Value and Relation Display: Interactive Visual Exploration of Large Data Sets with Hundreds of Dimensions

Jing Yang, Daniel Hubball, Matthew O. Ward, *Member*, *IEEE Computer Society*, Elke A. Rundensteiner, *Member*, *IEEE Computer Society*, and William Ribarsky, *Member*, *IEEE Computer Society*

Abstract—Few existing visualization systems can handle large data sets with hundreds of dimensions, since high-dimensional data sets cause clutter on the display and large response time in interactive exploration. In this paper, we present a significantly improved multidimensional visualization approach named *Value and Relation* (VaR) display that allows users to effectively and efficiently explore large data sets with several hundred dimensions. In the VaR display, data values and dimension relationships are explicitly visualized in the same display by using dimension glyphs to explicitly represent values in dimensions and glyph layout to explicitly convey dimension relationships. In particular, pixel-oriented techniques and density-based scatterplots are used to create dimension glyphs to convey values. Multidimensional scaling, Jigsaw map hierarchy visualization techniques, and an animation metaphor named Rainfall are used to convey relationships among dimensions. A rich set of interaction tools has been provided to allow users to interactively detect patterns of interest in the VaR display. A prototype of the VaR display has been fully implemented. The case studies presented in this paper show how the prototype supports interactive exploration of data sets of several hundred dimensions. A user study evaluating the prototype is also reported in this paper.

Index Terms—Multidimensional visualization, high-dimensional data sets, visual analytics.

1 Introduction

Large data sets with hundreds of dimensions are common in applications such as image analysis, finance, bioinformatics, and antiterrorism. For example, in order to detect the semantic contents of large image collections, it is common to analyze hundreds of low-level visual attributes of the images. It is a challenge to make decisions based on these data sets, since they are hard to analyze due to the dimensionality curse [5], that is, the lack of data separation in high-dimensional space. Using multidimensional visualization techniques to present this data to analysts and allowing them to interactively explore and understand the data sets is an important approach to addressing this challenge. However, most traditional multidimensional visualization techniques suffer from visual clutter and only scale up to tens of dimensions. Up to now, few multidimensional visualization systems have claimed to be scalable to data sets with hundreds of dimensions. In this paper, we present such a system, called the *Value and Relation* (VaR) display, which is an improved version of a technique reported in an earlier paper [27].

Our work is based on multiple concepts proposed and explored in prior efforts toward visual exploration of large data sets in the Information Visualization field. They include:

- Using condensed displays to provide as much information as possible to users. Typical approaches include pixel-oriented techniques [12], [13] and density-based displays [9], [24]. For example, in pixel-oriented techniques, information is so condensed that each pixel presents a single data value.
- Examining relationships among dimensions to discover lower dimensional spaces with significant features. Example approaches include ranking low-dimensional projections by their features such as linear relationships [19] and placing dimensions in a layout revealing their relationships to help users construct meaningful subspaces [28].
- Providing a rich set of interactions to allow users to explore data sets from multiple coordinated views. In these views, different subsets of dimensions and/or data items can be examined at different levels of detail using different visualization techniques. Examples of such approaches include the Hierarchical Parallel Coordinates [10] and the VIS-5D system [11].

The concepts above are significant features of the VaR display since its initial version [27]. In the first version (see Figs. 1a and 1b), pixel-oriented displays were used to show

- *J. Yang and W. Ribarsky are with the Department of Computer Science, University of North Carolina, Charlotte, NC 28223. E-mail: {jyang13, ribarsky}@uncc.edu.*
- *D. Hubball is with the Department of Computer Science, University of Wales Swansea, Singleton Park, Swansea, SA28PP, United Kingdom. E-mail: csdan@swansea.ac.uk.*
- *M.O. Ward and E.A. Rundensteiner are with the Computer Science Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609. E-mail: {matt, rundenst}@cs.wpi.edu.*

Manuscript received 25 Nov. 2005; revised 6 July 2006; accepted 31 Oct. 2006; published online 3 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-0185-1105.
Digital Object Identifier no. 10.1109/TVCG.2007.1010.

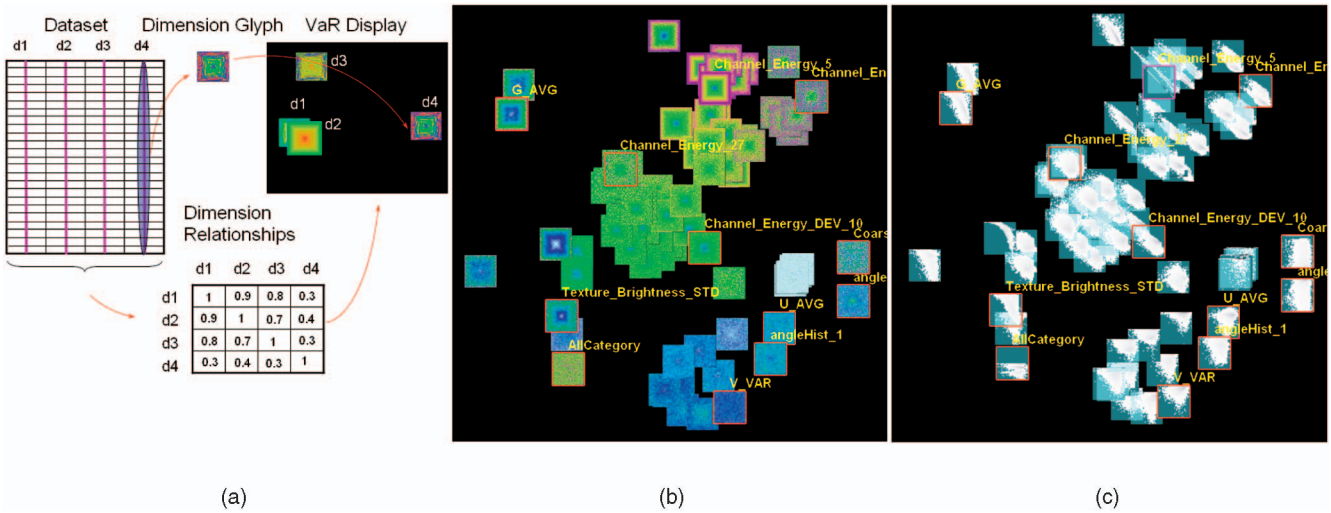


Fig. 1. (a) Illustration of the VaR display. On the left is the spreadsheet of a 4D data set with each column representing a dimension. At the bottom is a matrix that records the pairwise relationships (such as correlations) among the dimensions. In the middle is the glyph of the fourth dimension. On the right is the VaR display of the data set. (b) The Pixel MDS VaR display of the Image-89 data set (89 dimensions and 10,417 data items). (c) The X-Ray scatterplot MDS VaR display of the same data set.

data values and group them into dimension glyphs representing individual dimensions. The dimension glyphs were then positioned on the screen using a fast multidimensional scaling (MDS) algorithm [4] according to dimension correlations to reveal their interrelationships (dimension correlation is used since it is a typical measure of dimension relationships, but other relationship measures can also be used). A rich set of interactions was provided to facilitate navigation in the display and generate lower dimensional spaces of interest. To differentiate the first version from the improved version, we call it the Pixel MDS VaR display.

In the improved version of VaR presented in this paper, these features are significantly strengthened. A density-based scatterplot [9], [24] has been added to the system as an alternate approach to generating dimension glyphs. A Jigsaw map layout [23] and the Rainfall metaphor have been added into the system as alternate dimension glyph layout approaches. The new version also supports a broader range of interaction tools than the original version, including a new data item selection and highlighting tool. The labeling issue, which was ignored in the initial version, is addressed in this version. A case study is included in this paper involving the visual analysis of a data set with 838 dimensions. A user study comparing the VaR display with the rank-by-feature framework [19], [20] is also reported.

This paper is organized as follows: Section 2 reviews related work. Section 3 briefly introduces the original Pixel MDS VaR display. Section 4 presents the approach of using density-based scatterplots to generate dimension glyphs. Section 5 describes the new Jigsaw and Rainfall dimension glyph layout strategies. Section 6 summarizes the correlation calculation algorithm used in the VaR display. Section 7 presents the interaction tools. Section 8 addresses the labeling issue. Section 9 describes the implementation of the VaR display and addresses the scalability issue. Section 10 discusses visual exploration approaches with the VaR display. Section 11 presents a case study, and Section 12 presents a user study for the VaR display. Section 13 presents our conclusions and future work.

2 Related Work

Many techniques for generating condensed displays for large data sets exist. The work most related to our work is pixel-oriented techniques and scatterplots. Pixel-oriented visualization techniques [12], [13] are a family of multidimensional display techniques that map each data value to a pixel on the screen and arrange the pixels into subwindows to convey relationships. The patterns of the subwindows may reveal clusters, trends, and anomalies. Pixel-oriented techniques are one among several options to create the dimension glyph in the VaR display.

Scatterplots visualize 2D data sets or 2D projections of multidimensional data sets. In a scatterplot, there are a horizontal axis and a vertical axis, which are associated with two dimensions (X and Y). The data items are plotted onto the display according to their coordinates on X and Y. Scatterplots are widely used since they provide rich information about the relationship between two dimensions such as strength, shape (line, curve, and so forth), direction (positive or negative), and presence of outliers [18]. Density-based scatterplots [24], [9] scale to large data sets by using the intensity of the spot in a scatterplot to indicate the data density in that spot. We use the density-based scatterplot as an option for generating the dimension glyph and treat the areas with no data items in a scatterplot in a different way from existing approaches due to the possible overlaps among the scatterplots.

Scatterplots of multidimensional data sets are often organized together to show multiple 2D projections of the data sets. Scatterplot matrices [7] organize the scatterplots of all $N \times (N-1)/2$ 2D projections of an N-dimensional data set into a matrix. Scatterplot matrices easily get cluttered when the number of dimensions increases. Rather than displaying all 2D projections, we display N scatterplots between all dimensions and a focus dimension and

position them in a manner conveying dimension relationships in our density-based scatterplot VaR option.

There exist multiple visualization approaches to examining relationships among dimensions to discover lower dimensional spaces with significant features. The rank-by-feature framework [19] ranks 1D or 2D axis-parallel projections of multidimensional data sets using statistical analysis to help users detect 1D or 2D projections with desired features, such as linearly related dimensions. MacEachren et al. [16] visualize correlations between each pair of dimensions in a matrix and allow users to interactively select dimensions from the matrix to construct lower dimensional spaces. The interactive hierarchical dimension reduction approach [28] visually conveys dimension relationships using a dimension hierarchy to facilitate lower dimensional space construction. The VaR display is different from these approaches since it integrates data value visualization with dimension relationship visualization in the same display to use screen space more efficiently.

Multidimensional scaling (MDS) [4], [15] is an iterative nonlinear optimization algorithm for projecting multidimensional data down to a reduced number of dimensions. It is often used to convey relationships among data items of a multidimensional data set. For example, INSPIRE [25] uses MDS to map data items from a document data set to a 2D space. It generates a Galaxies display as a spatial representation of relationships within the document collection. In our approach, MDS is used in a different way, namely, to convey relationships among dimensions rather than data items.

The Jigsaw map [23] is a recent space filling hierarchy layout method. By placing the leaf nodes of a hierarchy into a 1D layout using a depth-first traversal and mapping the 1D layout into a rectangular 2D mesh using space-filling curves, this method creates hierarchy displays of nicely shaped regions, good continuity, and stability. When all leaf nodes are of the same size, a Jigsaw map can draw all leaf nodes without any distortion in shape, namely, they can be all equal-sized squares. This property of the Jigsaw map makes it a perfect option for us to lay out dimensions organized into a hierarchy on a 2D mesh, with each dimension drawn as a square glyph.

The similarity-based dimension arrangement proposed in [1] also addressed the problem of arranging pixel-oriented subwindows (dimensions) on a 2D mesh. It aimed to place similar dimensions close to each other on the 2D mesh. The Jigsaw map dimension layout is different in that it aims to use the dimension layout to convey the hierarchical structure among the dimensions. As a consequence, not only similar dimensions but also outlier dimensions are revealed.

Yi et al. [29] present a multidimensional visualization technique called Dust & Magnet. It represents dimensions as magnets and data items as dust particles and attracts dust particles using magnets to reveal data item values in the dimensions. The Rainfall metaphor proposed in this paper was inspired by Dust & Magnet. The difference is that the Rainfall metaphor attracts dimensions using dimensions, whereas Dust & Magnet attracts data items using dimensions.

3 Pixel MDS VaR Display

Fig. 1a illustrates the approach to generating a Pixel MDS VaR display. First, *a dimension glyph*, called a glyph in short, is generated to represent data values in each dimension, that is, values in the same column in the spreadsheet, using pixel-oriented techniques [13]. In particular, each value is represented by a pixel whose color indicates a high or low value, and pixels representing values from the same dimension are grouped together to form a glyph. In a glyph, each pixel occupies a unique position without overlap. In the original version, a spiral pixel layout was used. Rows in the spreadsheet are ordered according to their values in one dimension. (Note: Actually, any 1D order can be used.) Data values in each column are positioned into a spiral according to this order. In all glyphs, pixels representing values in the same row occupy the same position so that glyphs can be associated with each other.

Second, the correlations among the dimensions are calculated and recorded into an $N \times N$ matrix (where N is the dimensionality of the data set). In order to calculate the correlations, different approaches can be used according to different purposes. For example, if users are most interested in linearly related dimensions, Pearson's correlation coefficient can be used to capture the linear relationships among dimensions. We proposed a scalable and flexible correlation calculation algorithm [27] and applied it in the VaR display. We will briefly introduce it in Section 6 for the purpose of completeness.

Third, the $N \times N$ relationship matrix is used to generate N positions in a 2D space, one position for each dimension. The proximity among the positions reflects relationships among the dimensions; that is, closely related dimensions are spatially close to each other, and unrelated dimensions are positioned far way from each other. In particular, an MDS algorithm [4] is used to create the 2D positions upon the relationship matrix.

Finally, the dimension glyphs are placed in the 2D space in their corresponding positions to form the VaR display. Fig. 1b shows an example of the VaR display. It shows the Image-89 data set of 89 dimensions and 10,417 data items. It is a real data set containing 88 low-level visual attributes and classification information for 10,417 image segments generated by an image analysis approach [8]. In the figure, each block is a dimension glyph, and there are 89 glyphs. In each glyph, data values of the dimension are mapped to colors of pixels, and pixels are ordered in a spiral manner. The closeness of the glyph positions reveals the correlations among the dimensions calculated by the underlying algorithm. For example, several clusters of closely correlated dimensions and a few dimensions that are distinct from most other dimensions can be detected from the glyph positions in Fig. 1b.

The above approach can be summarized as dimension glyph generation and layout. Glyphs explicitly convey data values and their layout explicitly conveys dimension relationships. Moreover, dimension relationships are also revealed by the patterns of the glyphs. Similarity among glyph patterns indicates dimension relationships, whether there is a linear or nonlinear relationship or they are partially correlated (such as dimensions for which a subset

of the data items is closely related). Since humans are good at pattern recognition, the patterns of the glyphs provide straightforward and intuitive comparison of the dimensions. On the one hand, the layout approach brings related dimensions close to each other to make the pattern comparison easier. On the other hand, the patterns allow users to confirm or refute the relationships suggested by the layout using their eyes and reveal how the dimensions are related in detail.

Besides the techniques used in the original VaR display, there are other approaches to creating glyphs and laying them out, which will be introduced in the following sections. Since glyph generation and layout are independent from each other, they can be combined freely to form various VaR displays.

4 Dimension Glyph Alternative: X-Ray Scatterplots

The glyph generation approach used in the original VaR display is not the only approach for creating dimension glyphs. For example, different layouts of the pixels within a glyph reveal different patterns. As an example, organizing pixels into a calendar pattern according to the timestamps of the data items can reveal time-dependent patterns among the data items. Since these techniques have been widely studied in pixel-oriented techniques [12] and they can be integrated into the VaR display easily by replacing the original pixel-oriented dimension glyph generation approach, they will not be discussed in this paper. Instead, we present our work on customizing a density-based scatterplot glyph (called an X-Ray glyph) generation approach. This approach has been introduced into the improved version (see Fig. 1c for an example VaR display using the scatterplot approach).

In the VaR display, a scatterplot is generated for each dimension. The Y dimension of a scatterplot dimension glyph is the dimension it represents, whereas all of the glyphs have the same X dimension. We choose to use the same X dimension since it will be hard for users to associate different dimension glyphs if both X and Y dimensions change from one glyph to another. Although this causes information loss, users can always interactively change the X dimension guided by the semiautomatic selection tool (see Section 7) and their visual exploration (see Section 11).

The VaR display is targeted at large data sets. It is time-consuming to draw the projection of each data item on each of the N scatterplots. Also, the large number of projections would clutter the glyph. In order to avoid clutter and increase scalability, we store each glyph as an $M \times M$ pixel matrix, where M is an adjustable integer, and divide the 2D space within the value range of the data set into $M \times M$ equal-size bins. The number of projections falling into each bin is recorded and translated into the color of its corresponding pixel in the pixel matrix. In particular, the intensity of the pixel is proportional to the data density of the area it represents.

The first image (Fig. 2a) we generated is disappointing since it is hard to differentiate unoccupied area (areas with zero data items) from areas with a few data items. In order to solve this problem, we assign a different hue to the pixels representing unoccupied areas. In the image generated (Fig. 2b), there are no data items in the blue area. We then observed that, in contrast to glyphs generated using pixel-oriented techniques where every pixel represents a data value, there are often large contiguous unoccupied areas in a scatterplot glyph, especially when the X and Y dimensions are closely related. Recalling that some glyph layout approaches such as MDS could cause overlaps among different glyphs, we make the unoccupied areas semitransparent so that users can see hidden glyphs through the unoccupied areas of the hiding glyphs. Fig. 2c shows this final solution. Since, in the figure, the glyphs look very much like X-Ray photos, we named this VaR display the X-Ray scatterplot VaR display. To give users more flexibility, we allow them to interactively choose the color and transparency of the unoccupied areas. If users dislike the semitransparent unoccupied areas, they are able to set them to opaque.

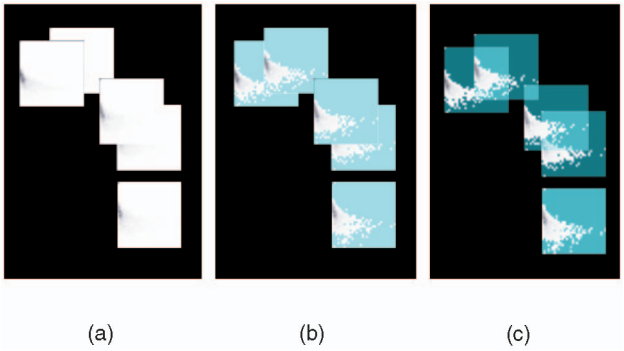


Fig. 2. X-Ray scatterplots. (a) The first solution. (b) The second solution. (c) The X-Ray scatterplot solution.

5 Dimension Layout Alternatives: Jigsaw Map Layout and Rainfall

5.1 Jigsaw Map Glyph Layout

The MDS approach is effective in conveying dimension relationships. However, using the MDS approach, the positions of two glyphs could be very close to each other if they are closely related. Glyphs might overlap in this case, which is sometimes undesired by the users. Besides, allowing the users to reduce overlaps in the MDS layout using interactions (see Section 7), we propose a Jigsaw map dimension layout based on the recently proposed Jigsaw map [23]. In this approach, dimensions are grouped into a dimension hierarchy. The Jigsaw map, which is a space-filling hierarchy visualization method, is then used to lay the dimensions on a grid. This approach not only prevents glyphs from overlapping, but also conveys the hierarchical structure among the dimensions. Fig. 3 shows VaR displays with a Jigsaw layout.

The motivation of this approach is that grouping dimensions of high-dimensional data sets into dimension hierarchies makes it easy to capture the relationships among the dimensions. In a dimension hierarchy, dimensions are organized into a hierarchy of clusters. Dimensions within a cluster have closer relationships among each other

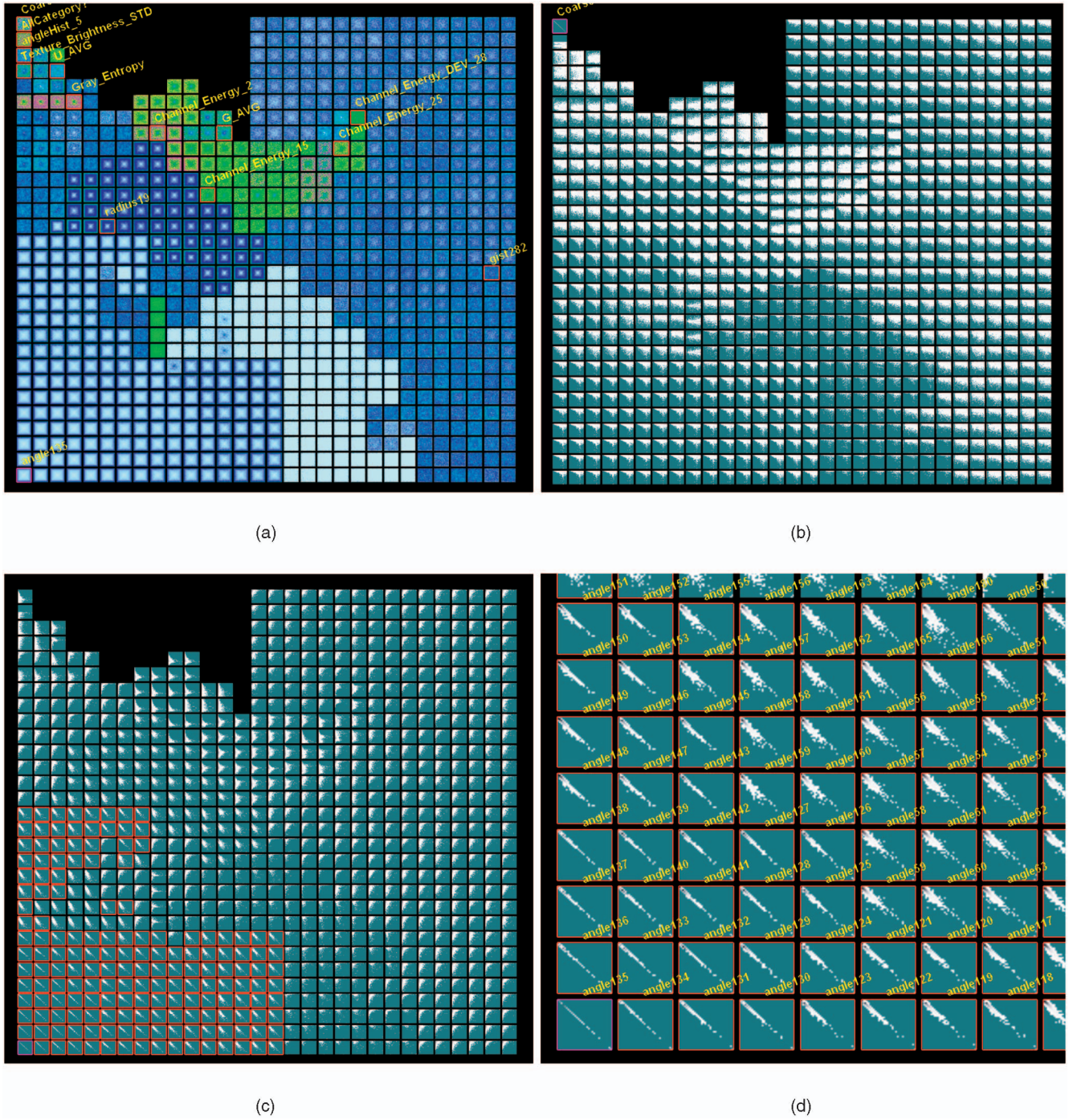


Fig. 3. The Image-838 data set (838 dimensions and 11,413 data items). (a) The Pixel Jigsaw map VaR display with separated dimensions selected and labeled. (b) The X-Ray scatterplot Jigsaw map VaR display with dimension Coarseness as the X dimension. The X dimension is in a pink frame and labeled. (c) The X-Ray scatterplot Jigsaw map VaR display with dimension angle_135 as the X dimension. The X dimension is at the left bottom corner of the map, and dimensions closely related to it are in red frames. (d) A zoomed-in display of the selected dimensions with their labels shown.

than with dimensions outside the cluster. Clusters in different levels of the hierarchy divide the dimensions into groups of different granularity. With the dimension relationship matrix, it is convenient to generate a dimension hierarchy using existing hierarchical clustering approaches. In the hierarchy, each leaf node is a dimension in the high-dimensional data set.

In order to turn the dimension hierarchy into the dimension layout, we examined existing hierarchy visualization techniques. The basic requirements are given as follows: 1) The layout should be space efficient since our target is high-dimensional data sets and 2) each dimension should be assigned a space of the same size, shape, and orientation since it is difficult for users to compare and associate glyphs with different sizes, shapes, or orientations. Since node-linked diagrams do not use space efficiently, we only considered the space-filling hierarchy visualization techniques [3], [21], [23]. Among them, only the Jigsaw map [23] and quantum treemaps [3] are capable since all other techniques assign areas of different shapes or

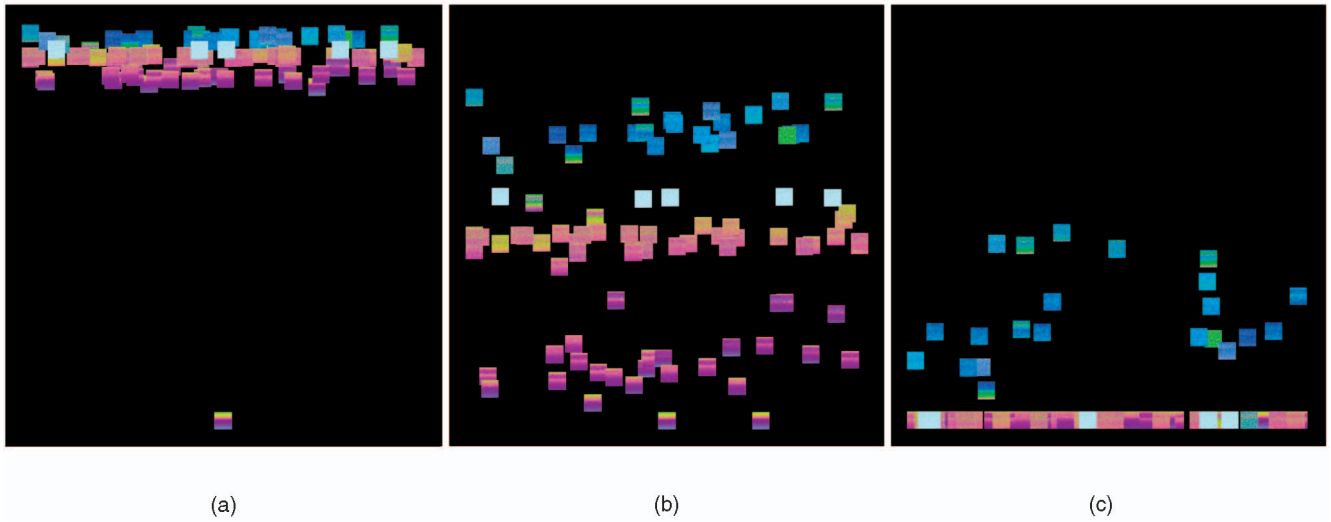


Fig. 4. The Rainfall metaphor. (a) At the beginning of the rain. Dimensions more closely related to the dimension of interest at the bottom are falling in a faster acceleration than less related dimensions. (b) The rain continues. The dimensions with different correlations to the dimension of interest are separated. It can be seen that there are roughly three levels of association between the dimension of interest and other dimensions. (c) The rain is close to its end. Dimensions significantly distinct from the dimension of interest are revealed. The data set is the Image-89 data set. The glyphs are pixel-oriented glyphs (pixels are ordered in a line-by-line (horizontal lines) manner).

orientations to leaf nodes. We chose the Jigsaw map since it generates layouts of nicely shaped regions and is stable with regard to changing tree structures and leaf nodes [23].

To generate the Jigsaw map layout, we first hierarchically cluster the N dimensions in a data set based on their pairwise distances (a pair of more closely related dimensions has a smaller distance than a pair of less related dimensions) using the minimum single linkage metrics [17]. Then, the N dimensions are ordered into a 1D sequence according to their positions in the hierarchy using a depth-first traversal of the hierarchy, and then the sequence is mapped to a 2D $L \times K$ ($L \times K >= N$) mesh by applying a space-filling curve called an H curve (please refer to [23] for more details). Fig. 3a shows an example of the Jigsaw layout. In this figure, similar dimensions are close to each other, and significant boundaries of groups of closely related dimensions, such as the group of dimensions at the left bottom part of the map, can be detected. Outlier dimensions, such as the dimensions on the left top part of the map, are also distinguishable since their textures look different from their neighbors.

5.2 Rainfall Metaphor

When exploring a high-dimensional data set, users are often interested in the relationships between a single dimension of interest with all other dimensions. Besides the X-Ray scatterplot, which reveals the relationships using glyph textures, we also provide a simple animation approach to dynamically illustrate the relationships by changing glyph positions. This approach is named the Rainfall metaphor since it imitates rain (see Fig. 4 for an example).

In the beginning of the animation, the dimension of interest is placed at the center bottom of the display (the ground), and all other dimensions (raindrops) are placed on the top of the display (the sky). The horizontal positions of the raindrops are randomly generated. After the rain starts, a raindrop falls toward the ground in an acceleration that is proportional to its correlation with the dimension of interest. Thus, a raindrop moves toward the ground faster than another raindrop if it has a closer relationship to the dimension of interest. A raindrop stops its movement after it hits the ground. There is a timer that starts from the beginning of the rain and ends when all raindrops hit the ground. Users can interactively play the animation by moving the slider representing the timer. Users can also interactively select the dimension of interest for the animation.

Figs. 4a, 4b, and 4c show some screen captures of the Rainfall layout. Using this metaphor, users can focus on the relationships between the dimension of interest and other dimensions, without being distracted by relationships among the other dimensions. In different moments of the rain, either similar dimensions or distinct dimensions to the dimension on the ground attract the users' attention.

6 Correlation Calculation

In the VaR display, a binning-based correlation calculation algorithm is used. We only briefly introduce it here since it has been presented in full detail in [27]. We claim that any relationship calculation algorithm can be used in the VaR display as long as it scales to large data sets. The layout of the glyphs reflects the type of relationship calculated by the underlying algorithm.

In our algorithm, distribution of the value differences (between the different dimensions for the same data item) is recorded into bins. In particular, the possible range of value differences between a pair of dimensions is divided into a sequence of bins. The number of data items whose value differences between these two dimensions fall into the bins is recorded. For an N-dimensional data set, $N \times (N-1)/2$ sequences of bins (one sequence for each pair of dimensions) are created. A pair of dimensions is considered to be closely related if a large number of data items fall into a small number of bins (K) in its sequence. With a given K, the correlations can be calculated in this

way: Sort the bins in the sequence according to their populations, and sum up the populations of K bins with the highest populations. The sum divided by the total population of the data items is proportional to the correlation between the dimensions. K is selected to be the number of bins that makes the global variance of correlations for all dimensions maximum. This algorithm scales to a large number of data items. Except for the first scan, which can be done with minimal cost when inserting the data set into the database, its efficiency is only related to the number of dimensions.

The above algorithm is a heuristic approach whose purpose is to maximize the visibility of the structure of the MDS and Jigsaw layout. There are many other optimization problems in the VaR display, such as selecting a dimension ordering the pixel-oriented display in the initial view to provide the maximum information to users at a first glance. A detailed discussion of such problems is presented in [27] and not repeated here.

7 Interactive Tools in the VaR Display

A rich set of interaction tools has been developed for the VaR display. Navigation tools help users reduce clutter in the display and discover information about the data set. Automatic and manual dimension selection tools allow users to perform human-driven dimension reduction by selecting subsets of dimensions for further exploration in the VaR display, as well as other multidimensional visualizations. Data item selection tools allow users to select subsets of data items for further exploration. In addition, the data item masking tool allows users to examine details of selected data items within the context of unselected data items.

Most of the interactive tools make no special assumption about the glyph positioning and generation strategies; that is, they can be applied to any realization of the VaR display. These tools are called general tools. Unless specifically noted, an interaction tool is a general tool in the following sections, where details of each navigation and selection tool are presented.

7.1 Tools for Glyph Layout

The MDS dimension layout causes overlaps among the glyphs. Overlaps emphasize close relationships among the dimensions because glyphs overlap only if their dimensions are closely related. However, overlaps can prevent a user from seeing details of an overlapped glyph. We provide the following operations to overcome this problem (see [27] for more detail):

- **Showing names**. By putting the cursor on the VaR display, the dimension names of all glyphs under the cursor position are shown in a message bar. Thus, a user can be aware of the existence of glyphs hidden by other glyphs.
- **Layer reordering**. With a mouse click, a user can force a glyph to be displayed in front of the others. In this way, he/she can view details of a glyph that is originally overlapped. Users can also randomly change the ordering of all dimension glyphs by clicking a button in the control frame. In addition, selected dimensions are automatically brought to the front of the display.
- **Manual relocation**. By holding the control key, a user can drag and drop a glyph to whatever position he/she likes. In this way, a user can separate overlapping glyphs.
- **Extent scaling**. Extent scaling allows a user to interactively decrease the sizes of all the glyphs proportionally to reduce overlaps, or to increase them to see larger glyphs.
- **Dynamic masking**. Dynamic masking allows users to hide the glyphs of unselected dimensions from the VaR display.
- **Automatic shifting**. This operation automatically reduces the overlaps among the glyphs by slightly shifting the positions of the glyphs. There are many more advanced overlap reducing algorithms that can be used, such as those listed in [22].
- **Distortion**. Users can interactively enlarge the size of some glyphs while keeping the size of all other glyphs fixed. In this way, users are allowed to examine details of patterns in the enlarged glyphs within the context provided by the other glyphs.
- **Zooming and panning**. Users can zoom in, zoom out, and pan the VaR display. For example, in order to reduce overlaps, sometimes the size of the glyphs has to be set very small when there are a large number of dimensions. Zooming into the display will enlarge the glyphs so that the user can have a clear view of the patterns in the glyphs.
- **Refining**. A refined VaR display can be generated for a selected subset of dimensions and a selected subset of data items. The selected dimensions and data items are treated as a new data set. The relationship calculation, glyph generation, and positioning are applied to the new data set.

7.2 Tools for Glyph Regeneration

In the pixel-oriented dimension glyphs, the dimension used to sort the data items affects the glyph patterns significantly. Clusters in subspaces including this dimension can be easily detected, whereas clusters in other subspaces are not. Similarly, in the X-Ray scatterplot dimension glyphs, relationships between other dimensions and the X dimension are easier to detect than relationships among other dimensions. We allow users to interactively select the sorting dimension in the pixel-oriented mode and the X dimension in the X-Ray scatterplot mode by clicking the mouse button on the glyph of the desired dimension or selecting from a combo-box.

In addition, a comparing mode can be used in the pixel-oriented glyphs in order to compare the dimensions with a dimension of interest. In this mode, except for the glyph of the base dimension, the pixels of all other glyphs will be colored according to the differences between the values of the base dimension and their dimensions. A figure of the comparing mode can be found in [27].

7.3 Dimension Selection Tools

Dimension selection tools enable users to select dimensions of interest for further exploration using other multidimensional visualization techniques. They can also be used as a

filter to reduce the number of glyphs displayed in a VaR display, since we allow users to hide glyphs of unselected dimensions using dynamic masking (see Section 7.1). The selection tools we provide to users include automatic selection tools for closely related dimensions and well-separated dimensions in addition to manual selection.

The automatic selection tool for related dimensions takes a user-assigned dimension and correlation threshold as input. Here, we assume that a pair of more closely related dimensions has a larger correlation measure than a pair of less related dimensions. Users pick the assigned dimension by clicking its glyph and adjust the threshold through a slider. The tool automatically selects all dimensions whose correlation measures to the input dimension are larger than the threshold by traversing the dimension relationship matrix. This tool enables the users to select a set of closely related dimensions.

The automatic selection tool for separated dimensions takes a user-assigned dimension and correlation threshold as input and returns a set of dimensions that describes the major features of the data set. The assigned dimension will be included in the returned set of dimensions. Between each pair of dimensions in the result set, the correlation measure is smaller than the threshold. For any dimension that is not in the result set, there is at least one dimension in the result set whose correlation measure with it is larger than the threshold. Using this tool, a user is able to select a set of dimensions to construct a lower dimensional subspace, revealing the major features of the data set without much redundancy. In Fig. 1b, separated dimensions selected automatically are labeled.

The following algorithm can be used for automatic selection of separated dimensions:

1. Get the assigned dimension and the selection threshold.
2. Set the assigned dimension as "selected" and all other dimensions as "unselected."
3. Find all unselected dimensions whose correlation measures to all existing selected dimensions are smaller than the threshold. Mark them as "candidates."
4. If there is no candidate dimension, go to Step 5. Else, set one candidate dimension as "selected" and every other candidates as "unselected." Go back to Step 3.
5. Return all dimensions marked as "selected."

It is interesting that it is not defined how to pick one dimension among the candidate dimensions in Step 4. Thus, it can be customized according to the task of interest. For example, in Section 8, this approach is customized to reduce the clutter among the labels of the selected dimensions for a good labeling result. Here, we present another customization.

When users start to explore an unknown data set, it is often desired to find dimension groups containing large numbers of closely related dimensions. Thus, a heuristic approach can be used in Step 4: setting a threshold for each candidate dimension counting the number of dimensions having correlation measures to it that are larger than the threshold and selecting the dimension with the highest count. Using this approach, dimensions with a larger number of closely related dimensions have higher priority to be selected.

Manual selection allows a user to manually select a dimension by clicking its corresponding glyph. The user can unselect the dimension by clicking the glyph again. The combination of manual and automatic selection makes the selection operation both flexible and easy to use.

7.4 Data Item Selection and Masking Tools

Rather than allowing a user to select data items directly from the glyphs in the VaR display (which is hard when glyphs are small), we allow the user to select data items from a dialog. First, the user selects a dimension name from a name list in the dialog. Then, a brief summary of the dimensions will be provided to help the user set up the selection criteria for the selected dimension. If the dimension is a categorical dimension, the distinct values in that dimension, as well as the number of data items for each value, will be provided. The user can then select the desired distinct values. If the dimension is a numeric dimension, a histogram of the dimension will be provided. The user then sets up a minimum value and a maximum value for the selection using two sliders. The user can set the selection ranges for multiple dimensions.

After the user sets the selection criteria, he/she can click a button in the dialog to trigger the selection. A problem here is how to highlight the selected data items. In most visualization systems, selected data items are highlighted using either a special color or a surrounding box around the selected items. However, in the VaR display with pixel-oriented techniques, color has been used to represent the values and it is hard to put a surrounding box in a condensed glyph, especially if the selected data items are not adjacent to each other in the glyphs.

A straightforward solution to this problem is to display only the selected data items. This is a general solution suitable for all realizations of the VaR display. However, a drawback of this approach is that the context provided by unselected data items is lost. Such a context is often useful. For example, the users might want to compare the selected data items with the unselected data items among the dimensions.

In order to overcome this drawback, we developed an approach called data item masking. This approach is only useful for VaR displays using pixel-oriented techniques. In this approach, both selected and unselected data items are drawn on the screen. Unselected data items are covered by a mask. Users can interactively change the color of the mask and adjust the transparency of the mask through a slider. When the mask is opaque, as shown in Fig. 5b, unselected data items are hidden. When the mask is fully transparent, as shown in Fig. 5a, the selected data items are not highlighted. When the mask is semitransparent, as shown in Fig. 5c, the selected data items are highlighted within the context provided by the unselected data items. Users can interactively change the transparency of the mask to adjust the strength of the context.

The implementation of this masking operation is simple. First, a mask is generated using an approach similar to the generation of a normal dimension glyph. The only difference is that the pixels are set to be transparent for selected data items and with user-assigned color and transparency for unselected data items. Our mask generation mechanism has no dependency on the order of the data

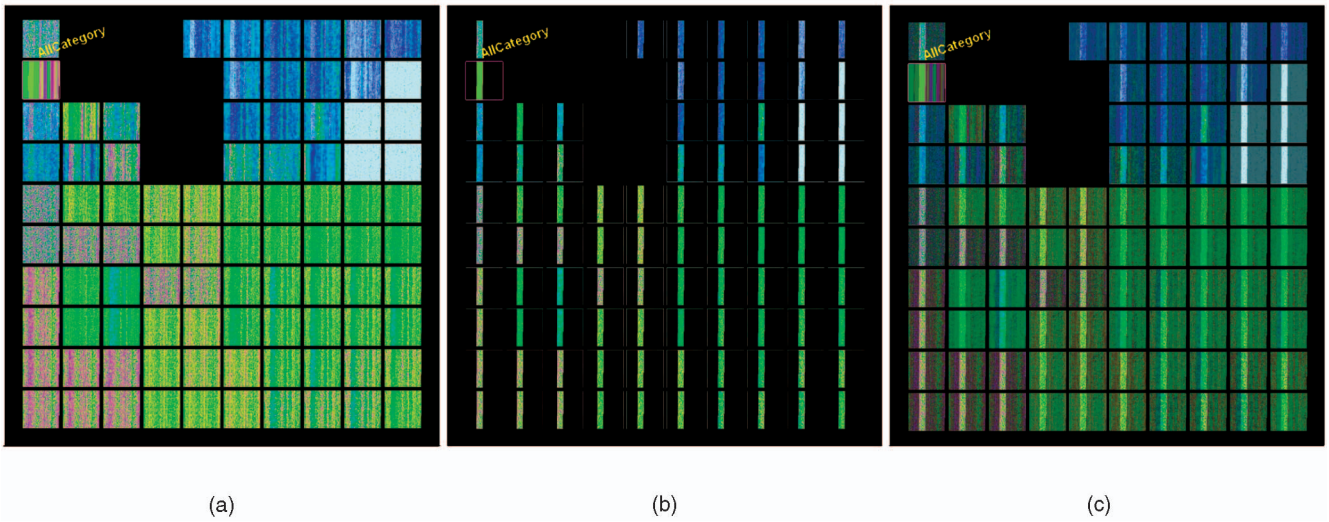


Fig. 5. Masking of unselected data items. Unselected data items are covered by a mask with adjustable color and transparency. (a) No mask or fully transparent mask. (b) Opaque mask. (b) Semitransparent mask. The data set is the Image-89 data set. The glyphs are pixel-oriented glyphs (pixels are ordered in a line-by-line (vertical lines) manner).

items; that is, it is not necessary for the selected data items to be adjacent to each other in the glyphs. Since the color and shape of the masks are the same for all the glyphs, the mask is only generated once, stored as a texture object, and pasted in front of all the glyphs. Since the texture mapping operation is efficient in OpenGL, displaying masks has a minimal effect on the rendering of a VaR display.

8 Labeling

In the original version of the VaR display, dimension names are labeled horizontally in the middle top region above the dimension glyph for all dimensions shown on the screen (see Fig. 6a). The labels clutter the screen seriously for a high-dimensional data set; thus, we did not provide the labeling option to users. Rather, when users moved the cursor over a glyph, the glyph name is shown in the message bar below the display. However, users complained that finding dimension names in this way was tiring. They argued that the VaR display without dimension labels is much less meaningful than one with names labeled. In order to solve this problem, we chose to label a subset of dimensions on the screen for the MDS layout (see Fig. 6b). The dimensions to be labeled are selected according to the two heuristic criteria: 1) They should be distinct dimensions, that is, two similar dimensions should not be labeled at the same time. Dimensions distinct from all other labeled dimensions should be labeled. 2) They should be separated from each other as much as possible to avoid clutter on the screen. In addition, we allow users to interactively change the number of dimensions labeled to get a less cluttered view or to see more labels.

Criterion 1 is exactly the criterion used for automatic selection of separated dimensions (see Section 7.3). Criterion 2 adds more constraints to the selection. Recall that there is some freedom in Step 4 of the selection algorithm, that is, any dimensions in the candidate dimension set can be selected. We modified the algorithm for labeling as follows:

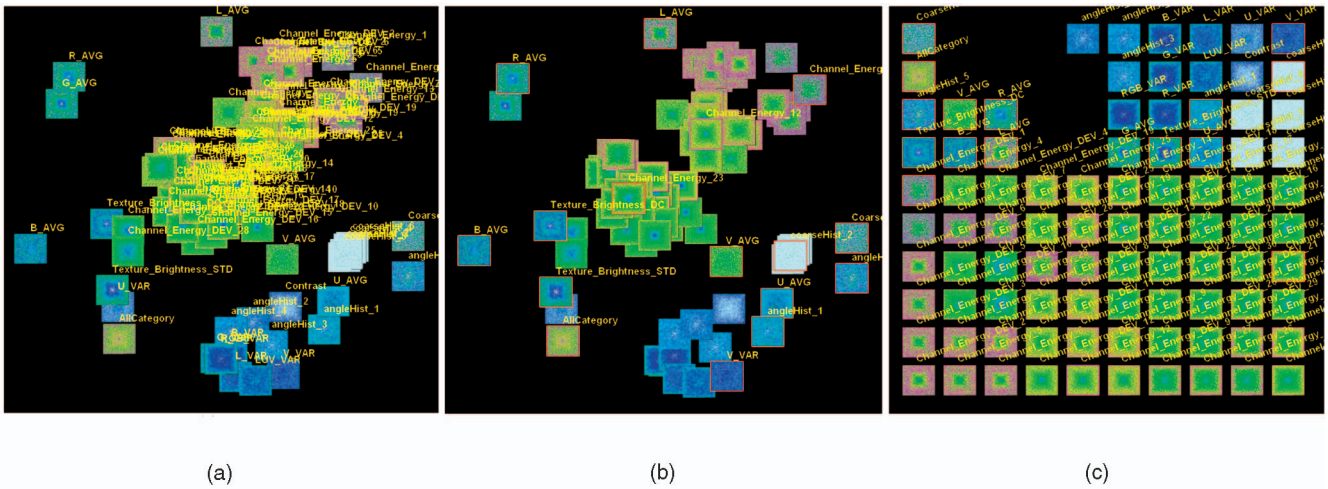


Fig. 6. Labeling solutions. (a) All dimensions are labeled with names. (b) Dimensions selected by the labeling algorithm are labeled. Clutter is reduced. (c) Angled text is used to label all dimensions in the Jigsaw map layout. The data set is the Image-89 data set.

1. Assign a dimension and a selection threshold.
2. Set the assigned dimension as "selected" and all other dimensions as "unselected."
3. Find all unselected dimensions whose correlations with all existing selected dimensions are smaller than the threshold. Mark them as "candidates."
4. If there is no candidate dimension, go to Step 5. Else, set the candidate dimension that is the farthest away on the screen from its closest existing selected dimension as "selected" and other candidates as "unselected." Go back to Step 3.
5. Return all dimensions marked as "selected" and label them.

When calculating the screen distance between two dimensions in Step 4, we must consider the fact that horizontal labels are used. Their lengths are much larger than their widths. Assume that labels have five characters on the average and the characters have equal height and width. The screen distance between two dimensions d1 and d2 is

$$\mathrm{D}(\mathrm{d1}, \mathrm{d2}) = \mathrm{fabs}((\mathrm{d1.x} - \mathrm{d1.x})) + 5 * \mathrm{fabs}((\mathrm{d1.y} - \mathrm{d2.y})),$$

where x and y are the screen coordinates of the dimensions. The equation means that we prefer dimensions separated in the vertical direction rather than in the horizontal direction. Fig. 6b shows the same display as Fig. 6a with selected dimensions labeled using the above algorithm.

The same labeling approach can be applied to the Jigsaw map layout. In addition, since the glyphs are placed in a regular mesh in the Jigsaw map, applying an angle on all the labels greatly reduces the clutter on the screen even when all labels are shown. Fig. 6c shows the Image-89 data sets in the Jigsaw map layout with all dimension names displayed at a 20° angle. Almost all of the dimension names can be distinguished from this display.

In our prototype, we bind labeling with selections, that is, users have the option to show labels of selected dimensions only. When a user chooses this option and uses the automatic selection tool for separated dimensions, it is exactly the above clutter-reducing labeling approach. When a user uses the selection tool for related dimensions, the dimensions closely related to the user-assigned dimensions are labeled (see Fig. 3d for an example).

9 Implementation and Scalability Issue

When there are several hundred dimensions, the data sets can easily contain millions of data values even if they only contain thousands of data items. Data sets often have a higher number of data items. Such large data sets not only cause large response time during interactions and problems in storing the data structures in a visualization system, but also cause clutter on the display. Scalability is a critical issue for visualization systems aimed at high-dimensional data sets.

We have implemented a fully working prototype of the VaR display. The biggest data set that has been successfully loaded into the VaR display so far is an image classification data set containing 838 dimensions and 11,413 data items, which means over 9 million data values (see Fig. 3 for its VaR display). Most interactions can be processed within a few seconds on a typical PC for this data set. This data set is the biggest data set that we currently have. In the future, we will test larger data sets on the prototype.

The critical techniques we used in the prototype for increasing scalability are texture mapping, binning, and sampling techniques. Using the texture mapping techniques provided by OpenGL, our prototype stores all dimension glyphs (including the mask in the masking operation) as texture objects and pastes them on the screen as needed. As long as the glyph textures do not change, the data set does not need to be rescanned, which is time consuming for large data sets. By keeping the texture objects small (such as hundreds of pixels), which is reasonable since each dimension glyph will not be too big on the screen in order to reduce clutter, the system can draw hundreds of dimension glyph textures on the screen in almost real time. This approach greatly reduces the response time for most interactions because, except for reordering for pixel-oriented glyphs and resetting the X dimension for X-Ray scatterplot glyphs, almost all other interactions do not change the glyph textures. Rather, they refresh, resize, reposition, or reorder the glyphs.

According to our experience, drawing fonts in OpenGL is a time-consuming task. Our prototype stores all dimension name labels as texture objects. These texture labels are created one time and can be quickly pasted on the screen until users change the contents or colors of the labels. The texture labels can be scaled and rotated easily on the screen.

Binning, that is, using buckets to stored statistical information about groups of values rather than recording them individually, is an approach widely used in data-mining techniques for large data sets. We use binning techniques to increase the speed of the correlation calculation algorithm (see Section 6) and the X-Ray scatterplot glyph generation (see Section 4).

The prototype stores data sets in an Oracle database server. It dynamically requests data from the server when needed, making use of the sorting and query functions provided by the database server. When generating a VaR display for a data set containing a large number of data items, we use a random sampling approach to reduce the response time for fetching data items from the server as well as the number of values to be processed. In particular, the system keeps a default maximum number. When the number of data items contained in a data set exceeds it, a uniform random sampling is performed on the data set to only fetch the maximum number of data items. Users are allowed to interactively adjust the maximum number in order to trade between the response time and visualization accuracy.

Random sampling is easy to implement. However, it has the big drawback that a large sampling rate is needed in order to reduce small group loss in the samples [6]. In order to overcome this problem, many solutions have been proposed such as biased sampling [14] or dynamic sample selection [2]. It has been shown in the literature that these approaches successfully reduce small group loss. We will explore these approaches in the future.

10 Discussion

The VaR display can serve as an overview tool for a high-dimensional data set. Starting from the VaR display, other visualization techniques can be used for more detailed visual analysis. For example, the VaR display is coordinated

with parallel coordinates, star glyphs, and scatterplot matrix views in our prototype. Although these techniques could not handle hundreds of dimensions, they work well in examining data items and dimensions selected by the VaR display. Recently, we completed an interesting project in coordinating the VaR display with an image exploration interface. The VaR display was used to show the high-dimensional image content annotations. Users were allowed to select images by contents from the VaR display. The images were then examined in detail in an image exploration interface. This work is described in [26].

The MDS and Jigsaw map glyph layout approaches have their advantages and disadvantages. From its nature, MDS is better in capturing high-dimensional relationships than the hierarchical approach. However, the nonoverlap feature of the Jigsaw map layout makes it a popular approach for users of the VaR display thus far.

Although the pixel-oriented glyphs are mentioned less than the X-Ray scatterplot glyphs in this paper, this is only because the usage of the pixel-oriented techniques has been widely studied, and their effectiveness has been shown in many papers. Compared to scatterplots, the pixel-oriented glyphs are more effective in pixel usage since they make use of each pixel. However, it is easier to compare the relationship between a dimension of interest and all other dimensions using the scatterplot glyphs. Users find it difficult to compare the patterns of pixel-oriented glyphs if they are far from each other.

Compared to scatterplot matrices, the X-Ray scatterplot VaR display has its advantages and disadvantages. For data sets with a small number of dimensions, scatterplot matrices might be preferred since all possible axis-parallel 2D projections are provided in them. However, for data sets with tens, hundreds, or thousands of dimensions, the X-Ray scatterplot VaR display might be preferred since it causes less clutter. Its disadvantage that only part of the possible 2D projections are displayed is leveraged by two facts: First, dimension relationships conveyed by the VaR display give strong hints on the shapes of the undisplayed 2D projections. Second, users can interactively access 2D projections of interest through interactions.

Compared to approaches that rank 1D or 2D projections according to their features and allow users to examine details of a projection by selecting it from diagrams or lists conveying the ranking (such as the rank-by-feature framework [19]), the VaR display also has its advantages and disadvantages. Obviously, for tasks such as finding the most linearly correlated dimensions, the ranking approaches are better choices. However, the VaR display is better in helping users grasp the global relationships among the dimensions.

11 Case Study

We have explored several real data sets using the VaR display, including the Image-838 data set [8] with 838 dimensions and 11,413 data items and the Image-89 data set [8] with 89 dimensions and 10,471 data items. They all contain low-level visual attributes for image classification. Image analysts are interested in finding outlier dimensions that are uncorrelated to most other dimensions and dimensions representing a group of correlated dimensions (a dimension cluster) in order to reduce the number of low-level visual attributes used in the image classification process.

For both data sets, we selected a Pixel MDS VaR display with all dimensions displayed as the initial view, since the pixel-oriented glyphs have a higher pixel usage efficiency and the MDS display conveys dimension relationships more accurately than the Jigsaw map layout. Fig. 7a and Fig. 1b show the Pixel MDS VaR displays of the Image-838 data set and the Image-89 data set, respectively. From the figures, we found that there are dimension outliers and clusters in both data sets. We then applied automatic selections for separated dimensions. Both outlier dimensions and dimensions representing dimension clusters were selected.

Then, we switched to the Jigsaw map layout. Fig. 3a shows the Pixel Jigsaw map VaR display of the Image-838 data set. There are several distinguishable regions that can be seen in the map where adjacent glyphs in the regions have similar patterns. For example, there is a distinguishable region composed of bright blue glyphs at the left bottom corner of the map. If only one dimension is selected in such a region, it means that the neighbors of the selected dimension are closely related to it since selection for separated dimensions was used. Thus, they are a dimension cluster, and the selected dimension can represent the cluster. The selected and labeled dimension angle_135 at the left bottom corner is such a representative dimension. Meanwhile, selected dimensions crowded together, such as the selected dimensions on the left top part of the map, are potential outliers since they are distinct from their closest neighbors. The selected and labeled dimension Coarseness on the left top corner is one such suspicious outlier.

In order to examine if dimension Coarseness is an outlier, an X-Ray scatterplot VaR display was created using it as the X dimension (see Fig. 3b). From the scatterplots in Fig. 3b, it can be seen that no other dimensions show strong correlations with dimension Coarseness. Thus, it is confirmed that dimension Coarseness is an outlier dimension.

Fig. 3c examines if dimension angle_135 is a representative dimension. The X dimension of the scatterplots is dimension angle_135, and dimensions closely correlated to dimension angle_135 are selected and highlighted. It can be seen that a large number of dimensions are selected and they all contain a clear diagonal pattern, which indicates a strong linear correlation. Fig. 3d shows a zoomed-in display of the selected dimensions in which their labels are shown.

A similar exploration approach was conducted for the Image-89 data set. An interesting pattern in this data set was found when we were examining dimension Channel_Energy_5 using the X-Ray scatteplot Jigsaw map VaR display (Fig. 7b): There was a glyph with a curved band (the glyph with a yellow frame, the frame was manually added into the figure for highlighting). It seemed that this dimension was nonlinearly related to the target dimension. It raised our interest and became our next target.

We clicked this dimension to set it as the X dimension in the X-Ray scatterplots and got Fig. 7c. It is labeled in Fig. 7c as Texture_Brightness_DC. Fig. 7c shows that dimension Texture_Brightness_DC is nonlinearly related to most dimensions in this data set. The curved bands are fairly thin in some dimensions, which means strong nonlinear relationships.

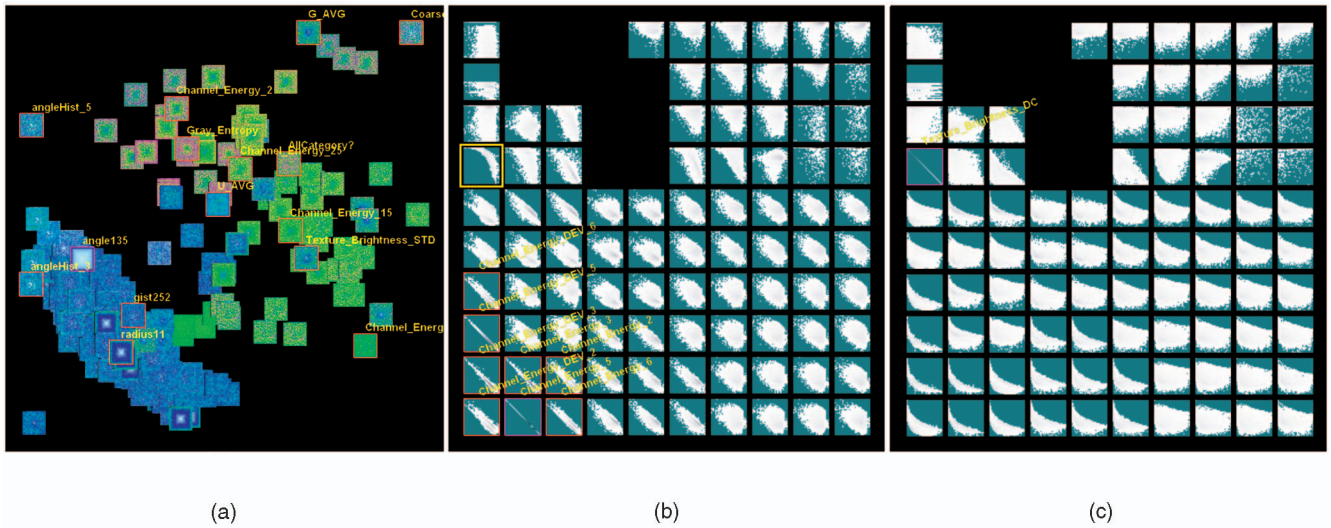


Fig. 7. (a) The Pixel MDS VaR display of the Image-838 data set with separated dimensions selected and labeled. (b) The X-Ray scatterplot Jigsaw map VaR display of the Image-89 data set. The dimension in a yellow frame is nonlinearly related to the X dimension. (c) The X-Ray scatterplot Jigsaw map VaR display with another X dimension (the dimension highlighted by the yellow frame in (b)).

12 User Study

A user study has been conducted to evaluate the VaR display by comparing it to the Rank-by-Feature feature of HCE [19]. To form a comparable study, we considered the X-Ray scatterplot glyph style of VaR and the scatterplot prism from the HCE system, namely, its 2D projection ranking, selection, and visualization feature. In HCE, 2D projections are ranked by features such as strength of linear relationship or least square error for curvilinear regression. The ranking is visualized in both a matrix and a list. A window beside the ranking windows shows the scatterplot of the 2D projection selected by the user. Our assumption was that the VaR display would better help users grasp global relationships among the dimensions in a high-dimensional data set. The reason is that VaR provides a detailed view of all dimensions at the same time while users of HCE need to take efforts to associate multiple dimensions since they can only examine a few detailed views at the same time.

Eight subjects participated in the user study. The subjects vary in educational backgrounds: One was a psychology graduate student, two were computer science undergraduate students, three were graduate students in the field of visualization, and two were researchers/postdoctorates in visualization. The subjects completed the user study one by one on the same computer with the same instructor. Each subject tested both systems. The order of using VaR and HCE was alternated for the subjects.

The study began with a 10-minute training session using both VaR and HCE and a further 10 minutes to allow subjects to explore the tools and ask the instructor questions. A set of tasks was then completed by the subjects using both tools. A posttest survey to find user preferences and a discussion were conducted immediately following the completion of the tasks. We used the Image-89 data set of 89 dimensions and 10,471 data items. As shown in the case study (Section 11), there are some strong linearly related dimensions and some strong nonlinearly related dimensions in the Image-89 data set.

The first task was to describe relationships between a given dimension and each of the other dimensions using the scatterplot displays by approximating the numbers of different scatterplot shapes involved with the given dimension. Samples of typical shapes, such as diagonal thin straight bands for linear relations curved bands for nonlinear relations and evenly distributed scatterplot indicating unrelated dimensions, were provided to users. The second task required users to describe relationships among five randomly assigned dimensions using their scatterplot shapes.

The majority of users performed the first task quicker and evaluated the task to be easier using the VaR display. The average time was 3.2 minutes and the standard deviation was 0.5 minutes for VaR and the average time was 4.7 minutes and the standard deviation was 3.2 minutes for HCE. On a scale of 0 (hard) to 5 (easy), the mean scores of 3.5 and 2.1 were given to VaR and HCE, respectively. A similar trend was identified in the second task: The average time was 3.5 minutes and the standard deviation was 0.4 minutes for VaR, and the average time was 8.5 minutes and the standard deviation was 2.9 minutes for HCE. The scores are 3.6 for VaR and 1.0 for HCE. Results from these tasks highlighted the advantage of the VaR display in providing a global view of the dimension relationships.

Qualitative results and qualitative feedback from the posttest survey were also encouraging. Users typically preferred using VaR over HCE for the given tasks. The reasons given by each user were generally similar and can be summarized by the ability to examine details of multiple relations on a single display. One user in the study preferred HCE over VaR due to the more detailed and visible scatterplots in the HCE system. Users were also asked if they agreed with the statement "this tool is useful for exploring high-dimensional data." On a scale of 0 (disagree) to 5 (agree), users responded with a mean score of 4.3 and 3.5 for VaR and HCE, respectively.

A number of comments and suggestions were made by the users regarding both systems. Positive feedback from VaR included an intuitive interface, the instantaneous global view and ability to quickly select the X dimension

of all scatterplots. Improvements suggested by the users involved ranking the dimensions by features and using color and best fit lines to enhance the scatterplot displays, which were considered too dense. In addition, users suggested ordering the dimension glyphs according to the shapes of the scatterplot using automatic image analysis techniques. For the HCE system, users preferred the ranking features and the scatterplot display with rich features and interactions. Users suggested that the global view provided by the prism in HCE lacked details compared to the VaR display. Future work may benefit by combining the best features of these two systems.

13 Conclusion

In this paper, the VaR display, which allows users to interactively explore large data sets with hundreds of dimensions, was presented. The essential idea of the VaR display is to represent each dimension in a high-dimensional data set using an information-rich glyph and arranging the glyphs to reveal the relationships among the dimensions. By integrating existing techniques such as MDS, Jigsaw map, pixel-oriented techniques, and scatterplots and allowing users to interactively explore large data sets according to their interests, the VaR display provides a rich metaphor for interactive exploration of high-dimensional data sets. The case studies and user study conducted proved that the VaR display is an effective approach with high scalability.

Although work presented in this paper has greatly extended the functionality of the original VaR display [27], we believe that the VaR display still has much potential for further development. Time-dependent dimension glyph generation or layout, the ability to convey spatial information, and the ability to visualize dynamically changing data streams are future directions we want to explore in the VaR display. In addition, detecting features by analyzing and comparing textures of dimension glyphs using automatic image analysis techniques is also an appealing subject of future work. Another important subject of future work is to conduct user studies to evaluate different options provided by the VaR display.

Acknowledgments

The authors thank Dr. Daniel A. Keim, who gave many valuable suggestions for this work, and the users who participated in the user study. This work was performed with partial support from US National Science Foundation Grant IIS-0119276 and the National Visualization and Analytics Center (NVAC), a US Department of Homeland Security Program, under the auspices of the Southeastern Regional Visualization and Analytics Center. NVAC is operated by the Pacific Northwest National Laboratory (PNNL), a US Department of Energy Office of Science laboratory.

References

[1] M. Ankerst, S. Berchtold, and D.A. Keim, "Similarity Clustering of Dimensions for an Enhanced Visualization of Multidimensional Data," *Proc. IEEE Symp. Information Visualization,* pp. 52-60, 1998.

[2] B. Babcock, S. Chaudhuri, and G. Das, "Dynamic Sample Selection for Approximate Query Processing," *Proc. ACM Special Interest Group on Management of Data (SIGMOD) Int'l Conf. Management of Data,* pp. 539-550, 2003.

[3] B. Bederson, B. Shneiderman, and M. Wattenberg, "Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies," *ACM Trans. Graphics,* vol. 21, no. 4, pp. 833-854, 2002.

[4] C.L. Bentley and M.O. Ward, "Animating Multidimensional Scaling to Visualize n-Dimensional Data Sets," *Proc. IEEE Symp. Information Visualization,* pp. 72-73, 1996.

[5] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is 'Nearest Neighbor' Meaningful?" *Lecture Notes in Computer Science,* vol. 1540, pp. 217-235, 1999.

[6] S. Chaudhuri, R. Motwani, and V. Narasayya, "Random Sampling for Histogram Construction: How Much Is Enough?" *Proc. ACM Special Interest Group on Management of Data (SIGMOD) Int'l Conf. Management of Data,* pp. 436-447, 1998.

[7] W.S. Cleveland and M.E. McGill, *Dynamic Graphics for Statistics.* Wadsworth, 1988.

[8] J. Fan, Y. Gao, and H. Luo, "Multi-Level Annotation of Natural Scenes Using Dominant Image Components and Semantic Image Concepts," *Proc. Ann. ACM Int'l Conf. Multimedia,* pp. 540-547, 2004.

[9] J.-D. Fekete and C. Plaisant, "Interactive Information Visualization of a Million Items," *Proc. IEEE Symp. Information Visualization,* pp. 117-124, 2002.

[10] Y. Fua, M.O. Ward, and E.A. Rundensteiner, "Hierarchical Parallel Coordinates for Exploration of Large Data Sets," *Proc. IEEE Visualization,* pp. 43-50, Oct. 1999.

[11] B. Hibbard and D. Santek, "The vis-5d System for Easy Interactive Visualization," *Proc. IEEE Visualization,* pp. 28-35, 1990.

[12] D.A. Keim, "Designing Pixel-Oriented Visualization Techniques: Theory and Applications," *IEEE Trans. Visualization and Computer Graphics,* vol. 6, no. 1, pp. 1-20, Jan.-Mar. 2000.

[13] D.A. Keim, H.-P. Kriegel, and M. Ankerst, "Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data," *Proc. Sixth IEEE Visualization (VIS '95),* pp. 279-286, 1995.

[14] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold, "Efficient Biased Sampling for Approximate Clustering and Outlier Detection in Large Data Sets," *IEEE Trans. Knowledge and Data Eng.,* vol. 15, no. 5, pp. 1170-1187, Sept./Oct. 2003.

[15] J.B. Kruskal and M. Wish, *Multidimensional Scaling.* Sage, 1978.

[16] A. MacEachren, X. Dai, F. Hardisty, D. Guo, and G. Lengerich, "Exploring High-D Spaces with Multiform Matrices and Small Multiples," *Proc. IEEE Symp. Information Visualization,* pp. 31-38, 2003.

[17] F. Murtagh, "A Survey of Recent Advances in Hierarchical Clustering Algorithms," *Computer J.,* vol. 26, no. 4, pp. 354-359, 1983.

[18] NetMBA, http://www.netmba.com/statistics/plot/scatter/, 2006.

[19] J. Seo and B. Shneiderman, "A Rank-by-Feature Framework for Unsupervised Multidimensional Data Exploration Using Low Dimensional Projections," *Proc. IEEE Symp. Information Visualization,* pp. 65-72, 2004.

[20] J. Seo and B. Shneiderman, "A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data," *Information Visualization,* vol. 4, no. 2, pp. 96-113, 2005.

[21] B. Shneiderman, "Tree Visualization with Tree-Maps: A 2D Space-Filling Approach," *ACM Trans. Graphics,* vol. 11, no. 1, pp. 92-99, Jan. 1992.

[22] M.O. Ward, "A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization," *Information Visualization,* vol. 1, no. 3-4, pp. 194-210, 2002.

[23] M. Wattenberg, "A Note on Space-Filling Visualizations and Space-Filling Curves," *Proc. IEEE Symp. Information Visualization,* pp. 181-186, 2005.

[24] E.J. Wegman and Q. Luo, "High Dimensional Clustering Using Parallel Coordinates and the Grand Tour," *Computing Science and Statistics,* vol. 28, pp. 361-368, 1997.

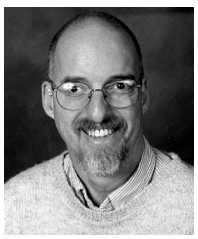
[25] J.A. Wise, J.J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow, "Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents," *Proc. IEEE Symp. Information Visualization,* pp. 51-58, 1995.

[26] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky, and M. Ward, "Semantic Image Browser: Bridging Information Visualization with Automated Intelligent Image Analysis," *Proc. IEEE Symp. Visual Analytics Science and Technology,* pp. 191-198, 2006.

[27] J. Yang, A. Patro, S. Huang, N. Mehta, M. Ward, and E. Rundensteiner, "Value and Relation Display for Interactive Exploration of High Dimensional Data Sets," *Proc. IEEE Symp. Information Visualization*, pp. 73-80, 2004.
[28] J. Yang et al., "Visual Hierarchical Dimension Reduction for Exploration of High Dimensional Data Sets," *Proc. Eurographics/IEEE Technical Committee on Visualization and Graphics Symp. Visualization*, pp. 19-28, 2003.
[29] J. Yi et al., "Dust & Magnet: Multivariate Information Visualization Using a Magnet Metaphor," *Information Visualization*, vol. 4, pp. 239-256, 2005.

Jing Yang received the PhD degree in computer science from Worcester Polytechnic Institute in 2005. She is an assistant professor in the Computer Science Department at the University of North Carolina Charlotte. She is a coprincipal investigator for the DHS Southeastern Regional Visualization and Analytics Center. She has been conducting research in the fields of information visualization and visual analytics, focused on large-scale multivariate visualization for the past seven years. She has proposed several distinct approaches to visually exploring multivariate data sets containing large numbers of data records and large number of dimensions. Her recent interests also include interactive image and multimedia retrieval and browsing. Her work has been published extensively in refereed journals and conferences. She has been a reviewer for the IEEE InfoVis Symposium, the IEEE Visualization Conference, the IEEE VAST Symposium, the *IEEE Transactions on Visualization and Computer Graphics*, and the *Information Visualization Journal*, and she was on the Program Committee for the IEEE Symposium on Information Visualization in 2005 and 2006.

Daniel Hubball received the BSc degree in computer science from the University of Wales, Swansea, in 2004. He is an MPhil candidate in computer science. His main research interests include visualization, image processing, and computer graphics.



Matthew O. Ward received the PhD degree in computer science from the University of Connecticut in 1981. He is a full professor in the Computer Science Department at Worcester Polytechnic Institute (WPI). He was employed as a member of the technical staff in the Robotics and Computer Systems Research Laboratory at AT&T Bell Laboratories between 1980 and 1984 and as a research scientist at Skantek Corporation until 1986, when he joined the faculty at WPI. His research interests include data and information visualization, visual languages, and exploratory data analysis. He has authored or coauthored more than 70 papers and book chapters in these areas and is actively involved in the development of a text book on data visualization. He has served in many roles on the organizing committee for the IEEE Symposium on Information Visualization (posters cochair, 2003, program cochair, 2004; papers cochair, 2005; program chair, 2006) and on the program committees for many US and other international visualization conferences. He was a contributor to the National Visualization and Analytics Center's research agenda for visual analytics and the US National Science Foundation/National Institutes of Health Visualization Research Challenges report in 2005. He is a member of the IEEE Computer Society.

Elke A. Rundensteiner received the BS degree (Vordiplom) from the J.W. Goethe University, Frankfurt, West Germany, the MS degree from Florida State University, and the PhD degree from the University of California, Irvine, all in computer science. She is a full professor in the Department of Computer Science at the Worcester Polytechnic Institute, after having been a faculty member at the University of Michigan, Ann Arbor. She is a well-known expert in databases and information systems, having spent 20 years of her career focusing on the development of scalable data management technology in support of advanced applications including manufacturing and automation, human genome, and digital libraries. Her current research interests include data integration and migration, XML and Web data management, data warehousing for distributed systems, continuous query processing, and large-scale information visualization. She has more than 200 publications in these and related areas. Her research has been funded by government agencies including the US National Science Foundation (NSF) and industry like IBM, Verizon Labs, GTE, NEC, and others. She is the recipient of numerous honors and awards, including the NSF Young Investigator grant. She is on the program committees of prestigious conferences in the database field and editor of several journals, including being an associate editor of the *IEEE Transactions on Knowledge and Data Engineering*. She is a member of the IEEE Computer Society.

William Ribarsky received the PhD degree in physics from the University of Cincinnati. He is the Bank of America endowed chair in information technology at the Univeristy of North Carolina Charlotte and the founding director of the Charlotte Visualization Center. He is the principal investigator for the new DHS Southeastern Regional Visualization and Analytics Center. His research interests include visual analytics, 3D multimodal interaction, bioinformatics visualization, virtual environments, visual reasoning, and interactive visualization of large-scale information spaces. He is the former chair and the current director of the IEEE Visualization and Graphics Technical Committee. He is also a member of the steering committees for the IEEE Visualization Conference and the IEEE Virtual Reality Conference, the leading international conferences in their fields. He was an associate editor of the *IEEE Transactions on Visualization and Computer Graphics* and is currently an editorial board member for *IEEE Computer Graphics and Applications*. He cofounded the Eurographics/IEEE Visualization Conference series (now called EG/IEEE EuroVis) and led the effort to establish the current Virtual Reality Conference series. He has published more than 100 scholarly papers, book chapters, and books. He has received competitive research grants and contracts from the US National Science Foundation, the US Army Research Laboratory, the US Army Research Office, the US Department of Homeland Security, the US Office of Naval Research, the US Environmental Protection Agency, the US Air Force Office of Scientific Research, the US Defense Advanced Research Projects Agency, the US National Aeronautics and Space Agency, the US National Imagery and Mapping Agency, and several companies. He is a member of the IEEE Computer Society.