Dynamic and Interactive Dimensional Anchors for Spring-Based Visualizations

Alexander G. Gee [*a], Min Yu [a], Hongli Li [a], Georges G. Grinstein [a]
[a] University of Massachusetts Lowell, Computer Science, 198 Riverside St., Olsen Hall Rm. 301F, Lowell, MA 01854

ABSTRACT

Modern visualization techniques are more than the display of data; they include appropriate interactions for exploratory analysis. While there are a number of visualization techniques to choose from, only a few support the ever-growing dimensionality of modern data sets. Radviz is a visualization technique that can handle very high-dimensional data sets and display thousands of dimensions simultaneously. This paper highlights new tools facilitating its interpretation and usage. Two major problems with high-dimensional visualization techniques, including radviz, are selecting the placement or layout of the dimensions and determining the important or more meaningful dimensions. In this paper, we review the radviz display paradigm and provide some intuitive guidelines for display interpretation. We then introduce new interactions for manipulating radviz layouts. These interactions support the dynamic layout of the dimensions within the display while providing informative feedback about the projected data. We define a number of interactive operators to provide insights into the underlying data. Finally, we generalize the radviz circular layout and its interactive operators to support unconstrained or freeform layouts, which offer greater flexibility and control for general data explorations.

Keywords: information visualization, interaction, multidimensional visualization, large data set visualization

1. INTRODUCTION

Visualization provides techniques and tools for transforming data (e.g., raw text and numbers) into information (e.g., understanding and insight). Card, MacKinlay and Shneiderman define visualization as "the use of computer-supported, interactive, visual representations of data to amplify cognition"[1]. In the past 20 years, visualization has been widely applied in a number of different application areas, including physical simulation, business analysis, data mining, and bioinformatics, because it offers graphical approaches to effectively present information for data understanding and knowledge discovery[2].

Information visualizations attempt to efficiently map data variables onto visual dimensions in order to create graphic representations. Visualized data come from many different sources including retail sales revenues, document collections, clinic trials, and biological datasets, each possessing very distinct features. The data may be a small data set with only a few records or a large data set with thousands of parameters. High-dimensional data are challenging because visualizing the data involves the critical issue of rendering a large number of dimensions in a limited display area (usually a two-dimensional display screen)[3]. High-dimensional visualization techniques to tackle this problem include parallel coordinates[4], circle segments[5], and value and relation displays (VaR)[6]. Hoffman proposed an approach known as radviz that places all data dimensions, called dimensional anchors, on the perimeter of a circle; in that way several thousand dimensions (and in fact many more) can be displayed simultaneously[7]. Radviz has been used in a number of applications[8-9].

However, in the standard radviz layout algorithm, the resulting display strongly depends on the ordering of the dimensions, and thus lacks what we call visual consistency. In other words, given the same data set, the arrangements of

[*] agee@cs.uml.edu; http://ivpr.cs.uml.edu

dimensions in different orders may generate significantly different output patterns, and hence, makes the interpretation of the data more difficult.

In this paper, we introduce a set of interactions for manipulating the dimensions in a radviz display to support dynamic querying. We present a number of new interactive operators for controlling and modifying the visual display and extending the traditional radviz circular layout by supporting unconstrained or freeform spring-force layouts, which provide greater flexibility and control during data exploration.

The rest of this paper is organized as follows. A brief review of related research is presented in Section 2. In Section 3, we review the radviz visualization technique. Section 4 presents insights into standard radviz projections. Section 5 discusses radviz interaction mechanisms for defining and manipulating radviz layouts along with the various interaction operators. Section 6 describes unconstrained spring-force layouts. Finally, a brief overview of open problems is presented in Section 7 with a summary in section 8.

2. POINT PLOTS AND HIGHER DIMENSIONAL VISUALIZATION TECHNIQUES

We focus on point plots most relevant to our discussion of radviz and its generalizations. Parallel coordinates and other high-dimensional visualizations are described in[7]. Most of these visualizations will also be able to benefit from the techniques we describe in this paper.

The most common visualization techniques are the statistical graphics techniques, including scatterplot, box plots, and histograms. These techniques are designed for presenting data in lower-dimensional data spaces. For example, the standard scatterplot easily displays five dimensions – x-position, y-position, color, size, and shape. To support larger data spaces, the standard scatterplot has been extended in several ways, the most common being multiple instances (i.e., rugplots and scatterplot matrices[10]), linear projections (i.e. projection pursuit[11]), nonlinear projections (i.e., principal component analysis (PCA)[12], Sammon plots[13], and multidimensional scaling (MDS)[14]), and complex shape representations (i.e., Chernoff faces[15], star glyphs[16] and iconographics[17]).

A number of other point plot visualization techniques can display high-dimensional data spaces. Pixel oriented visualizations (i.e., circle segments[18]) represent as much of the data as possible on the display space by mapping each data value to a display pixel and arranging the pixels adequately. Jing Yang et al. introduced value and relation (VaR) displays as the combination of MDS and pixel-oriented techniques to display large data spaces[6].

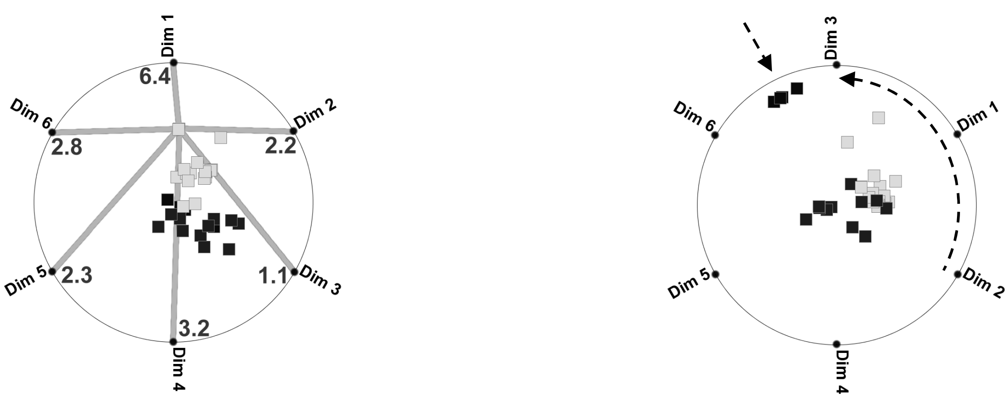

Figure 1a and 1b: Two radviz displays projecting a six-dimensional data space to the two-dimensional display space. In figure 1a, the green point represents one record and is connected by lines representing springs. This record is positioned where the sum of the forces is zero. This highlights the spring-based layout algorithm. Figure 1b uses the same data; however, dimension three has been moved from the lower right to the top and center. This reordering of the dimensions reveals a connection between dimension three and dimension six for the blue record glyphs.

3. RADVIZ

Radviz uses a spring-force analogy for projecting data points or records. The positioning of records is defined by the pull from a set of dimensions arranged around a circle within a two-dimensional display space. Individual records are then visually encoded using standard glyph representations with well-known perceptual properties (i.e., color, size, shape, opacity, etc.). Figure 1 displays two radviz projections of a six-dimensional data space.

Like glyphs representing the graphical version of records, the dimensions are encoded as dimensional anchors (DAs) that define the projection. Thus, radviz displays are specified by the arrangement of dimensional anchors within the display space. Whereas the standard scatterplot axis is the graphical mapping of a data variable, dimensional anchors provide constructs for prescribing the projection and the interaction between other dimensional anchors (see Hoffman[19] for further details).

In the case of radviz, all dimensional anchors prescribe a spring-force projection. All dimensional anchors are constrained to a circle defining the radial structure of a radviz display. Records are positioned in the display within the circle where the computed pull from all dimensional anchors equals zero. This pull is determined using virtual springs attached from individual records to all dimensional anchors (illustrated in figure 1a). These springs are encoded using Hooke's law: f=kd, where f is the resulting force, k is the spring's constant and d is the length of the spring – distance between to points. For a radviz display, k is defined as the record's values for the j dimension mapped to the h dimensional anchor, and where $0 \leq k \leq 1$.

There are a number of interesting facts to note when interpreting a radviz display. Most of these can be seen in figures 1a and 1b. First, dimensions with higher record coordinate values will pull the data points closer to their associated DAs on the circumference of the circle (these are interesting points with lots of information), for example the blue glyphs in figure 1b. Data points with approximately equal values for all dimensions will lie close to the center (these are uninteresting points with low information content).

Even though radviz is powerful, it has two critical problems: mapped points may overlap with dramatically different coordinates and the layout of the dimensions will significantly affect the relative positions of the data points (figures 1a and 1b). The overlapping simply says that radviz displays are lossy. For example, points in the center may be pulled by any two opposite DAs or may have all their coordinates equal (again the blue glyphs in figure 1a and 1b). Almost all visualization techniques of high-dimensional data exhibit this problem (three-dimensional scatterplots, principal component projections, and multi-dimensional scaling projections) as they all project high-dimensional data to typically two or three-dimensional displays. The layout problem is also universal in other high-dimensional visualization displays. If a point has two equal coordinates and these two dimensions are first opposite each other on the circle, the forces cancel; hence, the point lies in the circle's center. However, if the dimensional anchors are moved next to each other the point will move toward the DAs, closer to the circle (see dimensions 3 and 6 and their effect on the blue glyphs in figure 1b). This effect from different DA layouts, is called radviz oscillation, and makes the understanding and interpretation of the displays difficult. It is to overcome both the lossy mapping and such oscillations that we introduce several special interactions for dynamically manipulating dimensional anchors.

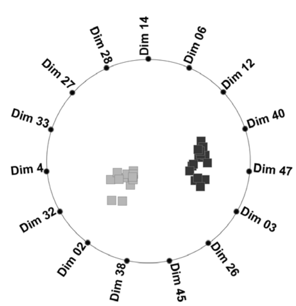

Figure 2: Radviz display showing two record clusters.

4. RADVIZ PROJECTIONS

Unlike the Euclidean projections of scatterplots that are easily understood, radviz projections require some initial explanations to appreciate their elegance and power. For example, how should we interpret the following radviz display?

Is this a real separation as it would be in scatterplots? The answer is yes. How about if points are in one cluster, does that mean that they cannot be separated? The answer, just like in scatterplots, is no. It depends on the other dimensions.

Additional facts or theorems can be proven easily since the radviz mapping is a linear transformation. For example, n-dimensional lines in the original data space map to two-dimensional lines in radviz, and convex sets map to convex sets in radviz[20]. Furthermore, if two vectors in the n-dimensional data space are mapped to the same radviz point then any linear combination of these two vectors is mapped onto the same point in radviz. Finally, lines that pass through the origin pass through the center of the radviz projection.

Additionally Hooke's law can be formulated as a single complex exponential for mathematical ease or as two dot products (the corresponding real and imaginary parts) for computational simplicity. Since this formulation involves are dot products, close points in radviz are angularly close in the original space (see Hoffman[19] for details).

One approach for explaining how radviz works is by projecting known data. The two cones defined and illustrated in figure 3 highlight some key points. Figure 4 provides the most important information about radviz, that projections of cones whose tip is at the origin of the original data space are cylinders in the final display. Furthermore, there is a strong relationship between the cosine of the angle in the original data space to the resulting projection. Figure 5 provides two additional projections of two different cones, one from the origin in a three-dimensional data space, and another cone offset from the origin in a six-dimensional data space.

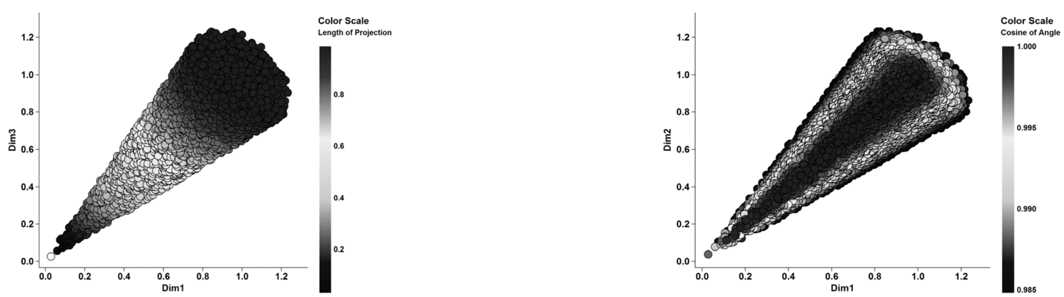

Figure 3: Two-dimensional scatterplot projections of a three-dimensional cone defined by one thousand randomly generated points within a three-dimensional volume defined to be no greater than 10° from the base vector relative to the cone's tip that starts at the origin (0, 0, 0) and ends at the point (1, 1, 1), and has a flat circular centered end. The left image displays the points colored by the length of projection from the cone's tip, and are drawn from shortest (blue) to longest (red); consequently, we are seeing all the points on the bottom of the cone. The right image displays the points colored by the cosine of the angle to the projection, and are drawn from widest (blue) to closest (red) revealing the mapping of inner points throughout the cone.

Figure 4: Two radviz displays showing the results of projecting the three-dimensional cone defined in figure 3 (both shown using a 700% relative projection zoom). The left image displays the points colored by the length of the projection. The left image shows that

there is an even distribution of fixed color points across the whole surface of the display, indicating that radviz projected the cone as a cylinder with a singularity at the origin. This result is further confirmed in the right image that displays points colored by the cosine of the angle.

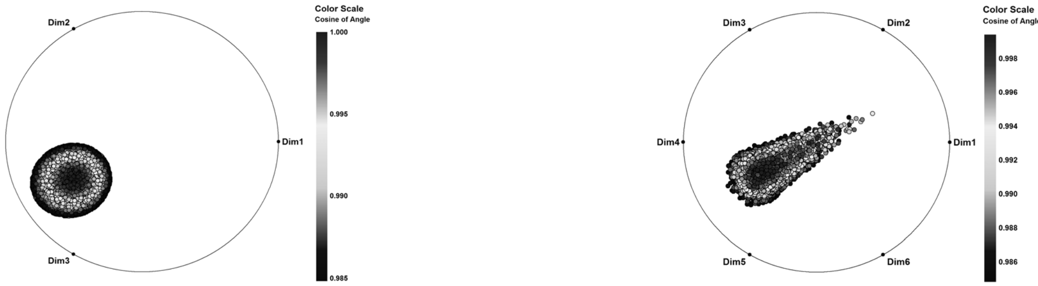

Figure 5: Two additional radviz displays of cones. The left image is the projection of a cone with a top at the origin (0, 0, 0) and base at (1, 2, 3). This image is important because it too exhibits the same cylindrical projection illustrated in figure 3 for a cone whose tip is at the origin (shown here using a 200% relative projection zoom). The right image displays a projection of a six-dimensional cone with top at (1, 1, 0, 0, 0, 0) and base at (0, 0, 1, 1, 1, 1) (shown using a 200% relative projection zoom).

5. POSITIONING DIMENSIONAL ANCHORS

The original radviz used a circular arrangement of uniformly distributed DAs so as to treat each mapped dimension equally when compared against all other mapped dimensions. This has proved to be limited, and it is often necessary to group dimensions closer to one another to highlight certain points.

There are three approaches to arranging DAs: user-defined orderings, computed orderings and dynamic reordering.

5.1 Computed Orderings

Algorithmic approaches to arranging the dimensions provide one way to deal with larger sets of dimensions. Here an algorithm computes the selection of dimensions, the mapping of dimensions to dimensional anchors, and the final positioning of dimensional anchors within the layout. These approaches compute the relative importance of dimensions for a user defined goal, and lay the dimensions out based on some criteria (such as to maximize cluster distances).

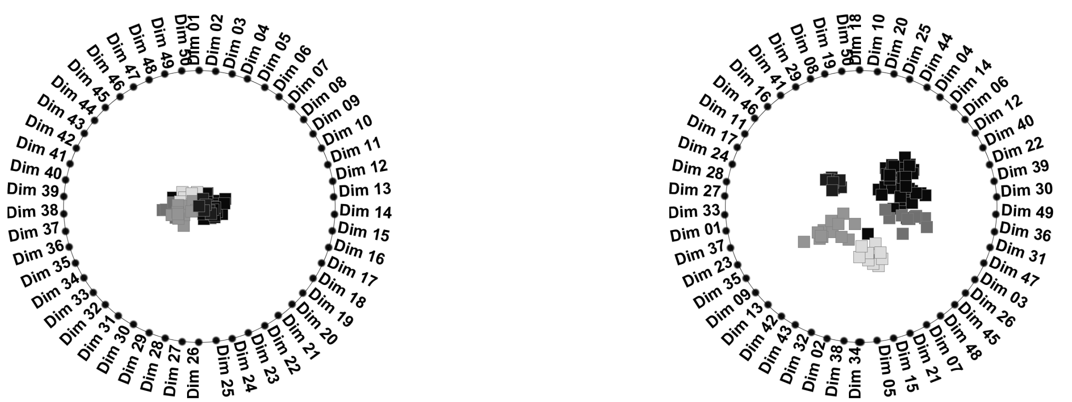

Figure 7: Here are two radviz displays of the same data space. The image on the left displays dimensions in their original default order, and is unable to separate the clusters of records. However, in the right image the dimensions have been reordered using a class distinction algorithm to maximize the display of record classes.

In the case of radviz, t-statistics and distribution probabilities based on class distinctions between dimensions are often used to spread cluster groupings[9]. The goal is to maximize the selection and arrangement of dimensions to segregate records based on class membership. First, the user selects a dimension that defines classes of records, with the aim to understand which dimensions are more influential to a given class of records for all classes defined in the selected dimension. Next, a decision is made to screen dimensions out using t-statistic values, probability values, or raw dimension counts per class. The algorithm calculates the t-statistics and probabilities for each dimension across all records for each class against all other classes. The result of this computation is a table of metadata describing the influences of each dimension for each class. The dimensions are then selected based on the chosen screening criterion and mapped to dimensional anchors. The radviz circle is divided into equally sized arcs, one for each class, and the DAs per class are arranged within each arc from most influential down in a counter-clockwise orientation (see figure 7).

While arrangements for class distinction are extremely useful in understanding relationships between dimensions and records, we have used a number of other possible computed arrangements. For example, one algorithm maximizes record placement for outlier detection, while another minimizes record obscuration to maximize screen utilization.

5.2 Dynamic Reordering

While the manual and automatic approaches for defining radviz layouts are important, the resulting DA arrangements are static, that is the DAs cannot be manipulated or queried. We often want to know which dimensions are most influencing the resulting point plots.

We extend the classic notion of direct manipulation to DAs, by providing interactions that affect their arrangements. Direct manipulation is one way to tie the actions of users directly to the underlying data through the visualization, rather than via indirect controls usually defined by visualization property sheets. Thus, radviz includes two direct interactive states, one for working with records and another for working with dimensional anchors.

In the record-interaction state, radviz provides for selecting, brushing and querying. Various selection and brushing tools are provided to highlight and identify groups of records in linked visualizations.

In the DA-interaction state, users can select, brush and query DAs in radviz. By changing the interactive state of radviz from record mode to a DA mode, users are able to directly manipulate the layout of dimensions and obtain valuable intuition about the projection via appropriate feedback.

Dimensional anchors are displayed as glyphs, much like the glyphs used to represent records. These graphical objects specify where a dimension is located. Like records, DAs can be selected and identified using standard visualization rendering techniques. However, the underlying information is not a collection of values, but a set of records.

5.3 Manipulating Individual Dimensional Anchors

By selecting individual DAs, users can interactively drag the anchor around the display constrained by the bounding circle. The key to working with a single dimensional anchor is its relationship to the unselected anchors, which define the projection. We commonly use four modes: overlapping, avoidance, switching, and bounded.

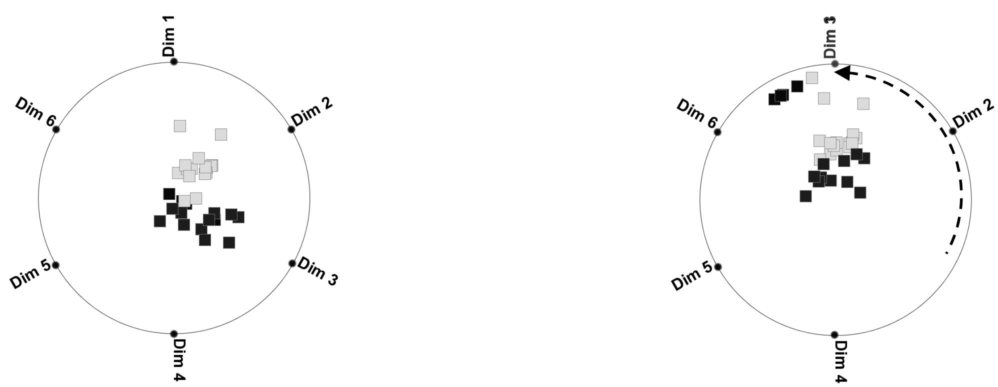

Figure 8: The right radviz display is the result of dynamically manipulating a single dimensional anchor (highlighted in red) using the overlapping interaction operator. In this scenario, dimensional anchors can overlap other anchors.

The overlapping mode is the most general of the four (illustrated in figure 8). Here, the selected dimensional anchor is free to be placed anywhere around the circle, even on top of other existing DAs. The placement of multiple DAs at a single position suggests that the mapped dimensions all define the same relationship across the records and will pull the records that much stronger to this position.

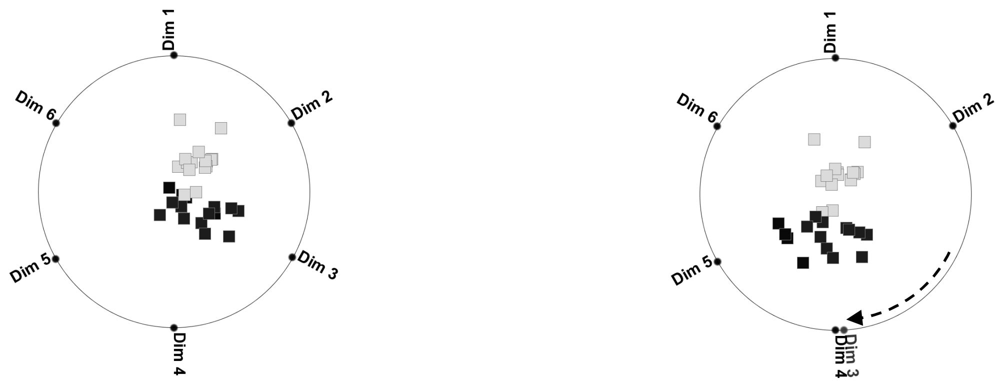

Figure 9: The right radviz display shown here is the result of dynamically moving a single dimensional anchor (highlighted in red) using the avoidance interaction mode. In this mode, selected dimensional anchors are not permitted to overlap unselected anchors and are required to map to unique display space locations.

The avoidance mode is an adaptation of the overlapping case (shown in figure 9). Here all dimensional anchors are positioned at unique locations. As there are a finite number of pixels on the screen but an infinite number of positions around the circle, the actually radviz display may render dimensional anchors on top of each other for large numbers of mapped dimensions, but the underlying mathematical position of the anchor is unique.

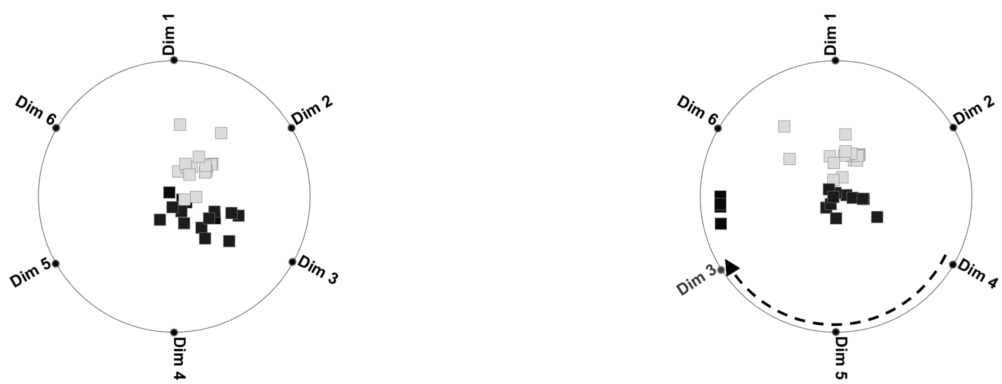

Figure 10: The right radviz display is the result of a single dimensional anchor (highlighted in red) dynamically manipulated by using the switching interaction operator. In this scenario, movement of dimensions is executed by switching the display space positions of a selected DA with an unselected DA in the direction of user control.

The switching mode is an example of a special case (see figure 10). Here the original placement of dimensional anchors is maintained. Movement of the selected dimensional anchor will switch places with the neighboring anchor in the direction of motion. Hence, the anchors are switched between positions with a snapping action. In this way, an initial pattern of anchors can be defined and the interaction of moving an anchor simply adjusts the mapped dimension to the anchor.

The bounded mode is a second special case (demonstrated in figure 11). Here the selected dimensional anchor is confined between the existing anchors surrounding it. That is the order of the DAs remains constant. As the user drags the selected anchor around the circle, so do the anchors in front of the selected anchor. Anchors ultimately bunch up as more and more anchors are pushed around the circle. Furthermore, a minimum distance between anchors may be specified helping deal with overlapping bunched anchors in the case of many mapped dimensions.

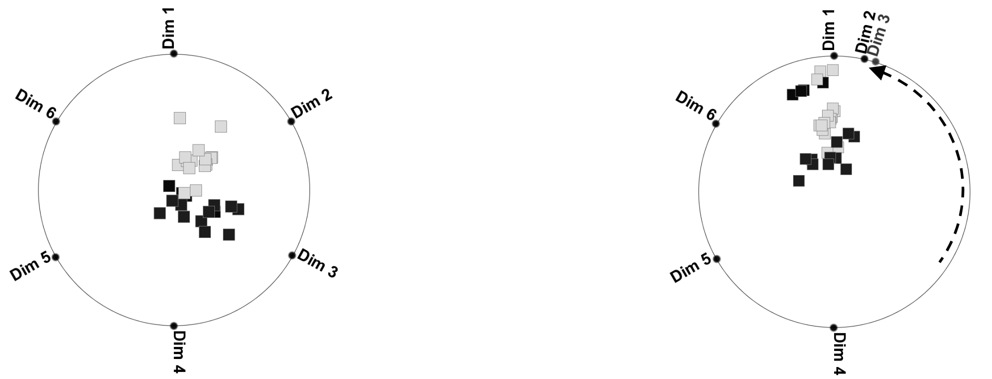

Figure 11: The right radviz display is the result of dynamically manipulating a single dimensional anchor (highlighted in red) using the bounded interaction mode. In this case, the order of the dimensional anchors does not change; consequently, movement of a select dimensional anchor moves adjacent anchors in the direction of user control.

5.4 Adjusting Multiple Dimensional Anchors

It is also possible to adjust multiple DAs as one group. Directly manipulating a single anchor provides visual feedback as to how records are related to individual dimensions, so does interacting with multiple anchors reveal relationships of records to dimensions. Moreover, there are times when dragging multiple DAs results in the clear definition of record groups, which appear as visually identifiable layers based on motion perception.

The selection of multiple DAs provides additional possibilities for working with dimensions and for customizing layouts. In this case, a decision needs to be made as to how to handle the spacing between multiple selected anchors and their interaction with unselected DAs. In addition to the four scenarios for single dimensional anchors, we include unbounded and bound group interaction modes.

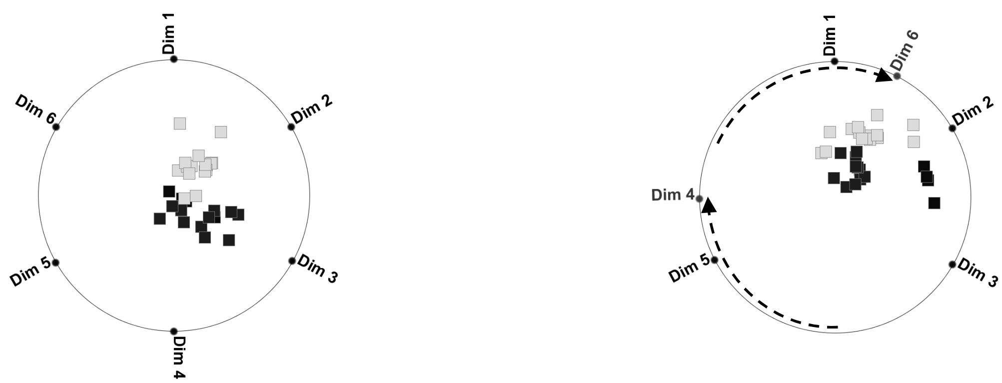

Figure 12: The right radviz display is the result of manipulating multiple dimensional anchors (highlighted in red) as a group using the unbounded group operator. The selected dimensional anchors will maintain their relative distances between each other during interactions. This unbounded group operator works in conjunction with interaction modes specified for single manipulated dimensional anchors.

The unbounded group interaction mode is the general case (see figure 12). In this mode, the selected DAs move together but attempt to maintain their distance between each other within the context of the unselected DAs. The selected DAs also follow the same four scenarios as defined for single DA adjustments.

In the bounded group interaction mode, selected DAs attempt to minimize their distance between each other (see figure 13). This mode supports interactive grouping of dimensions and provides support for hypothesis testing as to the relationships between dimensions and record sets.

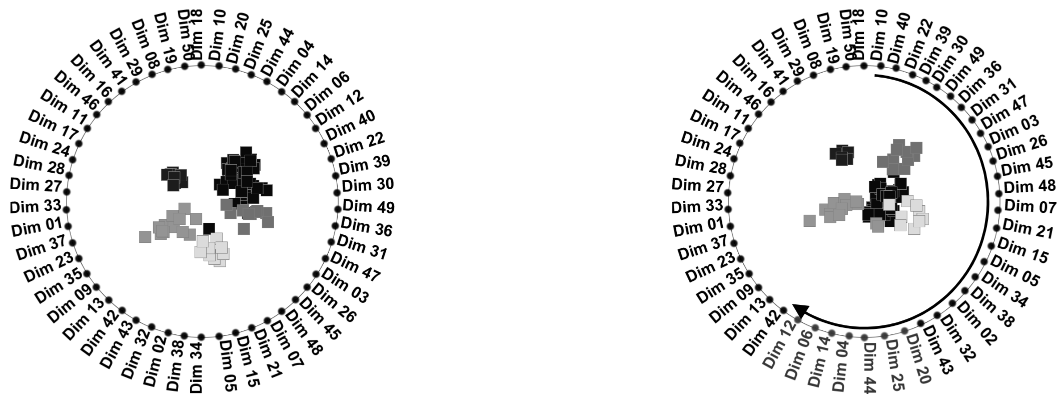

Figure 13: The right radviz display results from the dynamic manipulation of multiple dimensional anchors using the bounded group mode. In this scenario, the selected dimensional anchors are first brought together. Subsequent interactions are defined by the choice of single dimensional anchor interaction modes.

6. UNCONSTRAINED LAYOUTS

The original work on radviz by Hoffman provided other layouts of dimensional anchors using the same spring-force based definition[19]. Here the arrangements of dimensions were all defined by a small set of predefined patterns. Hoffman defined two other non-radial patterns: L-shape and grid. The L-shape pattern provided a way to specify scatterplot and scatterplot matrix displays. The grid layout was presented as an experimental layout showing the power of the spring-force based approach (see figure 14). Both patterns provide additional insight into how radviz works and how to define layouts beyond circles.

Figure 14: Two images showing unconstrained spring-based displays defined by a grid layout. The arrangement of dimensional anchors on the left is defined by the default order of the dimensions within the data set. The image on the right is the result of dynamic manipulations using a bounded group and switching interaction mode and has separated the data into two clusters.

It is clear that once we can move DAs around the circle, we are tempted and want to move them off the circle, especially away from the center (to increase its pull or effect on data). This led to the unconstrained layouts of DAs that start with an initial layout. Users are then able to drag individual anchors or groups of anchors to any position within the display to explore different projections (see figure 14).

Unconstrained layouts support the placement of dimensional anchors anywhere within a display. Users can drag and group dimensions based on their influences of separating or structuring the records within the resulting projection. This is an unconventional way to explore data as one is manually altering the projection rather than changing the set of projected dimensions. Rather than rapidly switching between projections of the data space via a small number of selected dimensions, these unconstrained layouts support the continuous adjustment of the projection that often reveal hidden relationships with the data. This process is very similar to using projection pursuit within XGobi, which defines linear projections from of the high-dimensional display space to the two-dimensional display space, and also supports this notion of incremental adjustments[11]. For example selecting anchors that move a cluster can reveal an outlier, one not dependent on a specific set of dimensions.

There are several approaches to the initial layout of these anchors. The first is simply to select a uniform rectangular grid (similar to radviz having the DAs equally spaced around the circle). We have tried several others: placing all anchors along a linear line across the display, randomly place dimensions across the display. The most successful approach has been defining a context dependent layout, where the data and some knowledge of its structure suggests a layout.

7. OPEN PROBLEMS

For a given radviz layout, there are two standard questions to consider. First, which dimensions are most important for laying out the selected records? Second, for a specific dimensional anchor layout, which records do the selected dimensional anchors influence the most? Furthermore, are there ways to automatically define arrangements of dimensional anchors for unconstrained layouts?

One approach to indicating the importance of individual dimensions for a given selection of records is to adjust the perceptual properties used to display the associated dimensional anchors mapping individual dimensions. We are experimenting with rendering the DAs using variable opacity; where solid glyphs represent the strongest dimensions associated with the selected record set, while glyphs that are more transparent indicate less influence. In the presence of multiple record selections, dimensional anchors can be defined by a compound glyph that provides information about the dimension's influence for each selection.

Similarly, we envision a record-based feedback mechanism for answering the second question. In this situation, dimensional anchors are selected and the user would like to know which records receive greater influence. The opacity of an individual record's representative glyphs can be adjusted to highlight their degree of influence from selected DAs.

Finally, for larger data sets, we foresee the need for automated approaches for laying out dimensional anchors within unconstrained non-circular layouts. Like the t-statistic approach for arranging dimensions with radviz, other computations can be performed across records and dimensions to define appropriate layouts for answering common questions. It would be very convenient to provide methods for defining displays that maximize distances between record clusters or classes within a grid-like layout. Furthermore, it would be useful to have a computation that maximizes outlier placement or minimizes record overlapping within the same layout.

8. CONCLUSION

We presented the application of dynamic manipulations and accompanying operators for customizing radviz and other spring-force visualization arrangements interactively. The circular layout of radviz has clearly shown to be useful when looking at classes of records and understanding their relationships between records and dimensions. The aim of this work was to improve the flexibility of radviz by providing appropriate mechanisms for handling and querying higher-dimensional data spaces. A number of interaction operators were introduced defining how selected dimensional anchors interact with unselected ones in such ways as to provide additional tools for data understanding and exploration.

We then generalized radviz to unconstrained layouts. We extended the interactive anchor operators from radviz to these unconstrained ones adding more general layout computations.

We are not arguing that radviz and its extensions will discover clusters or other patterns. In fact, no visualization is capable of providing such information. What we have presented is a set of tools facilitating interacting and exploring data sets using these general spring-force visualizations.

A fundamental problem of high-dimensional visualization is overcoming information loss when mapping hundreds or thousands of dimensions to a two or three--dimensional display. Spring-force layout techniques with interactive ordering of anchors as described in this paper can make a major contribution in this area.

REFERENCES

1. S.K. Card, Jock D. Mackinlay and Ben Shneiderman (Eds.). *Readings in Information Visualization: Using Vision to Think*. San Francisco, California: Morgan Kaufmann, 1999.

2. W.S. Cleveland and R. McGilll. The Many Faces of a Scatterplot. *Journal of the American Statistical Association*, Vol. 79, No. 388, December, pp. 807-822, 1984.

3. G. Grinstein, M. Trutschl and U. Cvek. High-Dimensional Visualizations. In *Proceedings of the Visual Data Mining Workshop*, KDD, 2001.

4. A. Inselberg. The Plane with Parallel Coordinates. *Special Issue on Computational Geometry of the Visual Computer*, Vol. 1, No. 2, October, pp. 69-91, 1985.

5. M. Ankerst, D.A. Keim, H.P. Kriegel. Circle Segments: A Technique for visually Exploring Large Multidimensional Data Sets. In *Proceedings of IEEE Visualization* (San Francisco, CA), 1996.

6. J. Yang, A. Patro, S. Huang, N. Mehta, M.O. Ward, and E.A. Rundensteiner. Value and Relation Display for Interactive Exploration of High Dimensional Datasets. In *Proceedings of IEEE Symposium on Information Visualization* (Austin, Texas, October 10-12), pp. 73-80, 2004.

7. P. Hoffman and G. Grinstein. Multidimensional Information Visualizations for Data Mining with Applications for Machine Learning Classifiers. In *Information Visualization in Data Mining and Knowledge Discovery*, Morgan-Kaufmann Publishers, 2000.

8. G.G. Grinstein, C.B. Jessee, P.E. Hoffman, P.J. O'Neil, and A.G. Gee. High-Dimensional Visualization Support for Data Mining Gene Expression Data. In E.V. Grigorenko (ed.), *DNA Arrays: Technologies and Experimental Strategies*. Boca Raton, Florida: CRC Press LLC, Chapter 6, pp. 75-773, 2002.

9. J.F. McCarthy, K.A. Marx, P.E. Hoffman, A.G. Gee, P.J. O'Neil, M.L. Ujwal, and J. Hotchkiss. Applications of Machine Learning and High-Dimensional Visualization in Cancer Detection Diagnosis and Management. In A. Umar, I. Kapetanovic and J. Khan (Eds.), *The Applications of Bioinformatics in Cancer Detection*, Annals of the New York Academy of Sciences, Vol. 1020, May, pp. 239-262, 2004.

10. E. R. Tufte. *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press, 1983.

11. D.F. Swayne, D. Cook and A. Buja. XGobi: Interactive Dynamic Data Visualization in the X Window System. *Journal of Computational and Graphical Statistics*, 7 (1), 1998.

12. M.H. Gross. Subspace Methods for the Visualization of Multidimensional Data Sets. In L. Rosenblum, R. A. Earnshaw, J. Encarnacao, H. Hagen, A. Kaufman, S. Klimenko, G. Nielson, F. Post, and D. Thalmann (Eds.), *Scientific Visualization: Advances and Challenges*. London: Academic Press, Chapter 11, pp. 171-186, 1994.

13. J.W. Sammon, Jr. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, Vol. c-18, No. 5, May, pp. 401-409, 1969.

14. S.S. Schiffman, M.L. Reynolds and F.W. Young. Introduction to Multidimensional Scaling. New York: Academic Press, 1981.

15. H. Chernoff. *The Use of Faces to Represent Points in n-Dimensional Space Graphically*. Department of Statistics, Stanford University, Technical Report No. 71, 1971.

16. J.M. Chambers, W.S. Cleveland, B. Kleiner, and P.A. Tukey. *Graphical Methods for Data Analysis*. Belmont, California: Wadsworth Press, 1983.

17. R.M. Pickett and G.G. Grinstein. Iconographic Displays for Visualizing Multi-dimensional Data. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* (Beijing and Shenyang, People's Republic of China), pp. 514-519, 1988.

18. D.A. Keim. Designing Pixel-Oriented Visualization Techniques: Theory and Applications. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, No. 1, 2000.

19. P. Hoffman. *Table Visualizations: A Formal Model and Its Applications*. Dissertation, University of Massachusetts Lowell, 2000.

20. P.J. O'Neil. *RadViz: The Visual Data Mining Tool*. Technical Report, AnVil, Inc., Burlington, Massachusetts, 2000. (Unpublished internal document)