuity (DG6), and context visibility (DG8). But it does not support focus nodes selection (DG3), scoped zooming (DG4) (as the whole layout is distorted for each selection of focus point), label zooming (DG8) and data exploration (DG9).

3. MULTI-FOCUS ZOOMING ALGORITHM

Our multi-focus zooming algorithm helps to retain the network structure thus limiting distortion to preserve the user’s mental map of the Bayesian network. The distortion algorithm is independent of the layout algorithm and is defined as a

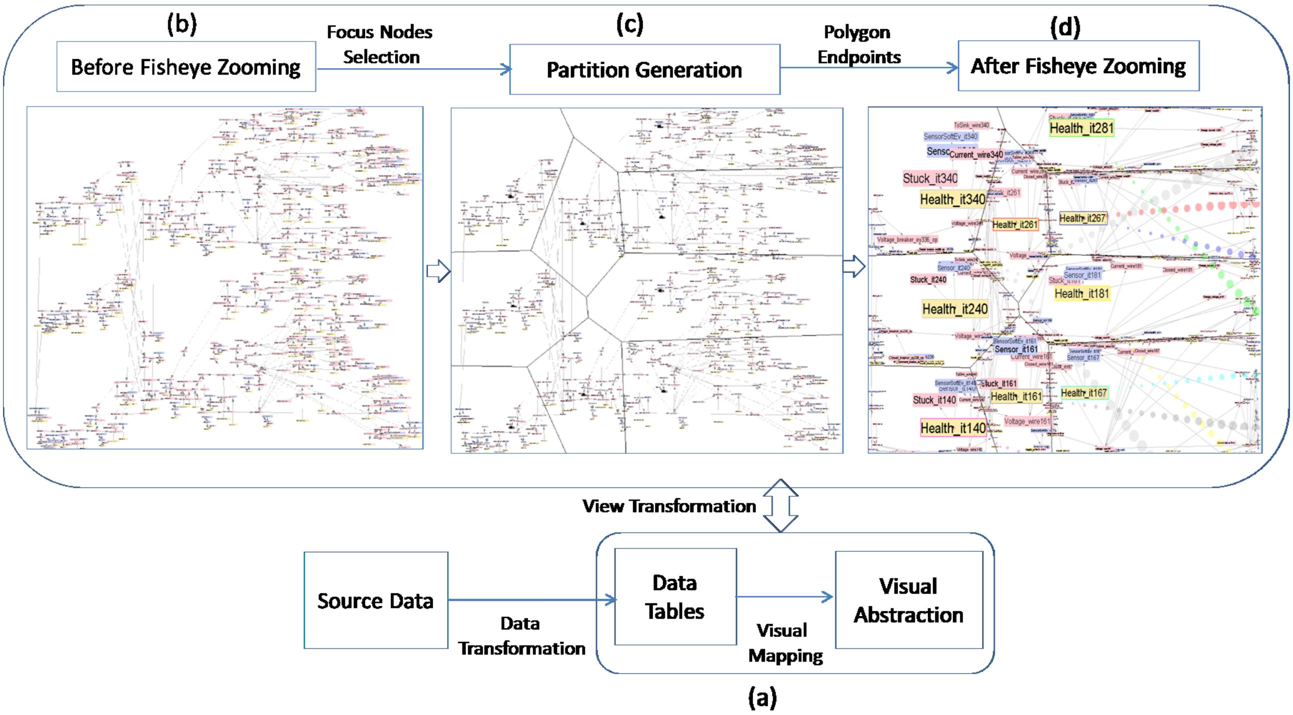


Figure 2: The diagram depicts the pipeline flow in the multi-focus algorithm. (a) The source data is loaded into Prefuse data tables and converted to visualizable attributes in visual abstraction. The view transformation takes place in three steps (b) before fisheye zooming (c) partition generation and (d) after fisheye zooming.

separate processing step on the layout of the graph. This allows for a modular organization of software[28] and helped us to easily understand, modify and reuse existing code to suit our visualization. However, care must be taken for the fisheye distortion not to reduce readability of the display. To combat the negative aspect of distortion while giving an analyst control, we take three steps shown in Figure 2:

- In the *focus node selection* step, the analyst selects a set of nodes for zooming (Figure 2a). This can be done using a search to select a similar set of nodes or manually by selecting interesting nodes from the dataview window.
- After selection of the focus nodes, regions around the focus nodes are created by the *partition generation* step which applies an incremental algorithm that maintains a set of partitions that varies over time by insertion or deletion[29] as shown in Figure 2b. This second step is to ensure that all the nodes are within the partition after distortion. This is done by measuring the maximum distance to move the node during distortion, the black lines as shown in figure 4d denotes the maximum distance to position the node so that it is within the partition. The distance is measured as the distance of the node label rectangle from the focus to its point of intersection with the partition edge.
- The third step, *fisheye zooming*, distorts each partition as shown in Figure 2c. The results in zooming the focus node and the zooming gradually decrease as we reach the edge of the partition.

The process of zooming in our approach therefore attempts to avoid the problems such as causing excessive distortions, excluding interactive rendering, excluding comparing multiple parts of a graph, excluding identifiers being readable or excluding simultaneous exploration of contents of nodes. Our technique is based on distorting the size and position of the label box based on its edge distances from the focus node. The large node is the focus, comparative similar-size closer nodes, and smaller distant nodes. This helps to identify the focused node and its neighboring nodes.

The Voronoi and rectangular partitioning approaches for a multi-focus technique was presented.[30] They use the traditional rectangular partitioning approach due to its simple computations. We use the Voronoi partitioning approach as

```
Algorithm 3.1: MULTI-FOCUS($NodeList$)

  procedure DRAWVORONOI($NodeList$)
    comment: returns endpoints of the polygons

    Fortune's Voronoi algorithm is called to create the polygons
    return ($PolygonList$)

  procedure FISHEYEZOOMING($PolygonList, NodeList$)
    comment: Apply fisheye distortion for each node

    for $polygon \leftarrow PolygonList.next()$
      do
        $endPoints \leftarrow polygon.getEndpointS()$
        for $node \leftarrow Graph.allnodes()$
          do
            Ray casting is used to find if a node is inside the polygon
            if $RayCasting(node) == true$
              then
                $NodesInPolygon \leftarrow addNode(node)$
                Get the intersectPoint of the node
                Compute $D_{max}$
                Apply arcTan fisheye distortion for the node
                Compute the new font size and position
    return ($NodesInPolygon$)

  main
    Graph layout is rendered and wait for user operation
    if $nodesSelected == true$
      then
        User can select nodes to start analysis
        $NodeList \leftarrow addNode(node)$
    $PolygonList \leftarrow$ DRAWVORONOI($NodeList$)
    $NodesInPolygon \leftarrow$ FISHEYEZOOMING($PolygonList, NodeList$)
```

Figure 3: Pseudocode for multi-focus algorithm. The DrawVoronoi function takes the focus nodes (Nodelist) as inputs and outputs the endpoints of the polygons (PolygonList) for each focus node. The FisheyeZooming function takes the PolygonList and the NodeList as inputs and returns the distorted nodes (NodesInPolygon).

opposed to the traditional rectangular approach to aid incremental partitioning and continuous zooming layout while preserving both structure in an efficient use of space for arbitrary network structures. Previous work has not combined the Voronoi and the fisheye algorithm for multi-focus zooming. The Voronoi polygons[31] are compactly partitioned for available screen space. After applying the fisheye technique locally inside each polygon, the nodes near the sides of all polygons are compressed, to preserve layout continuity. The recent GraphPrism [10] shows graph measures in stacked histograms and highlights nodes in a network based on selections in the histograms. We follow a similar approach using multiple small windows to show more details such as the CPT information of the nodes.

Like the bubbles connecting thoughts to a character in a carton, bubbles act as anchors and connect the nodes to the side panel. Using these bubbles, the user can trace a databox either in the side panel or floating, to its corresponding node in the network view. The colors of the bubbles is similar to the color of the title bar of the databox to help connect the node and the databox. We use an improved version of bubbles previously used[32] replacing the solid row of large dots with a progressively enlarged row of hallow bubbles. By using empty bubbles, the user can now see the underneath network previously hidden by the solid bubbles, see Figure 7 for solid bubbles and Figure 5 for empty bubbles. We experimented with different bubble color representations, solid and empty bubbles and different sizes of the bubbles. We found that empty colored bubbles which progressively increase in size as it reaches the side panel are more effective.

The pseudo-code of the multi-focus algorithm is shown in Figure 3; we now discuss each of the three steps in more detail.

3.1 Step 1: Selection of focus nodes

Many current visualization techniques mainly address how to display the data while the user's primary concern is what is selected for display.[5,33] We provide several options as discussed in the below Section 3.1 to help the users to study the details associated with each node to find the interesting nodes satisfying the design goal, data exploration (DG9).

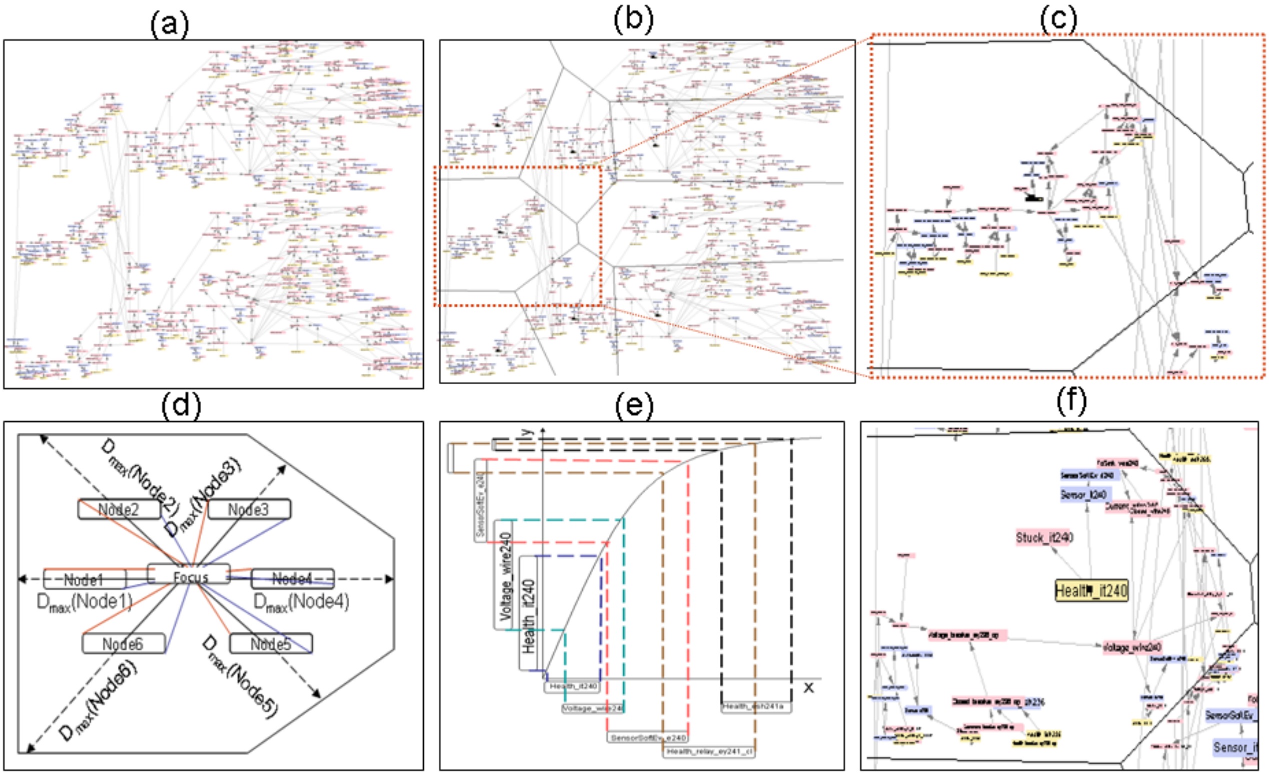


Figure 4: a) A hard to read baseline network; b) Partition lines have been drawn; c) Viewing one partition; d) Showing how the maximum distance for each node label positions are computed (so that they do not move out of the polygon). Blue and red lines show the start and end distance of each node label rectangle from the focus; e) arcTan distortion is applied for each node label rectangle and based on the start(upper-left) and end(lower-right) coordinates; f) Polygon shows focus nodes after distortion

1. The user can study the details of the nodes (databoxes) such as the time-series graphs or the CPTs as shown in the dataview window. If an interesting behavior is found, the databox can be selected so that the corresponding node in the network view also gets selected.

2. In the network view, the databoxes can be viewed as a tooltip as the user hover over the node. The databox associated with a node can be anchored if the time-series graph or the CPT requires further inspection, see Figure 5.

3. A search option can be used to select a set of nodes. This option is useful when the user wants to study all the current nodes or the voltage-sensor nodes in an electrical network, see Figure 7.

4. Rectangular selection allows the user to select a group of nodes in a particular region of the network. The databoxes associated with those nodes can be opened and studied in the side panel, see Figure 5.

5. The databox for a node can also be independently viewed in the side panel or overlaid on the network for a close inspection.

6. Users can select the parent and children nodes of a particular node in a Bayesian network. The selected nodes can be zoomed and studied, see Figure 10 and 9.

3.2 Step 2: Partition Generation

After selecting a set of focus nodes, the analyst has to decide the bounded region around each of the focus node. The zooming algorithm is applied inside this region. In a large dataset, it becomes difficult for the user to intelligently determine a bounded region. The user typically ends up with a random bounding region around the focus node. It would be more

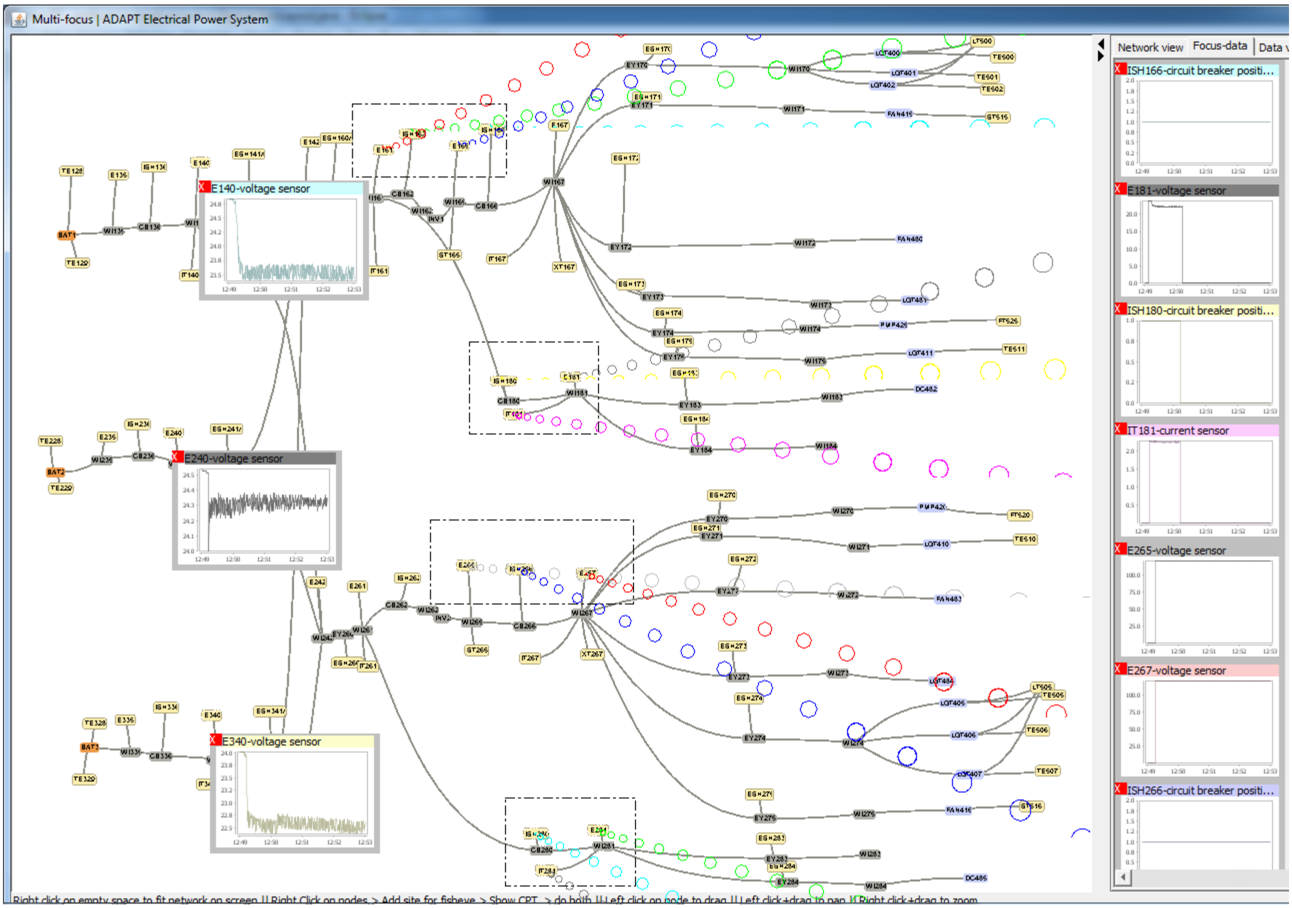

Multi-focus | ADAPT Electrical Power System
Network view
Focus-data
Data v
ISH166-circuit breaker positi...
E181-voltage sensor
ISH180-circuit breaker positi...
IT181-current sensor
E265-voltage sensor
E267-voltage sensor
ISH266-circuit breaker positi...
E140-voltage sensor
E240-voltage sensor
E340-voltage sensor
BAT1
BAT2
BAT3
Right click on empty space to fit network on screen || Right Click on nodes > Add site for fisheye > Show CPT > do both || Left click on node to drag || Left click+drag to pan || Right click+drag to zoom

Figure 5: The time-series graphs of all the nodes inside the rectangular selection are aligned and shown in the side panel. Analysts can hover over the node to display the time-series graph as a tooltip. The graphs of the voltage sensors E140, E240 and E340 have been anchored in the network view by clicking on the network nodes as shown.

beneficial if we could predetermine a region of user interest for a given focus node based on some intuitive technique. We experimented with a variety of rectangular partitioning approaches but found them causing discontinuities.[30] The Voronoi algorithm[29] can satisfy all of our design goals (DG4, DG5 and DG6); computing display area by dividing the plane into *n* selected regions for each of the *n* node selections. It is based on the principle that any node in the region will be nearer to the zoomed node in that region than to any of the neighbouring region's zoomed nodes. Thus the diagram forms regions of interest for each of the zoomed node resulting in an intuitive way of partitioning the screen space.

When the user selects a node, a partitioning algorithm[29] is applied to that node and all previously selected nodes, showing the incremental aspect of the algorithm. A divide and conquer approach is followed wherein the screen space is partitioned into polygons. We apply the local fisheye to a bounded area by retrieving the corner co-ordinates of the region and updating the display accordingly. Each node in the graph is checked to see if it is present in the selected nodes' partitioned area using a ray casting technique. * Only those nodes in the selected nodes' partitioned area undergoes the fisheye distortion.

3.3 Step 3: Fisheye Zooming

The analyst may want study the focus nodes in each partition. To do this, he may zoom-in on the focus nodes and the nearby nodes in the partitions. A local fisheye effect does this; the selected node is zoomed in and the size of the nearby nodes increases. We minimize the traditional fisheye effect that distorts the whole layout by localizing the fisheye effect within the partitions. The user selected node and the nearby nodes are zoomed out; the size of the nearby nodes decrease

*http://www.ecse.rpi.edu/Homepages/wrf/Research/Short_Notes/pnpoly.html

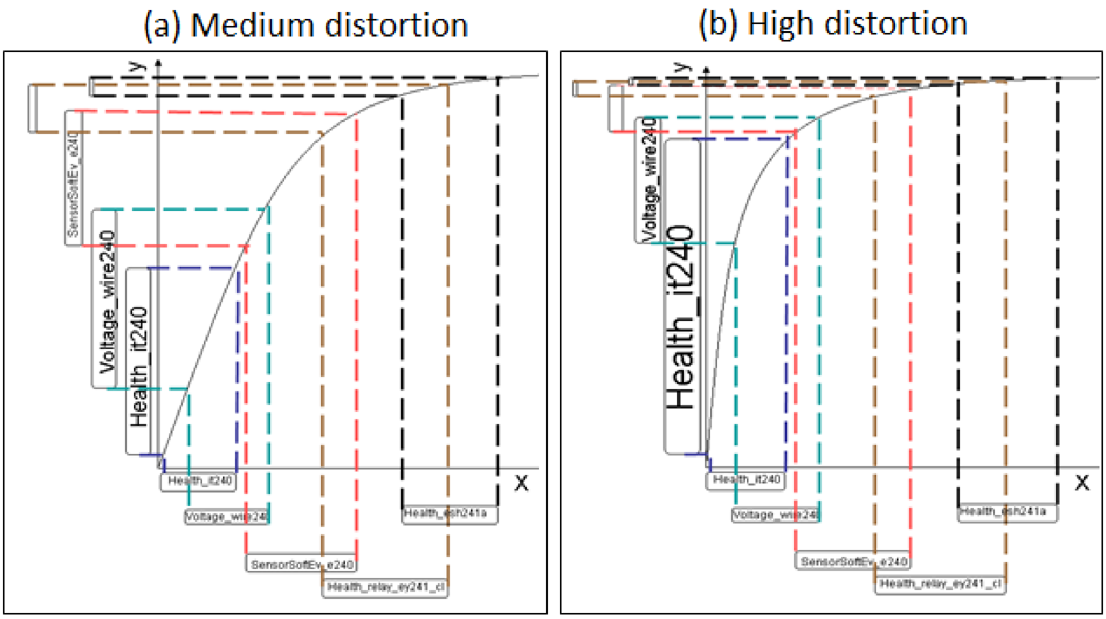


Figure 6: Arctan curve for different values of b, the distortion factor and its effect on the nodes size. a) $b = 2.5$ b) $b = 5.0$. The undisturbed nodes are shown on the x-axis, while the nodes after distortion are shown in the y-axis. It is hard for an analyst to read the small labels along the x-axis.

by the arctan of the distance from the focus as they reach the edge of the partition providing a continuous layout. For each node inside the polygon, the start (upper-left corner) and the end (lower-right corner) co-ordinates of a node label are transformed with respect to the focus (center of the focus node) similarly. The start and end distance of a node from the focus is denoted by D_{start} and D_{end}. It is shown by the red and blue lines in figure 4(d). The distance D_{max} is measured as the distance from the focus to the node label's point of intersection with the edge of the polygon as shown in Figure 4(d). These distance values are used to compute the transformed start and end co-ordinates of the node label box. The font size of the labels are then determined based on the difference between the transformed start and end co-ordinates in the X-dimension. The new position of the node is obtained by finding the center coordinates. The arctan fisheye distortion for start or end co-ordinates (x,y) is done in the following steps:

Transforming the co-ordinate system such that the focus centered at (x_{focus}, y_{focus}) is shifted to $(0,0)$ and finding the distance of the node centered at (x, y) from the focus is done using:

$$d_x = x - x_{focus}, d_y = y - y_{focus}$$
$$D = \sqrt{(d_x)^2 + (d_y)^2}$$

Conversion from Cartesian to Polar co-ordinates is done using:

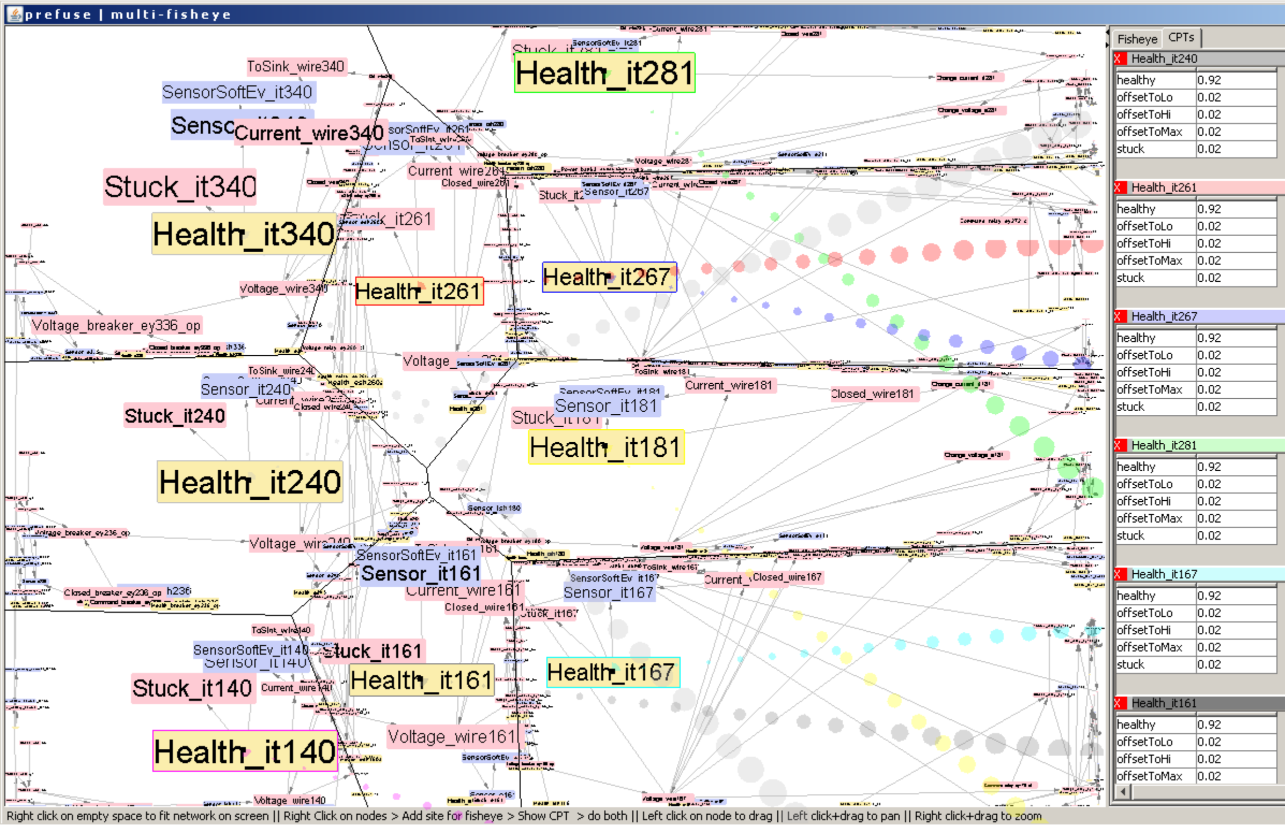
$$\theta = \arctan(\frac{d_y}{d_x})$$

Normalizing the distance so that the node is not moved outside its Voronoi polygon:

$$d_{norm} = D/D_{max}$$

Calculation of radial distance and de-normalization is done using:

$$r = a * \arctan(b * d_{norm}) * D_{max}, \text{ where } a = \frac{1}{\arctan(b)} \text{ so that the values of } r \; \varepsilon \; [0,1] \text{ before de-normalization.}$$



Health_it240

healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it261

healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it267

healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it281

healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it167

healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02
stuck	0.02

Health_it161

healthy	0.92
offsetToLo	0.02
offsetToHi	0.02
offsetToMax	0.02

Right click on empty space to fit network on screen || Right Click on nodes > Add site for fisheye > Show CPT > do both || Left click on node to drag || Left click+drag to pan || Right click+drag to zoom

Figure 7: A zoomed-in visualization of nine focus nodes using the distortion factor $b = 18$. The health node labels are all clearly readable. The node Health_it281 has a green color border. The bubbles connecting the node and the title bar of the databox in the side panel are also represented in green color.

Distortion can be increased or decreased by respectively increasing or decreasing the value of b, the distortion factor. Figure 6 shows example arctan curves and their effects on the sizes of nodes for different values of the distortion factor b.

Conversion from Polar to Cartesian co-ordinates is done using:

$$x' = r\cos(\theta) + x_{focus},\ y' = r\sin(\theta) + y_{focus}$$

The above steps are applied to both the start and end coordinates to get the new transformed (x'_{start}, y'_{start}) and (x'_{end}, y'_{end}) coordinates. Size distortion is then done by finding the new readable label based on the difference between the transformed coordinates $(x'_{end} - x'_{start})$. The new position of the node is calculated as:

$$x'_{center} = (x'_{end} - x'_{start})/2$$
$$y'_{center} = (y'_{end} - y'_{start})/2$$

Figure 7 shows the multi-focus zooming effect on nine focused nodes for a distortion factor $b = 18$.

4. APPLICATION

Our novel multi-focus technique is implemented using the Prefuse framework.[34] ADAPT Bayesian network as well as other networks have been visualized. Figure 8 [†] shows the usage of the multi-focus technique in detecting faulty components in an electrical power system. Figure 9 shows the Munin2 Bayesian network with 1003 nodes where the children of a node (Proximal Myopathy) are zoomed using our multi-focus technique.

[†] http://www.youtube.com/watch?v=oJh1kbQVcXc

4.1 Bayesian Networks

The ADAPT Bayesian network, which can be used for automatic fault diagnosis,[1] is a representative of an electrical power network found in aerospace vehicles. ADAPT has capabilities for power storage, distribution, and consumption, and contains batteries, electromechanical relays, circuit breakers, and different kinds of loads, such as pumps, fans, and light bulbs. Each component in ADAPT is modeled as a set of random variables. These random variables form the nodes and the dependency relationship among the nodes are shown by the edges. Health nodes (**H**) and the evidence (**e**) are of particular interest. The health nodes serve as the query variables, e.g. whether a component is defective or not, and the evidence nodes serve as the input variables, e.g. a command such as closing a relay to allow current from the battery to flow to the load. As an example fault scenario, Mengshoel et al. explains a fault diagnosis scenario in,[1] suppose that **e** = {CommandRelay = cmdClose, SensorCurrent = readCurrentLo, SensorVoltage = readVoltageHi, SensorTouchSensor = readClosed}. This gives $\arg\max P(\mathbf{H} \mid \mathbf{e})$= {HealthBattery = healthy, HealthLoad = healthy, HealthCurrent = stuckCurrentLo, HealthVoltage = healthy}. In this scenario, given the evidence of the command and sensor readings, the current flows from the battery to the load as both of them are healthy, also the voltage sensor is healthy. So the defective component is the current sensor which reads low instead of reading high.

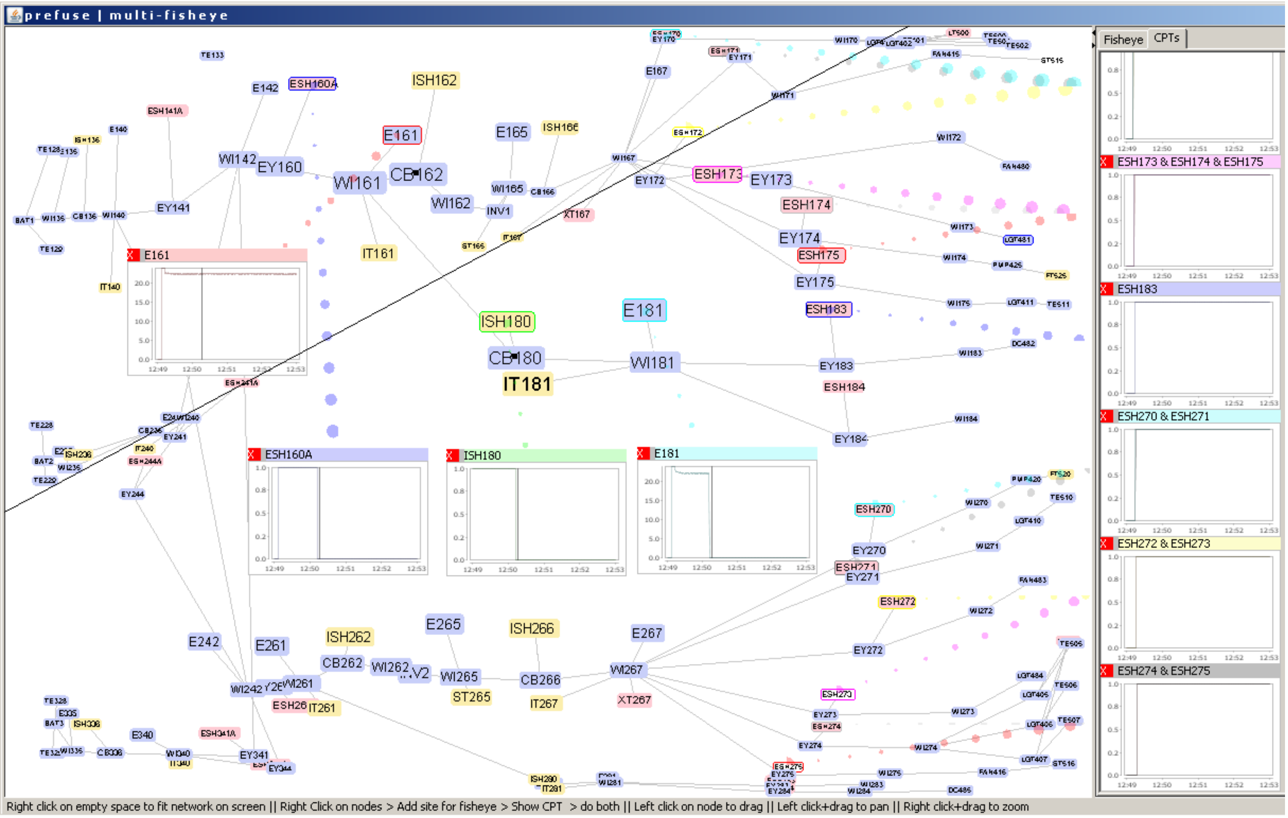
The hard to read ADAPT network with no distortion is shown in Figure 1a. The network consists of 671 nodes and 790 edges. The font size of the labels is the same for all the nodes. It is impossible to read these labels on the computer screen making it extremely difficult to understand, validate or edit the Bayesian network. The analyst interacts with the graph by changing the position of focus. The focus nodes which have been zoomed, will display a larger font than their neighboring nodes as shown in Figure 7. It is now possible to read the node labels for the focus nodes and the nodes close to them. In general, multi-focus selection is used to make the labels readable and for comparing various nodes to explore their differences and similarities. The side panel in the Figure7 shows detailed information about the nodes, specifically the conditional probability tables, for further comparison and analysis.[32] The bubbles help the user to trace the databox to its corresponding node in the network view.

4.2 Evaluation

Our evaluation approach has been to explore complex networks while taking note of how the tool aided an analyst in finding and remembering nodes of value during a problem solving session. Our exploration included working with a Bayesian network expert. Many features and algorithms were explored, including alternative distortion approaches under rectangular and optimized polygonal partitioning. In the end, simplifying the controls and amplifying the mechanisms for remembering where one is in the network exploration process were necessary for the user to make sense of the network.

The interface is a dramatic simplification over pop-up style controls, and helps focusing action on the essential gather and prune activities of a network visualization system. The interface felt agile and powerful to our Bayesian network expert as he was able to reformulate network questions several times a minute. He commented on and enjoyed discovering six mechanisms to orient, annotate and understand the relations between nodes. (1) Partitioning allowed him to quarantine (he used the word ”sacred”) parts of the network that contained interesting nodes. (2) The fisheye, he said, allowed him to highlight and remember which nodes he deemed important in a very visible way. (3) The bubbles gave him easy to follow indications of where important nodes were. (4) The motion of panning made the bubbles lines show how distant the nodes were separate from other mechanisms. (5) Panning motions and mouse-over helped resolve nodes that were overlapping. (6) The use of node colouring helped to focus on nodes of similar types.

The interesting nodes in the network was found by using the search techniques as discussed in Section 3.1. For Bayesian networks, in reviewing the conditional probability tables associated with the focus nodes, our network expert found himself using a collect-review-dispense loop to home in on the conditional probability tables that needed to be compared. Often he would collect 10-20 of these tables and then prune down to 4-5 in one iteration of this collect-review-dispense loop. He described the activity as a network review, similar to code review in software engineering, as he poked around hunting and foraging with the support of the system’s many memory aids. In particular, the tool helped in identifying important nodes for further analysis and comparison in the side panel. Two different tasks can be performed in a Bayesian Network, the construction and the diagnosis task.



Right click on empty space to fit network on screen || Right Click on nodes > Add site for fisheye > Show CPT > do both || Left click on node to drag || Left click+drag to pan || Right click+drag to zoom

Figure 8: Time-series graphs of the zoomed sensor nodes in the ADAPT electrical power network. The node labels and their time-series graphs are readable for further analysis. The time-series graphs around the node CB180 show a drop in their reading, suggesting that the component CB180 is faulty.

4.3 Analytical Tasks

4.3.1 Construction Task

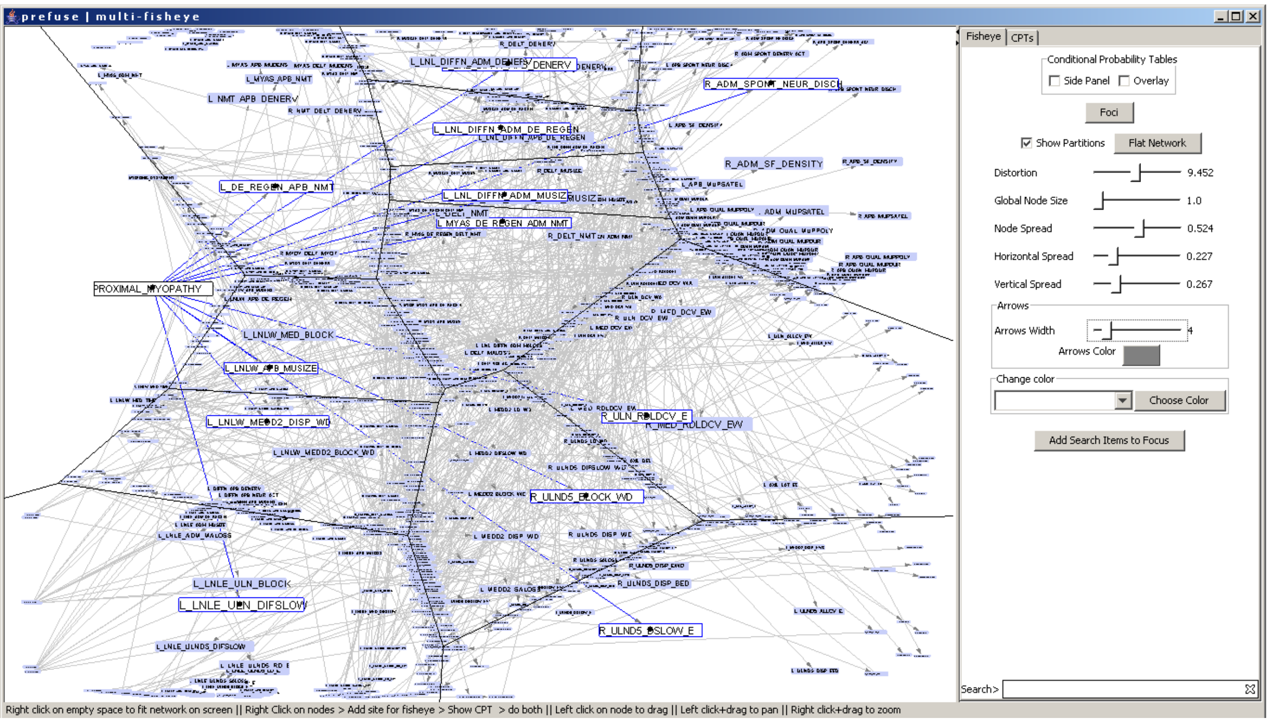
The ADAPT Bayesian network is used for identifying failed components in an electrical power system. When constructing such a network the user may want to compare the CPTs of a set of nodes, for example, health nodes that may represent health of different components but with similar conditional probability tables as shown in Figure 7. We investigate $P(H_i|pa(H_i))$ where $H_i \in \mathbf{H}$ is a Bayesian network health node and $pa(H_i)$ denote the parent nodes of H_i. There can be multiple nodes with high posterior probabilities that may be the major causal nodes for certain defects. These nodes can lie quite far-apart in large Bayesian network as shown in Figure 10. Using the multi-focus technique, they can be zoomed to analyze their CPTs.

4.3.2 Diagnosis Task

The diagnosis task investigates $P(H_i|\mathbf{e})$ where H_i is a Bayesian network health node and $\mathbf{e}$ is the evidence. Here, multi-focus can help a Bayesian analyst by allowing him or her to focus, at the same time, on multiple X_i's with interesting posteriors $P(H_i|\mathbf{e})$. For example, it might be that multiple nodes have high posterior probabilities of being defective in a diagnostic Bayesian network such as ADAPT. If the nodes are distant in the layout (as they can easily be in ADAPT and other Bayesian networks), our multi-focus technique provides valuable support for interactive analysis.

5. DISCUSSION AND FUTURE WORK

Distortion such as fisheye views can increase the ability to keep context visible. Multi-focus approaches expand visual analysis by allowing comparison of different parts of a network providing analysts with a collect-review-dispense visual



Conditional Probability Tables

☐ Side Panel ☐ Overlay

Foci

☑ Show Partitions Flat Network

Distortion 9.452

Global Node Size 1.0

Node Spread 0.524

Horizontal Spread 0.227

Vertical Spread 0.267

Arrows

Arrows Width 4

Arrows Color

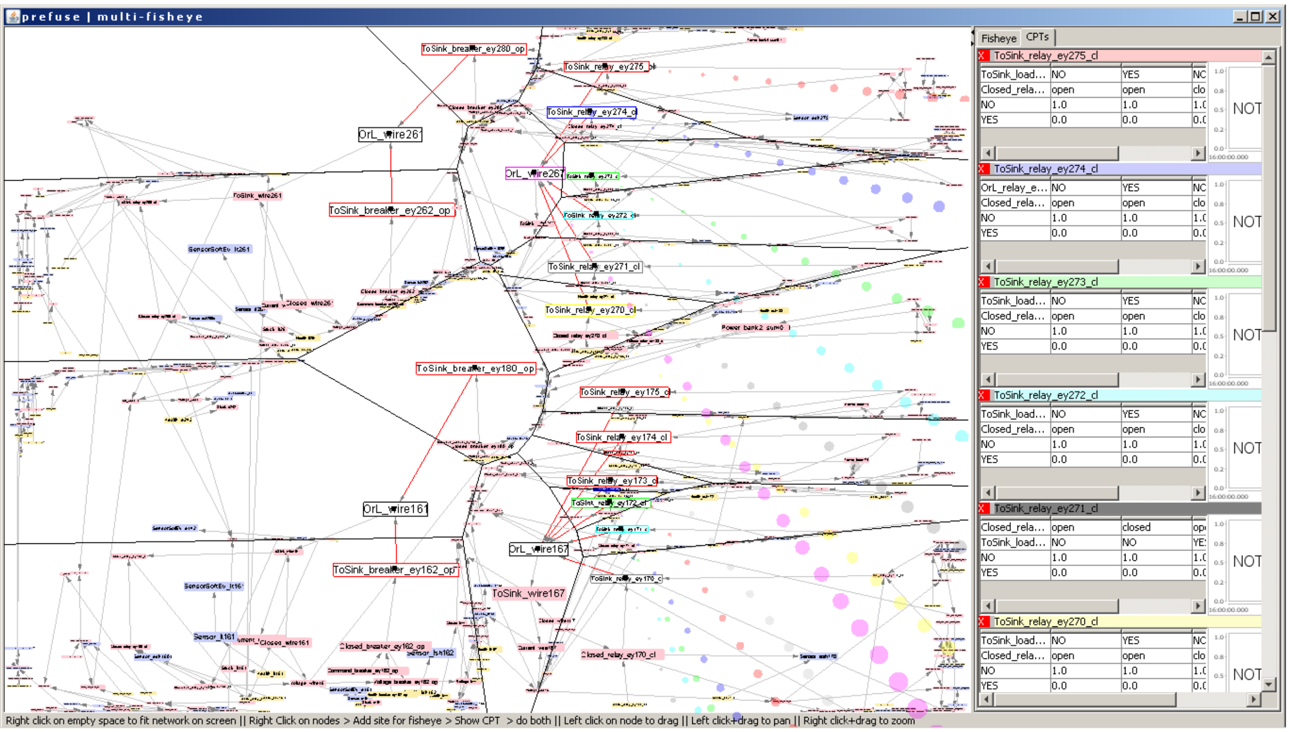
Change color

Choose Color

Add Search Items to Focus

Right click on empty space to fit network on screen || Right Click on nodes > Add site for fisheye > Show CPT > do both || Left click on node to drag || Left click+drag to pan || Right click+drag to zoom

Figure 9: The large Munin2 Bayesian network and application of multi-focus on the children nodes (*L_LNLE_ULN_DIFLOW*, *L_LNLW_MED2_DISP_WO* etc.) for neural disorder disease node called *Proximal Myopathy* (the left most focus node). The children and the disease nodes are clearly readable on the computer screen even though they are difficult to read in this screenshot.



ToSink_relay_ey275_cl

ToSink_load...	NO	YES	NC
Closed_rela...	open	open	clo
NO	1.0	1.0	1.(
YES	0.0	0.0	0.(

ToSink_relay_ey274_cl

OrL_relay_e...	NO	YES	NC
Closed_rela...	open	open	clo
NO	1.0	1.0	1.(
YES	0.0	0.0	0.(

ToSink_relay_ey273_cl

ToSink_load...	NO	YES	NC
Closed_rela...	open	open	clo
NO	1.0	1.0	1.(
YES	0.0	0.0	0.(

ToSink_relay_ey272_cl

ToSink_load...	NO	YES	NC
Closed_rela...	open	open	clo
NO	1.0	1.0	1.(
YES	0.0	0.0	0.(

ToSink_relay_ey271_cl

Closed_rela...	open	closed	op
ToSink_load...	NO	NO	YE
NO	1.0	1.0	1.(
YES	0.0	0.0	0.(

ToSink_relay_ey270_cl

ToSink_load...	NO	YES	NC
Closed_rela...	open	open	clo
NO	1.0	1.0	1.(
YES	0.0	0.0	0.(

Right click on empty space to fit network on screen || Right Click on nodes > Add site for fisheye > Show CPT > do both || Left click on node to drag || Left click+drag to pan || Right click+drag to zoom

Figure 10: The zoomed parent nodes of the *Orl_wire* nodes and their CPTs in the ADAPT electrical Bayesian network. These node labels are readable on the computer screen even though they are difficult to read in this screenshot.

task scenario. Our system generates distortion boundaries which can reduce global distortion, minimizing visual comparison changes. In particular, our technique allows an analyst to interactively bring important parts of a network ‘forward’ by selectively zooming in, to be compared both structurally in the network and in a multi-window semantic display.[32] Our system performs well over a dozen focus partitions.

This system gives simultaneous multi-focus zooming ability to enable interactive visual debugging of Bayesian networks. In case of a failure in an electrical circuit, the user may want to find the faulty component. For this, a deeper analysis of each of the component is required. Using the multi-focus technique similar components, such as voltage or current sensors, can be zoomed and their internal readings can be studied. The multiple windows can be floating or aggregated in the side panel. The side panel is designed to align and compare internal readings of multiple selected nodes. If there is any sudden drop or rise in the readings in any of the components, then it can be diagnosed further.

The multi-focus with the multi-window views shown in this paper should improve completion accuracy in network analysis. We have used our technique in electrical networks (Figure 8), Bayesian networks (Figure 9) and even on social networks. This paper pushes for adding techniques to the arsenal of ways to allow users to better view large networks and analyze their complex problems. Analytics and reasoning are being done on increasingly complex datasets. This paper demonstrates improvements towards and calls for work on systems that integrates scalable user interactivity into comparing parts and internal semantics of large-scale networks.

ACKNOWLEDGMENTS

This material is based, in part, upon work supported by NSF grants CCF-0937044 and ECCS-0931978.

REFERENCES

[1] Mengshoel, O. J., Chavira, M., Cascio, K., Poll, S., Darwiche, A., and Uckun, S., “Probabilistic Model-Based Diagnosis: An Electrical Power System Case Study,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **40**(5), 874–885 (2010).

[2] J.Pearl, [*Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*], Morgan Kaufmann (1988).

[3] Andersen, S. K., Olesen, K. G., Jensen, F. V., and Jensen, F., “HUGIN - A Shell for Building Bayesian Belief Universes for Expert Systems,” in [*Proc. of IJCAI’89*], 1080–1085 (Aug 1989).

[4] Druzdzel, M. J., “SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: A Development Environment for Graphical Decision-Theoretic Models,” in [*Proc. of AAAI’99*], 902–903 (Jul 1999).

[5] Furnas, G. W., “Generalized fisheye views,” *SIGCHI* **17** (April 1986).

[6] Reinhard, T., Meier, S., and Glinz, M., “An improved fisheye zoom algorithm for visualizing and editing hierarchical models,” in [*Proceedings of the Second International Workshop on Requirements Engineering Visualization*], *REV ’07*, 9–, IEEE Computer Society, Washington, DC, USA (2007).

[7] Storey, M. A. D., Fracchia, F., and Müller, H., “Customizing a Fisheye View Algorithm to Preserve the Mental Map,” *Journal of Visual Languages and Computing* (10), 245–267 (1999).

[8] Shneiderman, B., “The eyes have it: A task by data type taxonomy for information visualizations,” in [*Proceedings of the 1996 IEEE Symposium on Visual Languages*], 336–, IEEE Computer Society, Washington, DC, USA (1996).

[9] Fortune, S., “A sweepline algorithm for voronoi diagrams,” in [*Proceedings of the second annual symposium on Computational geometry*], *SCG ’86*, ACM, New York, NY, USA (1986).

[10] Cockburn, A., Karlson, A., and Bederson, B. B., “A review of overview+detail, zooming, and focus+context interfaces,” *ACM Comput. Surv.* **41**, 2:1–2:31 (Jan. 2009).

[11] Sarkar, M. and Brown, M. H., “Graphical fisheye views of graphs,” in [*Proceedings of the SIGCHI conference on Human factors in computing systems*], *CHI ’92*, 83–91, ACM, New York, NY, USA (1992).

[12] Lamping, J., Rao, R., and Pirolli, P., “A focus+context technique based on hyperbolic geometry for visualizing large hierarchies,” in [*Proceedings of the SIGCHI conference on Human factors in computing systems*], *CHI ’95*, 401–408, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1995).

[13] Sarkar, M., Snibbe, S. S., Tversky, O. J., and Reiss, S. P., “Stretching the rubber sheet: a metaphor for viewing large layouts on small screens,” in [*Proceedings of the 6th annual ACM symposium on User interface software and technology*], *UIST ’93*, 81–91, ACM, New York, NY, USA (1993).

[14] Formella, A. and Keller, J., "Generalized fisheye views of graphs," in [*Proceedings of the Symposium on Graph Drawing*], *GD '95*, 242–253, Springer-Verlag, London, UK (1996).

[15] Gansner, E. R., Koren, Y., and North, S. C., "Topological fisheye views for visualizing large graphs," *IEEE Transactions on Visualization and Computer Graphics* **11**, 457–468 (July 2005).

[16] Ghani, S., Riche, N., and Elmqvist, N., "Dynamic insets for context-aware graph navigation," in [*Computer Graphics Forum*], **30**(3), 861–870, Wiley Online Library (2011).

[17] Lee, T., "Coherent time-varying graph drawing with multifocus+ context interaction," *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS* **18**(8) (2012).

[18] Bartram, L., Ho, A., Dill, J., and Henigman, F., "The continuous zoom: a constrained fisheye technique for viewing and navigating large information spaces," in [*Proc ACM symposium on User interface and software technology*], *UIST '95*, ACM, New York, NY, USA (1995).

[19] Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., and Roseman, M., "Navigating hierarchically clustered networks through fisheye and full-zoom methods," *ACM Trans. Comput.-Hum. Interact.* **3** (1996).

[20] Plaisant, C., Grosjean, J., and Bederson, B., "Spacetree: supporting exploration in large node link tree, design evolution and empirical evaluation," in [*Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*], (2002).

[21] Munzner, T., Guimbretière, F., Tasiran, S., Zhang, L., and Zhou, Y., "TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility," *ACM Trans. Graph.* **22**, 453–462 (July 2003).

[22] Tu, Y. and Shen, H.-W., "Balloon focus: a seamless multi-focus+context method for treemaps," *IEEE Transactions on Visualization and Computer Graphics* **14**, 1157–1164 (November 2008).

[23] Batagelj, V., Didimo, W., Liotta, G., Palladino, P., and Patrignani, M., "Visual analysis of large graphs using (x,y)-clustering and hybrid visualizations," in [*Pacific Visualization Symposium (PacificVis), 2010 IEEE*], (march 2010).

[24] Elmqvist, N., Riche, Y., Henry-Riche, N., and Fekete, J.-D., "Melange: Space folding for visual exploration," *Visualization and Computer Graphics, IEEE Transactions on* **16**, 468 –483 (may-june 2010).

[25] Keahey, T. A., Gucht, D. V., Keahey, T. A., Jerde, N. G., and Keahey, T. E., "Nonlinear magnification," tech. rep., transformations,Proceedings of the IEEE Symposium on Information Visualization, IEEE Visualization (1997).

[26] Carpendale, M. S. T., Sheelagh, M., Carpendale, T., Cowperthwaite, D. J., and Fracchia, F. D., "3-dimensional pliable surfaces: For the effective presentation of visual information," in [*In Proc. of UIST'95*], ACM (1995).

[27] Kaugars, K., Reinfelds, J., and Brazma, A., "A simple algorithm for drawing large graphs on small screens," in [*Proc. of the DIMACS International Workshop on Graph Drawing*], *GD '94*, Springer-Verlag, London, UK (1995).

[28] Herman, I., Melancon, G., and Marshall, M. S., "Graph visualization and navigation in information visualization: A survey," *IEEE Transcations on Visualization and Computer Graphics* **6**(1), 24–43 (2000).

[29] Aurenhammer, F., "Voronoi diagrams a survey of a fundamental geometric data structure," *ACM Comput. Surv.* **23**, 345–405 (September 1991).

[30] Sundararajan, P. K., Mengshoel, O., and Selker, T., "Multi-fisheye for interactive visualization of large graphs," in [*The AAAI-11 Workshop on Scalable Integration of Analytics and Visualization*], (2011).

[31] Balzer, M., Deussen, O., and Lewerentz, C., "Voronoi treemaps for the visualization of software metrics," in [*Proceedings of the 2005 ACM symposium on Software visualization*], *SoftVis '05*, ACM, New York, NY, USA (2005).

[32] Cossalter, M., Mengshoel, O. J., and Selker, T., "Visualizing and understanding large-scale Bayesian networks," in [*The AAAI-11 Workshop on Scalable Integration of Analytics and Visualization*], 12–21 (2011).

[33] Furnas, G. W., "A fisheye follow-up: further reflections on focus + context," in [*Proceedings of the SIGCHI conference on Human Factors in computing systems*], *CHI '06*, 999–1008, ACM, New York, NY, USA (2006).

[34] Heer, J., Card, S. K., and Landay, J. A., "Prefuse: a toolkit for interactive information visualization," in [*Proceedings of the SIGCHI conference on Human factors in computing systems*], *CHI '05*, ACM, New York, NY, USA (2005).