HIERARCHICAL EXPLORATION OF LARGE MULTIVARIATE DATA SETS

Jing Yang, Matthew O. Ward and Elke A. Rundensteiner
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609
{yangjing,matt,rundenst}@cs.wpi.edu

Abstract Multivariate data visualization techniques are often limited in terms of the number of data records that can be simultaneously displayed in a manner that allows ready interpretation. Due to the size of the screen and number of pixels available, visualizing more than a few thousand data points generally leads to clutter and occlusion. This in turn restricts our ability to detect, classify, and measure phenomena of interest, such as clusters, anomalies, trends, and patterns. In this paper we describe our experiences in the development of multi-resolution visualization techniques for large multivariate data sets. By hierarchically clustering the data and displaying aggregation information for each cluster, we can examine the data set at multiple levels of abstraction. In addition, by providing powerful navigation and filtering operations, we can create an environment suitable for interactive exploration without overloading the user with dense information displays. In this paper, we illustrate that our hierarchical displays are general by successfully applying them to four popular yet non-scalable visualizations, namely parallel coordinates, glyphs, scatterplot matrices and dimensional stacking.

Introduction

One important approach for supporting the human in analyzing and exploring data sets is to appropriately visualize the data and allow the human to apply their innate perception and pattern recognition abilities to make sense out of the data. Multivariate data visualization focuses on the display of multidimensional data sets. Many multivariate visualization techniques and systems have emerged during the last three decades, including:

- glyph techniques (Andrews, 1972; Chernoff, 1973; Ribarsky et al., 1994),

- parallel coordinates (Inselberg and Dimsdale, 1990; Wegman, 1990),
- scatterplot matrices (Cleveland and McGill, 1988),
- dimensional stacking (LeBlanc et al., 1990), and
- pixel-based techniques (Keim et al., 1995).

Each method has its strengths and weaknesses in terms of the exploration tasks and data set characteristics for which it is most useful.

However, most existing techniques do not scale well as data sets get larger and larger. The reason is that the available screen space is limited. Hence when the data set reaches a certain size, we are no longer able to place all data on the screen at the same time without extensive cluttering and occlusion. For example, if every pixel on a 1024*1024 screen could present one data item, then the maximum number we could put on the screen at the same time without clutter is approximately one million. Unfortunately, large data sets nowadays easily exceed this size, especially when the number of dimensions is large. Worse yet, most of the existing multivariate visualization techniques require much more than one pixel per data item.

In this paper we present a number of techniques for visual exploration of large multivariate data sets we have recently developed to overcome the serious clutter problem. By hierarchically clustering the data and displaying cluster summarizations to users, along with tools for navigating the hierarchy and filtering the clutter, we can visualize data sets with millions, or even billions, of data points. To validate our ideas, we demonstrate how we have extended several traditional visualization techniques, including parallel coordinates, glyphs, scatterplot matrices and dimensional stacking, to handle hierarchies of clusters. Lastly, we also describe several powerful tools for aiding the exploration process. All techniques described in this paper have been implemented and integrated into a working visualization system, called XmdvTool (Ward, 1994; Martin and Ward, 1995; Fua et al., 1999a; Fua et al., 1999b; Fua et al., 2000).

1. Hierarchical Cluster Visualization

In this section, we outline the stages of creating the hierarchical clusters and visualizing the resulting tree structure. Details of each component of the process can be found in (Yang et al., 2001).

1.1 Hierarchical Cluster Tree

To overcome the problem of clutter on the display, the key strategy is to put fewer items on the screen. Thus we need to compress the data sets while preserving their significant features. Moreover, we prefer multiresolutional displays so that users can interactively select their preferred level of detail. Given the above considerations, we have explored the concept of constructing a hierarchical cluster tree for a data set and designing hierarchical displays capable of conveying the contents of this tree.

Each node T_i of a hierarchical cluster tree represents a cluster. A non-leaf cluster is composed of all its child clusters, while a leaf cluster contains only a single data item from the data set. A hierarchical cluster tree structures and presents a large data set at different levels of abstraction. On extreme points, the collection of all leaf clusters presents exactly every data item of the data set, while the root is a cluster representing the whole data set.

A hierarchical cluster tree is typically formed by grouping objects based on some measure of proximity between pairs of objects (Jain and Dubes, 1988). A number of clustering algorithms have been proposed for building hierarchical cluster trees of large data sets (Andreae et al., 1990; Guha et al., 1998; Zhang et al., 1996). We note that our visualization techniques are independent of the particular choice of the clustering algorithms and in fact could equally be applied to hierarchical data sets constructed based on some explicit hierarchical clustering. Hence, clustering algorithms are not further discussed here.

1.2 Visual Depiction of a Cluster

There are many characteristics of a cluster that can be visualized in order to convey useful information regarding the cluster contents. Typical aggregation information such as the cluster population, center, extents (range of values for each dimension), and distribution can be effective in helping the viewer decide whether she is interested in exploring the cluster in more detail. In our initial implementation of cluster visualization, we have selected to emphasize two cluster characteristics, namely the cluster center and its extents. The center is simply a point in data space, computed as the average of all cluster members. It is visualized in the same manner as a data point in traditional single resolution displays. The extents are computed as the extreme values for each dimension found in the member points. Extents are depicted as a hyperbox surrounding the center point, colored according to the location of the cluster relative to all other clusters at the given level of

detail (see Section 1.3 for a more detailed explanation on the coloring). Rather than using uniformly shaded boxes to depict extents, we vary the opacity such that it decreases linearly from the cluster center to the cluster edges. The opacity is set to 0 at the edges. The opacity at the center is set proportional to the cluster population, so that large clusters are more readily discerned than small ones.

1.3 Proximity-Based Coloring

Data elements (or clusters, in our case) can have many types of relationships between them in screen-space (proximity after projection), data-space (proximity in values), and structure-space (proximity based on the results of structuring). It is often useful to highlight these relationships to encourage interactive exploration. To this end, we have been examining techniques to convey the structural relationships found in the cluster hierarchy. In (Fua et al., 1999a) we introduced a coloring strategy called proximity-based coloring to assign colors to clusters. It maps colors by cluster proximity based on the structure of the tree, and has the following properties:

- sibling clusters have nearly the same color,
- a parent cluster has a color within the range of its children's colors,
- the color space is effectively utilized, i.e., there are no significant parts of the color space to which no cluster is assigned, and
- differences in color between non-sibling clusters are readily discernable compared to the difference between siblings.

Such color coding is achieved by imposing a linear order on all clusters in the tree and assigning colors to each cluster by indexing into a linear colormap table. Details of the algorithm can be found in (Fua et al., 2000).

2. Hierarchical Display Techniques

We have extended four traditional multivariate display techniques, namely parallel coordinates, star glyphs, scatterplot matrices, and dimensional stacking, to convey cluster information in the manner outlined in the previous section. A key point to make is that the generic approach was indeed easily applicable to each of the four traditional display techniques, thus convincing us that the general approach is a sound one.

2.1 Hierarchical Parallel Coordinates

In traditional parallel coordinates (Figure 1, left), each dimension is represented as a uniformly spaced vertical axis. A data item in this multidimensional space is mapped to a polyline that traverses across all the axes. We generate hierarchical parallel coordinates from the flat form parallel coordinates using the cluster summarization information. In the hierarchical parallel coordinates (Figure 1, right), the clusters rather than individual data items are displayed. The mean of a cluster is mapped to a polyline traversing across all the axes, with a band around it depicting the extents of the cluster. The lower edge of the band intersects each axis at the minimum value of its respective cluster in that dimension. The upper edge of the band intersects each axis at the maximum value of its respective cluster in that dimension. Obviously, if we display each data item included in that cluster, they will all be inside the band. Notice that even if two polylines intersect each other at some axis, we now can easily differentiate them because they have different colors (see Section 1.3).

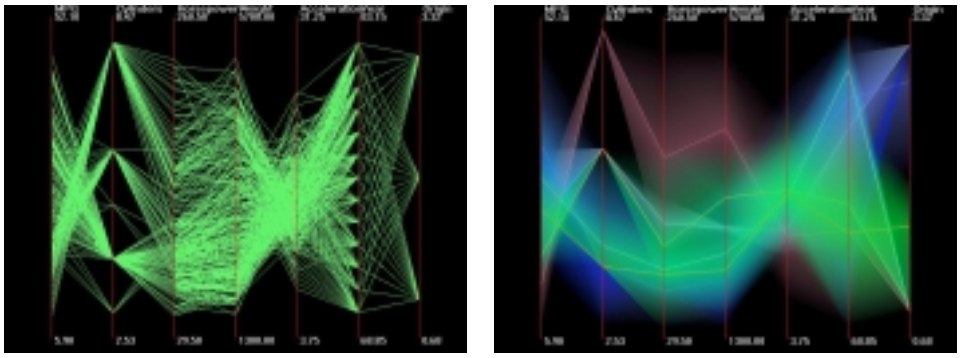

Figure 1. Flat and hierarchical parallel coordinates display of Cars data set (Obtained via anonymous ftp from unix.hensa.ac.uk in the directory /pub/statlib/datasets).

2.2 Hierarchical Glyphs

In the traditional form of star glyphs (Figure 2, left), each glyph presents a single data item. The data values are mapped to the length of rays emanating from a central point, and the ends of the rays are linked to form a polygon. We can view these rays as axes, with each axis presenting a dimension. Once again, we generate hierarchical glyphs from the flat form glyphs using cluster information. In the hierarchical glyphs (Figure 2, right), each star glyph presents a cluster. The mean

values of a cluster are mapped to the length of rays emanating from a central point in the star glyph. The ends of these rays are linked to form the mean polygon. The band around the mean polygon has two edges; one is outside the mean polygon and another one is inside the mean polygon. The inside edge intersects each axis at the minimum value of its respective cluster in that dimension, while the outside edge would intersect each axis at the maximum value of its respective cluster in that dimension if we extended the axes. Obviously, if we were to draw a star glyph starting from this center point to present a data item included in that cluster, this star glyph would be inside the band of that cluster.

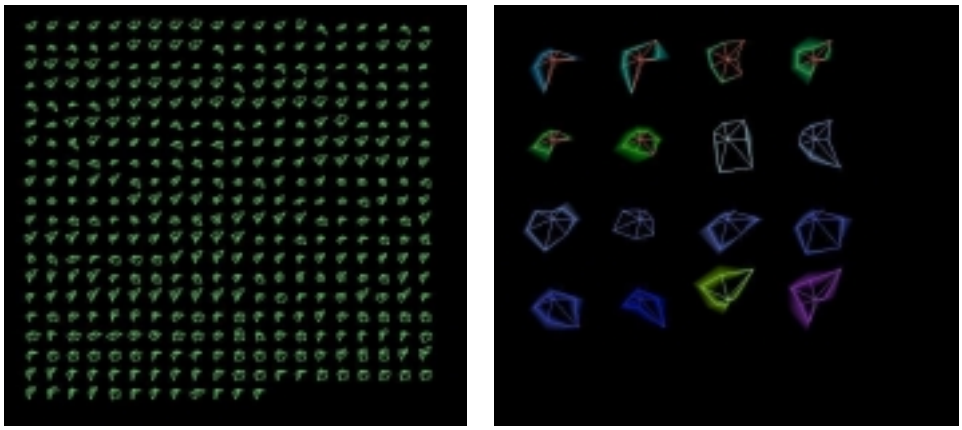

Figure 2. Flat and hierarchical star glyphs display of Cars data set.

2.3 Hierarchical Scatterplot Matrices

In traditional scatterplot matrices (Figure 3, left), each data item is projected to N*N plots (N is the number of dimensions), each of which is a pairwise projection of the data set. The position of the projected point in a plot is decided by the values of the data item in the two dimensions that comprise this plot. We generate a hierarchical scatterplot matrix from the flat form scatterplot matrix using the cluster information. In the hierarchical scatterplot matrix (Figure 3, right), the clusters are shown in the N*N plots. The mean of a cluster is projected to the N*N plots as an ordinary data item in the flat form scatterplot matrix. The extent of the cluster is also projected to the N*N plots, forming rectangles around the projected mean. The projections of the same cluster on different plots are colored in the same way, which helps users link clusters from one plot to another. In the non-hierarchical scatterplot

matrix, all data items have the same color, hence users can find it difficult to trace a data item between plots in those traditional displays.

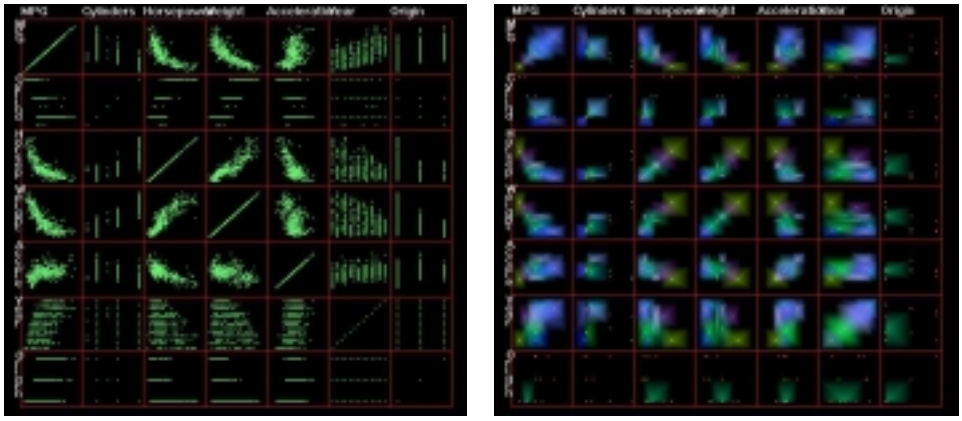

Figure 3. Flat and hierarchical scatterplot matrices display of Cars data set.

2.4 Hierarchical Dimensional Stacking

Traditional dimensional stacking (Figure 4, left) displays an N dimensional data set by recursively embedding dimensions within one another. The range of each dimension is broken into a user-selected number of discrete bins, and the screen space is divided into a grid of sub-images based on the number of bins for two of the dimensions. These sub-images are then decomposed based on the bin count of two more dimensions, and the process continues. Each small block on the final display thus represents a discrete position in the N dimensional space. A data item will fall into one of these small blocks. We generate the hierarchical dimensional stacking from the flat form dimensional stacking using the approach followed with the other display methods. In the hierarchical dimensional stacking (Figure 4, right), the clusters replace the data items. The mean of a cluster will fall into a single small block similar to where an ordinary data item in the flat form dimensional stacking would be placed. The band of this cluster depicts the extent of the cluster. This time it is possible that some parts of the band are disjoint from others in display space due to the nature of the dimensional stacking technique.

3. Interactive Tools

Having introduced methods for mapping hierarchical cluster information into a visualization, we now need to tackle the problem of how to give users the power to interactively perform their data exploration

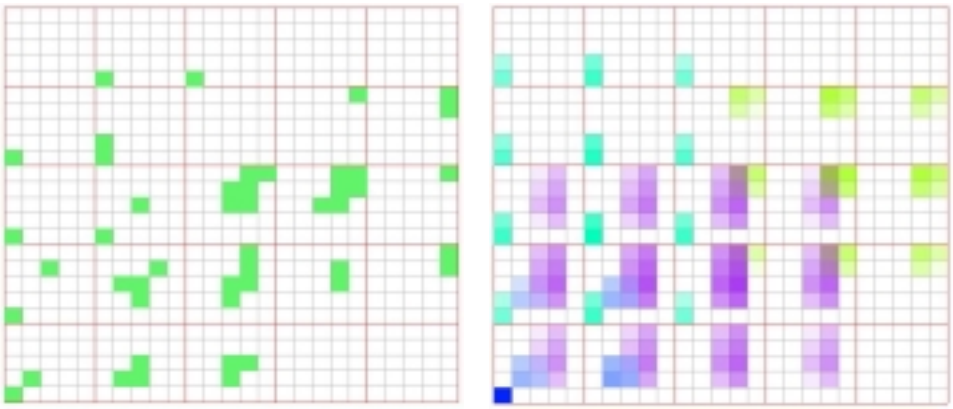

Figure 4. Flat and hierarchical dimensional stacking display of Cars data set.

tasks. We have developed a suite of interactive tools, such as the structure-based brush, drill-down/roll-up operations, extent scaling, and dynamic masking, to reduce clutter and interactively explore the hierarchical displays.

Structure-based Brush: Users often need to explore a particular subspace of interest after obtaining an overview of the hierarchical structure. One way of achieving this is though *brushing*. Brushing is a *direct* and *data-driven* metaphor. It is an interactive process for selecting subsets of data or localizing a subspace within an N-dimensional space (Martin and Ward, 1995; Wong and Bergeron, 1996; Ward, 1994). Many useful operations such as highlighting, deleting, masking or aggregation may be performed on elements that lie within the selected subspace. Brushing has traditionally been performed in either *screen space* or *data space*. One example of brushing in screen space is the use of rubber-banding rectangles; an example of brushing in data space is interactively creating hyperboxes by painting over data points of interest (Martin and Ward, 1995).

Since brushing is useful, we want to keep it in the interactive hierarchical displays. However, brushing in screen space or data space cannot perform necessary selection operations in our hierarchical displays. According to the fact that the hierarchical cluster tree is highly structured, we have developed the concept of a *structure-based brush*. A structure-based brush allows users to select subsets of the hierarchical structure by specifying focal extents as well as a levels-of-detail on a visual representation of the structure. Details of the structure-based brush can be found in (Fua et al., 1999b; Fua et al., 2000).

Drill-down/Roll-up Operations: Drill-down/roll-up operations allow users to change the level of detail of our hierarchical displays intuitively and directly. Drill-down refers to the process of viewing data at a level of increased detail, while roll-up refers to the process of viewing data with decreased detail (Fua et al., 1999a). When users perform drill-down, the number of clusters in the display increases because the visible clusters split into smaller clusters. Conversely, when users perform roll-up, the number of clusters in the displays decreases because the displayed clusters merge together and form larger clusters.

We couple our drilling operations with brushing. Our system permits selective drill-down/roll-up of the brushed and non-brushed region independently. This flexibility is important as it allows the viewing of a subset of elements in varying levels of detail while maintaining the overall context.

Extent Scaling: Though the bands of the clusters on the hierarchical displays are translucent, it is often difficult to isolate or to tell them apart when they are overlapping. Moreover, it is possible that the users may want to see the bands indicating the relative scale of the clusters but do not want to see them covering a large portion of the screen. One way to solve this problem is to decrease the extents of all the bands in each dimension by scaling them uniformly via a dynamically controlled extent scaling parameter $E \in [0, 1]$. E affects the extents of the bands in this way:

$$bandExtent_i \quad = \quad E * clusterExtent_i \tag{1}$$

where i refers to the identity of a cluster T_i, $clusterExtent_i$ refers to the distances from the mean to the maximum and minimum value of cluster T_i in each dimension, and $bandExtent_i$ refers to the distances from the mean point to the maximum and minimum value of the displayed band of cluster T_i in each dimension.

Dynamic Masking: *Dynamic masking* refers to the ability to control the relative opacity between brushed and unbrushed clusters. It allows users to deemphasize or even eliminate brushed or unbrushed clusters. With dynamic masking, the viewer can interactively fade out the bands of the unbrushed clusters, thereby obtaining a clearer view of the brushed clusters while maintaining context regarding unbrushed areas. Conversely, the bands of the brushed clusters can be faded out, thus obtaining a clearer view of the unbrushed region. Used together with the structure-based brush, dynamic masking reduces the overlap and density of the clusters on the screen by fading out uninteresting clusters.

4. Related Work

In recent years, many research efforts have focused on the problem of clutter when visualizing large multivariate data sets.

Wong and Bergeron (Wong and Bergeron, 1996) have constructed a multiresolution display using wavelet approximations, where the data size is reduced by repeatedly merging neighboring points. Their approach is to construct hierarchical structures using a wavelet transform and view different levels of detail interactively upon the hierarchical structures. However, the wavelet transform requires the data to be ordered, making it useful only for data sets with a natural ordering, such as time-series data.

Another approach is to let the characteristics of the data set reveal itself. For example, Wegman and Luo (Wegman and Luo, 1997) suggest over-plotting translucent data points or lines so that sparse areas fade away while dense areas appear emphasized. The disadvantage of this method is that it relies on overlapping points or lines to identify clusters. Clusters without overlapping elements will not be visually emphasized.

Keim et al. (Keim et al., 1995) studied pixel-level visualization schemes which permit the display of a large number of records on a typical workstation screen based on recursive layout patterns. In this case, the number of displayable records is dependent on the size of the display area. This limitation restricts the scalability of their method. Moreover, since each pixel only represents one variable, it is difficult to convey the interactions among variables. We instead take a different approach of preserving the look and feel of well-known traditional display techniques in as much as possible, and addressing the clutter problem by abstracting the data itself into several different levels of detail.

5. Conclusions and Future Work

In this paper we have presented a generalized strategy for visually depicting and exploring very large multivariate data sets. All of the visualization, navigation, and filtering methods we described have been implemented in the XmdvTool system (Ward, 1994), a public-domain software package developed at WPI (see `http://davis.wpi.edu/~xmdv`). The system has been successfully applied to the analysis of data in a wide range of disciplines, including the earth and space sciences, economics and business, statistics, optimization, and performance analysis.

Our current and future work will focus on both the evaluation and refinement of the techniques presented, addressing issues such as coping with very deep or broad hierarchies, alternate navigation tools, and database strategies to optimize system performance. Result of some of

our evaluation experiments can be found in (Yang et al., 2001) We are also interested in investigating methods for handling very large numbers of data dimensions, as most techniques lose their effectiveness when the dimensionality exceeds 20-30. Finally, we are exploring other domains to which the techniques can be applied, as this often leads to new avenues for research.

Acknowledgments

The authors wish to thank Ying-Huey Fua, Daniel Stroe, Punit Doshi, and Geraldine Rosario for their contributions to the XmdvTool Project. This work was partially supported under U.S. National Science Foundation grants IIS-9732897, IRIS-9729878, and IIS-0119276.

References

Andreae, P., Dawkins, B., and O'Connor, P. (1990). Dysect: An incremental clustering algorithm. *Document included with public-domain version of the software, retrieved from Statlib at CMU.*

Andrews, D. (1972). Plots of high dimensional data. *Biometrics, Vol. 28, p. 125-36.*

Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association, Vol. 68, p. 361-68.*

Cleveland, W. and McGill, M. (1988). *Dynamic Graphics for Statistics.* Wadsworth, Inc.

Fua, Y., Ward, M., and Rundensteiner, E. (2000). Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. *IEEE Visualization and Computer Graphics, Vol. 6, No. 2, p. 150-159.*

Fua, Y., Ward, M., and Rundensteiner, E. (Oct. 1999a). Hierarchical parallel coordinates for exploration of large datasets. *Proc. of Visualization '99, p. 43-50.*

Fua, Y., Ward, M., and Rundensteiner, E. (Oct. 1999b). Navigating hierarchies with structure-based brushes. *Proc. of Information Visualization '99, p. 58-64.*

Guha, S., Rastogi, R., and Shim, K. (June 1998). Cure: an efficient clustering algorithm for large databases. *SIGMOD Record, vol.27(2), p. 73-84.*

Inselberg, A. and Dimsdale, B. (1990). Parallel coordinates: A tool for visualizing multidimensional geometry. *Proc. of Visualization '90, p. 361-78.*

Jain, K. and Dubes, C. (1988). *Algorithms for Clustering Data.* Prentice Hall.

Keim, D., Kriegel, H., and Ankerst, M. (1995). Recursive pattern: a technique for visualizing very large amounts of data. *Proc. of Visualization '95, p. 279-86.*

LeBlanc, J., Ward, M., and Wittels, N. (1990). Exploring n-dimensional databases. *Proc. of Visualization '90, p. 230-7.*

Martin, A. and Ward, M. (1995). High dimensional brushing for interactive exploration of multivariate data. *Proc. of Visualization '95, p. 271-8.*

Ribarsky, W., Ayers, E., Eble, J., and Mukherjea, S. (1994). Glyphmaker: Creating customized visualization of complex data. *IEEE Computer, Vol. 27(7), p. 57-64.*

Ward, M. (1994). Xmdvtool: Integrating multiple methods for visualizing multivariate data. *Proc. of Visualization '94, p. 326-33.*

Wegman, E. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association, Vol. 411(85), p. 664.*

Wegman, E. and Luo, Q. (1997). High dimensional clustering using parallel coordinates and the grand tour. *Computing Science and Statistics, Vol. 28, p. 361-8.*

Wong, P. and Bergeron, R. (1996). Multiresolution multidimensional wavelet brushing. *Proc. of Visualization '96, p. 141-8.*

Yang, J., Ward, M., and Rundensteiner, E. (2001). Interactive hierarchical displays: A general framework for visualization and exploration of large multivariate data sets. *Technical Report WPI-CS-TR-01-22.*

Zhang, T., Ramakrishnan, R., and Livny, M. (June 1996). Birch: an efficient data clustering method for very large databases. *SIGMOD Record, vol.25(2), p. 103-14.*