Geoffrey M. Draper ; Matthew G. Styles and Richard F. Riesenfeld
"Radial visualizations for comparative data analysis", Proc. SPIE 8294, Visualization and Data Analysis 2012, 82940Y (January 24, 2012);

http://dx.doi.org/10.1117/12.912450

Radial Visualizations for Comparative Data Analysis

Geoffrey M. Draper[a], Matthew G. Styles[a] and Richard F. Riesenfeld[b]

[a]Department of Computer and Information Sciences, Brigham Young University Hawaii, Laie, Hawaii, USA

[b]School of Computing, University of Utah, Salt Lake City, Utah, USA

ABSTRACT

SQiRL is a novel visualization system for querying and visualizing large multivariate data sets. Although initially designed for novice users, recent extensions to SQiRL facilitate more advanced analysis without sacrificing the simplicity that makes this visualization appealing to beginners. The default view provides a simple-to-learn interface for query evaluation. Intermediate users are provided a straightforward method for comparing the results of two queries. More advanced users can make use of a "radial crosstab," a new interactive visualization technique that melds the expressive power of traditional crosstabulation with a drag-and-drop canvas.

Keywords: Visual query languages, radial visualization, crosstabulation

1. INTRODUCTION

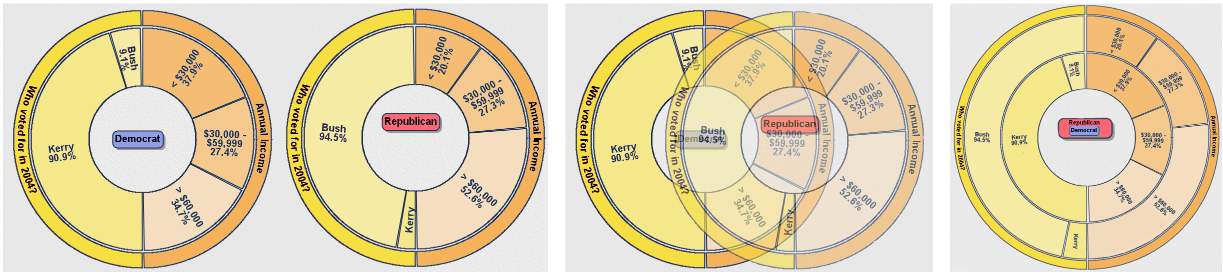
One goal of the "information visualization for the masses"[1] movement is to make easy-to-use tools available for people who may not necessarily be experts in data analysis. At the same time, system builders face the challenge of not diluting the interface to the point that it inhibits expert users from doing their work. A compromise, as proposed by Norman,[2] is to design systems such that advanced features are hidden at first. As users' experience and confidence grow, additional functionality is revealed.

This paper describes an application of this concept to a visual query system for demographic data analysis. The SQiRL system (Simple Query Interface with a Radial Layout)[3] provides an easy-to-learn user interface for exploring relationships in multivariate data. However, in order to maintain a simple design, its functionality was intentionally limited in the first iteration of the software. In particular, it lacked the ability to simultaneously compare intersections among multiple variables.

For novice users, a simplified interface such as this is entirely appropriate. However, our observations of SQiRL's users have shown that as soon as they become somewhat comfortable with SQiRL, they immediately ask how they can compare the results of two queries. Furthermore, advanced users with prior experience with data analysis usually request the ability to compare several variables at once, similar to a crosstabulation (see Section 4). To accommodate the diverse needs of users with varying experience levels, we propose a three-tiered visualization tailored to the needs of beginning, intermediate, and advanced users, respectively. In the default (basic) visualization, queries can be constructed quickly and easily via a familiar drag-and-drop style interface. For intermediate users, we provide a straightforward "visual diff"[4] visualization for comparing the results of two related queries. For advanced users, we propose a method called "radial crosstabs" for visualizing the results of several multivariate queries simultaneously.

The remainder of this paper is organized as follows. In Section 2, we review some of the literature in the area of radial visualization. In Section 3, we extend SQiRL's basic usage to include the comparison of two related queries. Section 4 discusses a simple yet powerful extension to SQiRL for more advanced data analysis. Finally, in Section 5, we draw conclusions and discuss possible directions for further work.

Further author information: (Send correspondence to G.M.D.)
G.M.D.: E-mail: geoffrey.draper@byuh.edu, Telephone: 808 675 3295
M.G.S.: E-mail: mstyles@go.byuh.edu, Telephone: 808 954 0852
R.F.R.: E-mail: rfr@cs.utah.edu, Telephone: 801 581 8224



(a) (b) (c)

Figure 1. (a) Proposed "side by side" visual diff. Although the charts shown above use the same set of attributes on the ring's circumference, this is not a strict requirement for side by side comparisons. This affords the side by side approach great flexibility. (b) Proposed "partial overlap" visual diff. While less demanding in terms of space, the occlusion issues may limit users' ability to read and interpret the data. (c) Proposed "concentric" visual diff. This is more space-efficient than the previous two techniques, and simplifies the problem of directly comparing sector magnitude.

2. RELATED WORK

SQiRL's user interface is an example of *radial visualization*, an increasingly prominent paradigm in information visualization. In a radial visualization system, information is rendered in a circular or elliptical pattern on an interactive canvas.

Despite its recent growth in popularity, radial visualization is not a new idea. Radial diagrams such as pie charts found usage in statistical graphics since at least the 1800's.[6] In the mid-twentieth century, interactive variants of radial diagrams were proposed for studying the social behaviors of gradeschool children.[7]

Modern radial visualizations can be broadly classified into three categories, *Polar Plot*, *Radial Space Filling*, and *Ring-based*.[8] Although SQiRL is most closely related to the Ring-based category, we briefly review all three to provide a context for our discussion.

In a *Polar Plot* layout, the centroid of the graphic has some semantic meaning, and other nodes are positioned relative to it. Nodes that have a specific relation to other nodes are typically connected with line segments. Thus, meaning is expressed both by which nodes are connected to each other, and their relative distance from the centroid.

Radial Space Filling visualizations[9] are similar to polar plot in that the centroid typically has some semantic meaning, and other nodes are positioned relative to it. The primary visual difference is that the nodes are packed tightly around the centroid, with a minimum of whitespace. The most common arrangements are to render the nodes either in concentric rings or as a tight spiral.

The *Ring-based* layouts differ from the previous two schemes in that the visualization's centroid is de-emphasized. Instead, the space is divided into three areas: the space inside the ring, the space outside of the ring, and finally the ring itself — with most nodes clustered around the ring's circumference.

Like other systems using the Ring layout, meaning in SQiRL is associated primarily with the relative location of nodes on the canvas. Query terms are placed by the user into the ring's interior, while results are read from nodes along the circumference. The space outside the ring is reserved as a temporary staging area for icons that were recently removed from, or will soon be added to, the current query.

3. VISUAL DIFFS

SQiRL is a data visualization system for performing database queries and visualizing the results, originally developed at the University of Utah. It features an integrated query interface that supports rapid exploration and "information foraging"[10] to focus on global trends in the data. Space limitations do not permit an extensive review of it here; for details, please see Ref. 3.

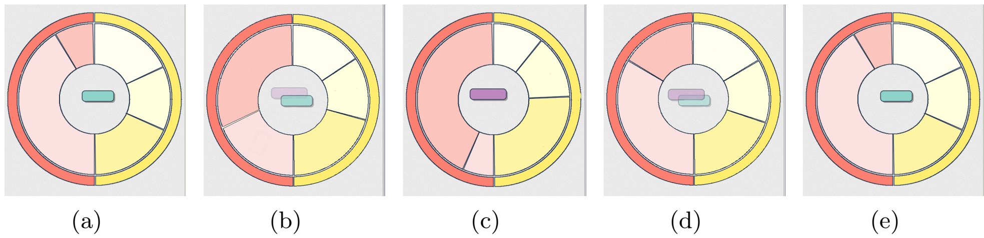


Figure 2. Proposed "animated" visual diff. The animation begins (a) by showing the results for the first query, then during the transition (b) to the second, the sectors resize and the terms in the middle fade in/out. The animation pauses to show the results (c) of the second query, and then undergoes a reverse animation (d) back to the first query (e).

Once users become comfortable with SQiRL's basic interface, we have observed that the first thing many of them want is a way to compare the results of two queries — in other words, a "visual diff".[4] For example, if one query shows what percentage of Democrats in a given population voted for a certain candidate, a possible follow-up query might be to see how the Republicans in that same population voted. The original SQiRL prototype afforded no way to do this, other than to perform the two queries in sequence and hope to remember the results from one query to the next. Clearly, this approach leaves considerable room for improvement.

We considered a number of visualizations for comparing two queries, such as: rendering visualizations side by side, overlapping visualizations, concentric rings, and time-delayed animations. We present a discussion of each, with its strengths and weaknesses. We then describe an empirical evaluation of these techniques in Section 3.1.

- **Side by side.** Perhaps the most obvious way of comparing two results is simply to render two charts and display them side by side (Figure 1a). The side-by-side comparison technique is at the heart of the spreadsheet interface for visualization exploration as proposed by Jankun-Kelly and Ma.[11] This technique is very flexible, but its space requirements grow linearly with the number of comparisons being made.
- **Partial overlap.** A slight modification to the side-by-side approach is to render two charts, but position them so that they overlap (Figure 1b). This decreases the amount of space required to render the visualization, at the cost of some occlusion. The occlusion problem can be ameliorated somewhat, however, by rendering one or both of the charts with partial transparency.
- **Concentric rings.** Another way of showing the results of multiple queries is to render two rings concentrically, one ring per query result (Figure 1c). In this way, each attribute/value pair for the various subpopulations is rendered adjacently, facilitating comparison. While this approach uses less real estate than the previous two techniques, it is inflexible in that its effectiveness depends on each query using the same set of attributes on ring's circumference.
- **Animated.** In the animated comparison scheme, the display toggles between two query results. The results of one query are shown for a moment, then the sectors smoothly resize to show the results of the other query. Like the concentric approach, the animated method is most effective when the attributes shown on the ring's circumference are the same across both queries. However, in this case, the terms in the ring's interior are easily distinguished from one query to the other by a "fade-in/fade-out" effect during the animation (Figure 2). Of all the comparison techniques discussed herein, this one makes the most efficient use of display real estate, since it uses *time*, rather than *space*, to communicate difference information.

3.1 Empirical Evaluation

To evaluate the usability of these different "visual diff" techniques, we conducted a user study involving 30 participants. Our goal was to measure how *quickly* and how *accurately* users could read and interpret the diagrams in each of the four styles.

Figure 3. The quiz program displayed a series of dialog windows presenting a visual diff graphic alongside a question related to it.

3.1.1 Experimental Design

In the experiment, participants were shown a series of 20 images, and asked to answer 3 analysis questions about each. Each image was a screenshot or mockup of a different query in SQiRL. Five images were of the "side-by-side" visualization. Five showed the "partial overlap" visualization. Five showed the "concentric rings" visualization, and the remaining five showed the "animated" visualization. (The "animated" examples were rendered as animated GIFs, rather than static screenshots. The animation lasted approximately 2 seconds, 1 second for each query, with the transition taking approximately 0.1 seconds.) The four styles were distributed evenly throughout the sequence of 20 images, with each visualization illustrating a distinct query of the data.

The questions were administered by a quiz program which presented the questions as a series of dialog windows (Figure 3). All of the questions were based on the ANES 2004 data set.[5] The program recorded the correctness of the user's answer, as well as the elapsed time for each question.

3.1.2 Subjects

Thirty (30) participants were recruited from among the student body at our university. Participants received no remuneration for their involvement in this study.

For privacy reasons, we collected no demographic information about individual participants, however our student body is one of the most international in the United States, with nearly half of the students coming from the Pacific Rim. All participants were fluent in the English language.

3.1.3 Procedure

The questions were presented as a series of 60 dialog windows, three for each of the 20 images. Each image typically compared between 2-5 values per attribute. The same question set was given to all participants. The first question for each image was qualitative, asking the user simply to compare two attributes, and identify which was larger. The second and third questions for each image were quantitative; asking the user to correctly select the values for attributes shown in the charts (Figure 3). The same image used for the qualitative question was also used for the qualitative questions. Participants completed the exercise individually, not as a "group project." The repeated order of appearance of each visualization was: Side by Side, Overlapping, Concentric, Animated.

3.1.4 Results

The results of our experiment are shown in Figures 4 and 5. This experiment was a preliminary study, and as such, it is difficult to draw definitive statistical conclusions from it. Nevertheless, our initial findings do suggest that the animated approach is more effective, in many cases, than the others. Figure 4 shows that the participants answered the questions with the animated visualization an average of 1 to 3 seconds faster than with the other visualizations. The animated visualization also performed favorably with respect to accuracy. Participants made anywhere from 3% to 4% fewer mistakes when they used the animated visualization. The raw percentages are shown in Table 1.

One might justifiably blame the occlusion artifacts inherent to the overlapping visualization for its low scores in both speed and accuracy. The side by side visualization, interestingly, performed reasonably well in terms

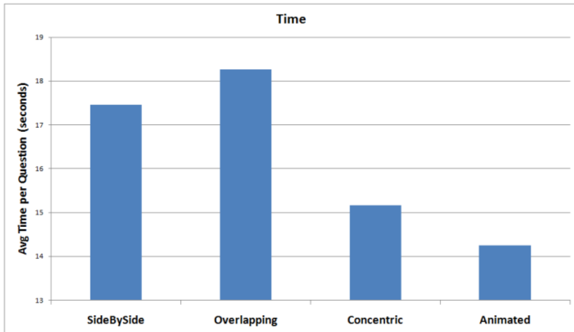


Figure 4. On average, participants tended to answer questions more quickly using the animated "visual diff" than with the other visualizations.

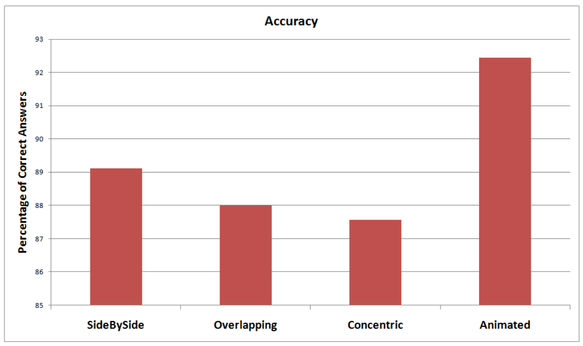


Figure 5. Respondents' accuracy was somewhat higher when using the animated visualization than with the other "visual diff" strategies.

of accuracy, but not in time. This is likely due to the fact that, while a side-by-side comparison of two charts is fairly unambiguous and straightforward to interpret, the reader must visually scan between the two, thus increasing the response time. In contrast, the concentric visualization was fast, but inaccurate. Perhaps the compactness of the concentric visualization led respondents to mistake the values in one ring for another. The animated visualization, however, seemed to strike the right balance between compactness and usability; thus, it is the visualization that we chose to include as the default "visual diff" in SQiRL.

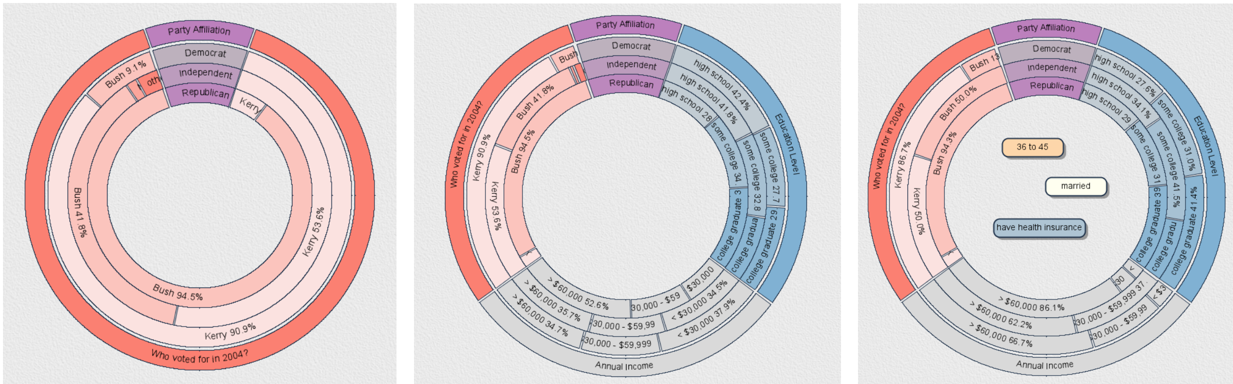
4. RADIAL CROSSTABULATION

Our next visualization is more advanced, and is intended for users with previous data analysis experience. It is based on crosstabulation, a common tool for statistical analysis.[12] In a crosstabulation (or *crosstab*), two variables are displayed on the axes of a table. Each row and column represents a different possible value for that variable. Each cell displays the frequency in which the combination of the values shown in the corresponding row/column occurs.

A crosstab in its basic form displays relationships between two and only two variables. Additional relationships can also be shown via the addition of control variables, yet the power gained with the addition of control variables

Table 1. Comparison of radial "visual diff" strategies (Empirical)

	Accuracy		**Time per Question**	
	Mean	σ	**Mean**	σ
Side by side	89.11%	14.489	17.46 sec	6.209
Partial overlap	88.0%	15.988	18.26 sec	8.757
Concentric rings	87.56%	18.915	15.16 sec	6.736
Animated	92.44%	13.07	14.25 sec	5.006



(a) (b) (c)

Figure 6. (a) Very simple radial crosstab. The "rows" are shown in the wedge at the top of the ring, while the "columns" are projected radially around the circumference. (b) More advanced radial crosstab. Multiple variables are projected around the circumference, creating a series of stacked bar charts in every ring. (c) By introducing additional constraints (control variables) into the interior of the ring, we can further specify the subpopulation being analyzed. The sectors in each ring update to their correct sizes via a smooth animation to help the user maintain context.

comes at the expense of a more complicated table. To address this issue, we propose a *radial crosstab* visualization, which combines the expressive power of multivariate crosstabs with the simple drag-and-drop usability of SQiRL. In contrast to traditional crosstabs, a radial crosstab supports the addition of many control variables with equal clarity.

Example radial crosstabs are shown in Figure 6. The "row" labels are shown in the top-center of the ring, and the "columns" are projected radially around the circumference. For improved readability, the column labels are printed in each sector, rather than in the column header only. Although radial crosstabs allocate less screen space per variable, our use of stacked bar charts ensures that the most prominent values for each variable will still be visible. As always, "mousing over" any portion of the chart reveals full details about a given value or variable.

While it is possible to construct radial crosstabs involving only 2 variables (see Figure 6a), the real power of a radial crosstab is manifest when several variables are to be examined simultaneously. Let us say that we want to evaluate the relationship between one's affiliation with a certain political party and a number of other variables: annual income, education level, and how one voted in the 2004 general election. Using traditional methods, this would require creating three separate crosstabs. The same information can be shown in just one radial crosstab diagram (Figure 6b). As in the basic interface, additional query terms may be placed into the ring's interior to further constrain the subpopulation of interest (Figure 6c).

The simplicity of the radial crosstab is, perhaps, its greatest strength. Creating crosstabs with current popular statistics software is a multi-step procedure, requiring the navigation of a series of menus and dialogs. Once a crosstab is created, it is static and non-interactive. In contrast, the radial crosstab method supports the creation of charts with similar expressive power as traditional crosstabs, via a straightforward "one step" drag-and-drop interface. Furthermore, radial crosstabs are fully interactive. Variables can be quickly and easily added and removed from both the circumference and the interior, and the chart updates in real time. Hence, this visualization scheme encourages a more fluid workflow in which hypotheses can be rapidly tested and iteratively revised.

Radial crosstabs do share an unfortunate side-effect common to all ring-based visualization systems. Due to the circular shape of the crosstab, the outer rings necessarily have a greater circumference than the inner rings. This difference is due to the fact that arc length is derived from the radius as well as the angular magnitude. This geometric side effect has potential consequences on the way that a radial crosstab is read and interpreted, namely, those sectors on the outer rings may receive undue emphasis due to their greater radius. Thus, this issue is not limited to SQiRL alone, and affects any radial visualization system that uses concentric rings. See, for example, Refs. 13–15.

5. DISCUSSION AND FURTHER WORK

In this paper, we have presented three complementary visualization techniques as implemented in SQiRL, a prototype software system for visual data analysis. We began by reviewing SQiRL's direct manipulation metaphor for querying and visualizing tabular data. We then introduced two extensions to this basic visualization method. First, we presented a "visual diff" scheme, along with the results of a user study which suggests that animation is a useful technique for quickly and accurately comparing two queries. Second, we introduced a more general visualization for comparing many variables simultaneously, based on the familiar metaphor of crosstabulation. Each of these three visualizations builds upon the other, allowing the user to progress from simple to more advanced visualizations as he or she becomes familiar with the tool.

This three-tiered user interface corresponds nicely to Bertin's suggestion that there are three ways to read a diagram: *Read Fact*, *Read Comparison*, and *Read Pattern*.[16] In the SQiRL system, basic queries allow users to find facts quickly. The "visual diff" interface lets users compare two different queries. The more advanced "radial crosstab" interface lets users find patterns across three or more variables.

In our implementation, each visualization canvas opens in a new tab, clearly separating the novice, intermediate, and advanced modes of operation. This has at least two important advantages. By making the user explicitly request a more advanced mode, we preserve a simple default user experience for beginners. Additionally, if a beginning user does inadvertently access an undesired mode, he or she can revert to the original interface simply by re-selecting the original tab.

There are several avenues for future work in this area. For instance, while we believe radial visualization to be a compelling and valuable methodology, Cleveland[17] argues that users cannot accurately judge angles in radial graphs. Thus, it remains to be seen whether the interactive techniques discussed in this paper would not apply equally well to more traditional visualization schemes such as linear bar charts or line graphs. The study by Diehl et al.,[18] which compares users' ability to recall the location of items in radial versus rectangular diagrams, is an excellent first step, but more work remains to be done.

In addition, one valid criticism of radial visualization techniques in general is that the nodes near the periphery of the display are rendered very small.[19] This could be overcome in SQiRL by use of a fish eye lens.[20,21]

Finally, SQiRL currently has no straightforward way of "going back" to an earlier state of the program and viewing past queries. The importance of *computational provenance*,[22] the ability to trace a computational process over time, has attracted considerable interest in recent years.[23] As a future extension, an interface such as the one described by Callahan et al.[24] could be adapted to seamlessly and automatically maintain a record of previous queries, and allow the user to revisit any previous state.

REFERENCES

1. B. Shneiderman, "Session 2: "InfoVis for the Masses"," *IEEE Transactions on Visualization and Computer Graphics* **13**, p. iii, Nov. 2007.
2. D. A. Norman, *The Design of Everyday Things*, p. 209. Basic Books, 2002 ed., 2002.
3. G. M. Draper and R. F. Riesenfeld, "Who Votes for What? A Visual Query Language for Opinion Data," *IEEE Trans Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2008)* **14**, pp. 1197–1204, Nov-Dec 2008.
4. C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, and C. T. Silva, "Querying and creating visualizations by analogy," *IEEE Transactions on Visualization and Computer Graphics* **13**(3), pp. 1560–1567, 2007.
5. National Election Studies, "THE 2004 NATIONAL ELECTION STUDY." www.electionstudies.org. Accessed 11 November 2008.
6. W. Playfair, *The statistical breviary; shewing, on a principle entirely new, the resources of every state and kingdom in Europe; illustrated with stained copper plate charts, representing the physical powers of each distinct nation with ease and perspicuity. To which is added, a similar exhibition of the ruling powers of Hindoostan*, London: for J. Wallis, 1801.

7. M. L. Northway, *A Primer of Sociometry*, University of Toronto Press, 2nd ed., 1967.
8. G. M. Draper, Y. Livnat, and R. F. Riesenfeld, "A Survey of Radial Methods for Information Visualization," *IEEE Transactions on Visualization and Computer Graphics* **15**, pp. 759–776, 2009.
9. J. Stasko and E. Zhang, "Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations," in *InfoVis 2000*, pp. 57–65, 2000.
10. P. Pirolli and S. Card, "Information foraging in information access environments," in *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 51–58, ACM Press/Addison-Wesley Publishing Co., (New York, NY, USA), 1995.
11. T. Jankun-Kelly and K.-L. Ma, "Visualization exploration and encapsulation via a spreadsheet-like interface," *IEEE Trans Visualization and Computer Graphics* **7**, pp. 275–287, July 2001.
12. M. J. Norušis, *SPSS 15.0 Guide to Data Analysis*, Prentice Hall, 2006.
13. M. Ankerst, D. A. Keim, and H.-P. Kriegel, "Circle segments: A technique for visually exploring large multidimensional data sets," in *IEEE Visualization '96, Hot Topic Session*, 1996.
14. M. J. Mohammadi-Aragh and T. Jankun-Kelly, "Moiretrees: Visualization and interaction for multi-hierarchical data," in *EuroGraphics - IEEE VGTC Symposium on Visualization*, K. W. Brodlie, D. J. Duke, and K. I. Joy, eds., pp. 59–66, 2005.
15. D. A. Keim, F. Mansmann, J. Schneidewind, and T. Schreck, "Monitoring network traffic with radial traffic analyzer," in *2006 IEEE Symposium On Visual Analytics Science And Technology (VAST)*, pp. 123–128, 2006.
16. S. K. Card, J. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann, 1999.
17. W. S. Cleveland, *The Elements of Graphing Data*, Hobart Press, revised ed., 1994.
18. S. Diehl, F. Beck, and M. Burch, "Uncovering strengths and weaknesses of radial visualizations—an empirical approach," *IEEE Transactions on Visualization and Computer Graphics* **16**, pp. 935–942, 2010.
19. J. Lamping and R. Rao, "The hyperbolic browser: A focus+context technique based on hyperbolic geometry for visualizing large hierarchies," *Journal of Visual Languages and Computing* **7**, pp. 33–55, 1996.
20. G. W. Furnas, "Generalized fisheye views," in *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 16–23, ACM Press, (New York, NY, USA), 1986.
21. M. Sarkar and M. H. Brown, "Graphical fisheye views of graphs," in *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 83–91, ACM, (New York, NY, USA), 1992.
22. C. T. Silva and J. E. Tohline, "Computational provenance," *Computing in Science and Engineering* **10**, pp. 9–10, May-June 2008.
23. J. Freire, D. Koop, E. Santos, and C. T. Silva, "Provenance for computational tasks: A survey," *Computing in Science and Engineering* **10**(3), pp. 11–21, 2008.
24. S. Callahan, J. Freire, C. Scheidegger, C. Silva, and H. Vo, "Towards Provenance-Enabling ParaView," in *Proceedings of the 2nd International Provenance and Annotation Workshop (IPAW 2008)*, 2008.