6-1-2002

Visual Hierarchical Dimension Reduction for Exploration of High Dimensional Datasets

Jing Yang
Worcester Polytechnic Institute, yangjing@cs.wpi.edu

Matthew O. Ward
Worcester Polytechnic Institute, matt@cs.wpi.edu

Elke A. Rundensteiner
Worcester Polytechnic Institute, rundenst@cs.wpi.edu

Visual Hierarchical Dimension Reduction for Exploration of High Dimensional Datasets

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

Traditional visualization techniques for multidimensional data sets, such as parallel coordinates, glyphs, and scatterplot matrices, do not scale well to high numbers of dimension. A common approach to solving this problem is dimensionality reduction. Existing dimensionality reduction techniques usually generate lower dimensional spaces that have little intuitive meaning to users and allow little user interaction. In this paper we propose a new approach to handling high dimensional data, named Visual Hierarchical Dimension Reduction (VHDR), which addresses these drawbacks. VHDR not only generates lower dimensional spaces that are meaningful to users, but also allows user interactions in most steps of the process. In VHDR, dimensions are grouped into a hierarchy, and lower dimensional spaces are constructed using clusters of the hierarchy. We have implemented the VHDR approach into XmdvTool, and extended several traditional multidimensional visualization methods to convey dimension cluster characteristics when visualizing the data set in lower dimensional spaces. Our case study of applying VHDR to a real data set confirms that this approach is effective in supporting the exploration of high dimensional data sets.

Keywords: Dimension reduction, high dimensional visualization, visual data mining.

1 Introduction

High dimensional data sets are becoming commonplace in an increasing number of applications, including digital libraries, simulations, and surveys. It is no longer unusual to have data sets with hundreds or even thousands of dimensions. However, traditional visualization techniques for multidimensional data sets, such as glyph techniques [1, 19, 4, 18], parallel coordinates [12, 21], scatterplot matrices [5], and pixel-level visualization [15], do not scale well to high dimensional data sets. For example, Figure 1 shows the Iris data set, which has 4 dimensions and 150 data items, displayed using parallel coordinates. Individual data items and clusters can be seen clearly from the display. Figure 2 shows a subset of the Census Income data set [10], which has 42 dimensions and 200 data items. While the number of data items in this display is comparable to Figure 1, individual data items can no longer be seen clearly from this display, since the number of dimensions has greatly increased. A large number of axes now crowd the figure, preventing users from detecting any patterns or details. Even with low numbers of data items, high dimensionality presents a serious challenge for current display techniques.

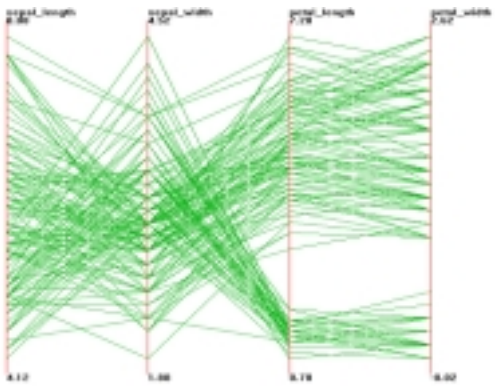

Figure 1: Iris data set (4 dimensions, 150 data items) in Parallel Coordinates. Individual data items can be seen clearly.

To overcome the clutter problem, one promising approach frequently described in the literature is dimensionality reduction [9]. This idea is to first reduce the dimensionality of the data while preserving the major information it carries, and then visualize the data set in the reduced dimensional space. There are several popular dimensionality reduction techniques used in data visualization, including Principal Component Analysis (PCA) [13], Multidimensional Scaling (MDS) [17], and Kohonen's Self Organizing Maps (SOM) [16, 6]. These approaches have a major drawback in that the generated low dimensional subspace has no intuitive meaning to users. In addition, little user interaction is generally allowed in those highly automatic processes, thus users have difficulty applying their domain acknowledge to improve the quality of the dimensionality reduction process.

The clutter problem not only exists in visualizing data sets with high dimensionality, but also when visualizing data sets with a large number of data items. Our previous work has addressed the clutter problem in the latter situation using an Interactive Hierarchical Display (IHD) framework [7, 8, 24]. The underlying principle of this framework is to develop a multi-resolution view of the data via hierarchical clustering, and to design extensions of traditional multivariate visualization techniques to convey aggregation information about the resulting clusters. Users can then explore their desired focus regions at different levels of detail, using our suite of navigation and filtering tools [7, 8].

Inspired by the IHD framework, we now propose a new methodology for dimensionality reduction that combines au-

*This work is supported under NSF grants IIS-9732897, IRIS-9729878, and IIS-0119276.

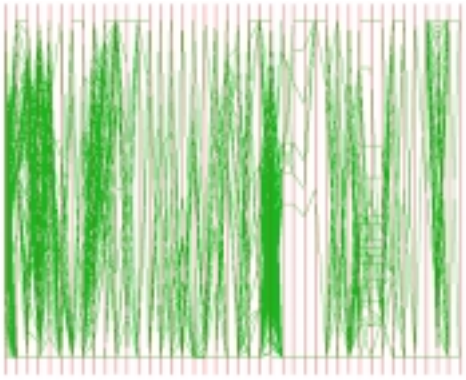

Figure 2: A subset of the Census Income data set (42 dimensions, 200 data items) in Parallel Coordinates. Individual data items cannot be seen clearly.

tomation and user interaction for the generation of meaningful subspaces, called the Visual Hierarchical Dimension Reduction (VHDR) framework. First, VHDR groups all dimensions of a data set into a dimension hierarchy. This hierarchy is then visualized using a radial space-filling hierarchy visualization tool, named InterRing. Users can interactively explore and modify the dimension hierarchy, and select interesting dimension clusters at different levels of detail for the data display. A representative dimension for each selected dimension cluster is then determined automatically by the system or interactively by the users. Finally, VHDR maps the high-dimensional data set into the subspace composed of these representative dimensions and displays the projected subspace using traditional multidimensional displays. We have implemented a fully functional prototype of the above approach, as well as several extensions to traditional multidimensional visualization methods that convey dimension cluster characteristics when visualizing the data set in lower dimensional spaces. We have applied VHDR to several real data sets using our prototype and found that this approach is helpful in exploring high dimensional data sets. The prototypes have been integrated into XmdvTool, a public-domain visualization package developed at WPI [20].

VHDR employs the process of dimension clustering, which is orthogonal to data clustering approaches. Thus it has no conflicts with data clustering intended to cope with large-scale data sets. We have combined the VHDR approach with our previous hierarchical data visualizations easily and did not encounter any fundamental problems.

The remainder of this paper is organized as follows: Section 2 provides an overview of the VHDR approach, while Sections 3 through 7 describe details of each stage of the VHDR pipeline. Section 8 presents a case study that uses VHDR to explore a high dimensional data set. Section 9 presents related work, and Section 10 summarizes our work and presents open issues for future work.

2 Overview of Visual Hierarchical Dimension Reduction

Figure 3 shows the system structure of the VHDR framework. The methodology can be divided into five steps:

- Step 1: Dimension Hierarchy Generation

  First, all the original dimensions of a multidimensional data set are organized into a hierarchical dimension cluster tree according to similarities among the dimensions. Each original dimension is mapped to a leaf node in this tree. Similar dimensions are placed together and form a cluster, and similar clusters in turn compose higher-level clusters. Users have the option of using the system-provided automatic clustering approach, of using their customized clustering approaches, or specifying a hierarchical dimension cluster tree manually.

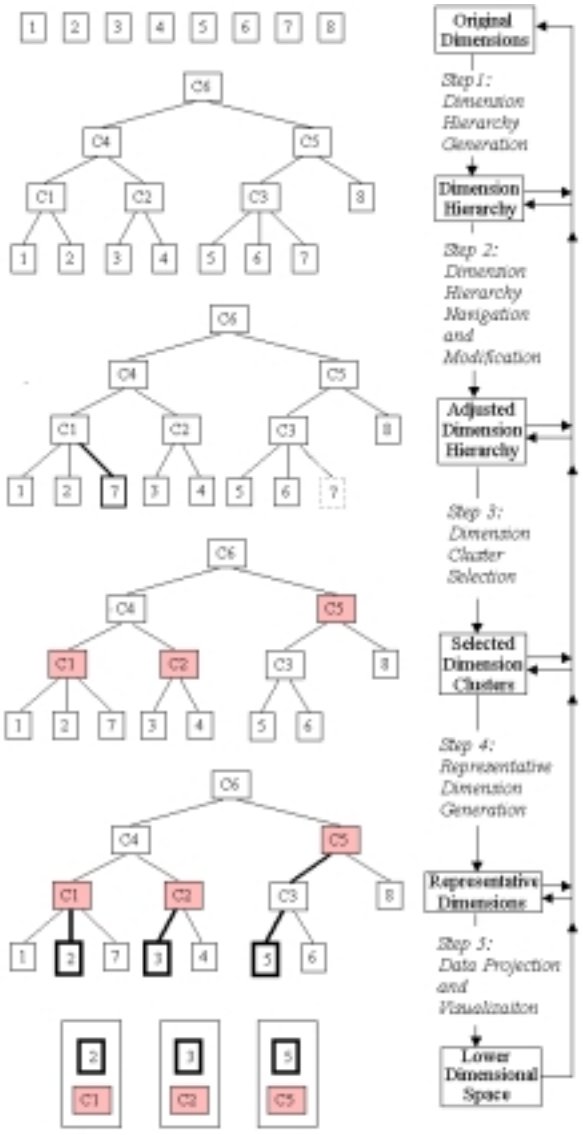

Figure 3: System Structure of VHDR

- Step 2: Dimension Hierarchy Navigation and Modification

  Next, users can navigate through the hierarchical dimension cluster tree in order to gain a better understanding of it. Users can also interactively modify the hierarchy structure and supply meaningful names to the clusters. The hierarchical dimension cluster tree is visualized in a radial space-filling display named InterRing, which contains a suite of navigation and modification tools.

- Step 3: Dimension Cluster Selection

  Next, users interactively select interesting dimension clusters from the hierarchy in order to construct a lower dimensional subspace. Several selection mechanisms are provided in InterRing to facilitate dimension cluster selection. Selected clusters are highlighted.

- Step 4: Representative Dimension Generation

  In this step, a representative dimension (RD) is assigned or created for each selected dimension cluster. The selected dimension clusters construct the lower dimensional space through these RDs. RDs are selected to best reflect the aggregate characteristics of their associated clusters. For example, an RD can be the average of all the original dimensions in the cluster, or can be an original dimension located in the center of the cluster. Users have the option to select one of the system-provided RD generation methods or use a customized one.

- Step 5: Data Projection and Visualization

  Finally, the data set is projected from the original high dimensional space to a lower dimensional space (LD space) composed of the RDs of the selected clusters. We call its projection in the LD space the *mapped data set*. The mapped data set can be viewed as an ordinary data set in the LD space and can be readily visualized using existing multidimensional visualization techniques. This is an advantage of VHDR; it is so flexible that it can be applied to any existing multidimensional data visualization technique. In order to provide further dimension cluster characteristics in the LD space, such as the dissimilarity information between dimensions within a cluster, we attach the dimension cluster characteristics information to the mapped data set and provide the option to display it using extensions to the data visualization techniques.

Users can return to any of the previous steps at any time to refine the process iteratively. For example, after the LD space has been generated, users can still modify the dimension hierarchy, redo the selection, and generate a different LD space.

One significant advantage of the VHDR approach is that the generated lower dimensional space is meaningful to users, in that:

- As long as users understand the similarity measure used in dimension clustering, the meaning of the dimension hierarchy is straightforward.

- Through dimension hierarchy visualization and navigation, if users find that a cluster contains some dimensions that have no semantic relationship with other dimensions in the cluster according to their knowledge of the data domain, they can manually remove them from this cluster. Similarly, they can merge two clusters that are semantically related. Thus each cluster can have a clear domain-specific meaning guided by a domain expert.

- Users can label the dimension clusters with “meaningful” names, thus helping the interpretation of the dimension clusters during later data exploration.

- Users interactively select the dimension clusters to be visualized in the LD space according to their knowledge of the data domain. As a result, the structure of the LD space is meaningful to them.

- Users can select the RD generation approach or even explicitly select the RDs, hence they know how the RDs are generated and how to interpret them.

Another advantage of the VHDR approach is that it integrates automatic and interactive techniques. As a tool designed to assist people in visualizing high dimensional data sets, the VHDR approach avoids trivial and boring manual work during the majority of the process. At the same time, an interactive visualization approach has been introduced into the dimension reduction process to allow users to apply their domain knowledge. This combination of automatic and interactive methods is reflected in that:

- VHDR provides an automatic dimension clustering approach. Users can adjust the automatic approach by replacing the similarity measure calculation methods by their own similarity measures or changing other algorithmic parameters in the system. They can also use their own dimension clustering approach by customizing the system, or even provide a dimension hierarchy directly by creating a dimension hierarchy file in the appropriate format.

- VHDR provides a tool suite for interactively navigating and modifying the dimension hierarchy.

- VHDR provides automatic and manual brushing mechanisms in the dimension hierarchy visualization. Users can interactively select interesting dimension clusters using a combination of the different brushing approaches.

- VHDR provides several alternative RD generation approaches. Users can interactively select one of them. Users can also customize the prototype to add more RD generation approaches.

- VHDR provides several options to visualize the dissimilarity information of the dimension clusters in the RD space. Users can interactively select one of them or turn this information on/off.

In the following sections, we will describe the details of each of the above five steps of VHDR.

3 Dimension Hierarchy Generation

Many possible dimension clustering algorithms, such as factor analysis or algorithms adapted from multidimensional data clustering techniques, could be employed to form a dimension hierarchy. We have implemented a bottom-up agglomerative dimension clustering algorithm in our prototype. The following is a description of this algorithm:

- Similarity Measure: To avoid complex calculations in order to cope with large scale data sets, we judge if the similarity between a pair of dimensions is below a similarity threshold using a simple counting approach rather than calculating its absolute value. More precisely, we count the number of data items in the data set whose normalized values of these two dimensions have differences less than the similarity threshold. If this number exceeds a certain percentage of the total data points in the data set, we regard the similarity of these two dimensions as being below the similarity threshold. For a dimension cluster, we use the average of all its decedent dimensions as its value for difference calculation.

- Data Clusters: To cope with large scale data sets, we make use of partial results of a bottom-up data clustering algorithm applied on the data set. The idea is that if several data items form a dense cluster, we can use the cluster center in the similarity measure as long as we use the number of data items included in the cluster in the counting mentioned above. In practice, we select all the data clusters with extents smaller than a specified threshold. These clusters contain all the data items in the data set exactly once. Then we use these data clusters instead of the original data items to measure similarities between dimensions. Since the precondition of this approach is that there exists a data hierarchy, which is common in our tool but may not be available in other tools, we only provide this approach as an option to the users.

- Iterative Clustering: To form the hierarchical tree structure, we define a series of decreasing similarity thresholds corresponding to bottom-up clustering iterations. In each iteration, dimensions that have not formed any clusters and clusters formed from the previous iteration are considered. If any pair of them has a similarity below the similarity threshold, the pair is recorded as a *similar pair*. Then the dimension or cluster that forms the largest number of similar pairs is extracted as a new cluster center. All the other dimensions and clusters that form similar pairs with it are put into the new cluster and their similar pairs are deleted. Repeating the above approach will form more new clusters. The current iteration ends when there are no similar pairs left. The whole clustering process terminates when all the dimensions have been included into a root cluster.

Our dimension clustering algorithm is a simplistic process. However, it satisfies our need in that it computationally inexpensive even when handle large-scale data sets with high dimensionality and large numbers of records, and generates dimension hierarchies that appear reasonable for the real data sets we tested.

4 Dimension Hierarchy Navigation and Modification

4.1 Dimension Hierarchy Navigation

It is important for users to be able to navigate the dimension hierarchy to get a better understanding of it. Navigation tools are also useful in dimension cluster selection by making the selection process easier. In VHDR we use a radial space-filling hierarchy visualization tool named InterRing to navigate the dimension cluster tree (See Figure 6 (a)). Details of InterRing can be found in [25].

In InterRing, we use the color of the nodes to redundantly convey the hierarchical structure of the dimension cluster tree. We have designed and implemented a color assignment approach according to the following principles:

- nodes belonging to the same cluster should have similar colors;

- a parent node's color should be related to the colors of its children; and

- a larger child (a cluster with a higher population) contributes more to its parent's color than its smaller siblings.

To give users focus + context interactivity, we provide a multi-focus distortion operation in InterRing. In this distortion, a focal area enlarges to occupy some of the space of its siblings. Several foci can coexist and a focus can be a descendant of another focus. The distortion operation involves a simple "drag and drop" action. No extra space or windows are used in this distortion technique.

There are many other useful navigation and filtering tools in InterRing, including rotation, drilling-down/rolling-up, and zooming/panning. The rotation operation allows users to rotate the InterRing display around its center in both directions to allow dimension labels to be more readily discerned. Drilling-down/rolling-up allows users to hide/show all the descendants of a cluster. Zooming in/zooming out and panning operations allow users to enlarge the InterRing and move around to examine details.

4.2 Dimension Hierarchy Modification

In interactive hierarchy modification, users can change the dimension cluster tree by removing a sub-cluster from a cluster and dropping it into another cluster. We provide this modification function to users for two reasons:

- Although users can adjust the dimension clustering process by setting different parameters and changing similarity calculation methods, it is still an automatic process. This does not enable users to interactively take part in this process.

- Users are often experts of the data sets being visualized. They usually know that some relationships exist in the data sets based on their experience and knowledge in their fields. These relationships may be undetected by the automatic dimension clustering approach. Hence allowing users to interactively adjust the generated dimension cluster tree benefits the whole process.

5 Dimension Cluster Selection

In order to construct a lower dimensional subspace, users need to select some interesting dimension clusters from the dimension hierarchy. In our implementation, we provide both an automatic and a manual brushing mechanism. The automatic method is called "structure-based brushing". It allows users to select/unselect multiple clusters with similar sizes or dissimilarities. The manual method is called "simple brushing". It involves simple clicking to select or unselect one node each time. The combination of these two different mechanisms makes the selection flexible. To find more details of the brushing, please refer to [25].

Selected dimension clusters are highlighted using the XmdvTool system highlighting color in the middle area of the nodes. Their cluster names are explicitly shown, while the names of the unselected clusters are shown only when the cursor passes over them.

6 Representative Dimension Generation

The selected dimension clusters form a low dimensional (LD) space. However, the question arises as to how we can visualize a dimension cluster in the LD space. We generate a representative dimension (RD) for each selected cluster and visualize it in the LD space as a reasonable representation of the dimension cluster.

RDs can be either assigned or created. We have developed several different approaches to assigning or creating RDs. Assignment approaches include selecting a dimension located in the center of a cluster, user-selected RDs, or randomly selecting one dimension from a cluster. Creation approaches include using the

weighted average of all the dimensions in a cluster as its RD, and applying Principal Component Analysis (PCA) to all the dimensions in a cluster and using the first principal component as its RD. Analyzing these approaches we find that:

- The RDs generated by the assignment approaches have clear meaning to the user, but for outliers that change greatly within the dimension cluster, the variance is hidden from the user. The drawback can be overcome by visualizing the dissimilarity information of the cluster in the data display (see Section 7.2), or checking the dimension cluster in detail by constructing an LD space composed of the leaf nodes of this cluster.

- The RDs generated by the creation approaches reflect the overall structure of the cluster better than the assignment approaches, but their meaning is not as explicit as those generated by the assignment approaches. However, it is much better than applying PCA to the whole data set, since the number of dimensions in the whole data set is generally much larger than in a dimension cluster, and some dimensions of the whole data set could have no meaningful interrelationships.

7 Data Mapping and Visualization

Having selected a set of dimension clusters and generated RDs for them, we construct a lower dimensional space using these RDs with/without dissimilarity information of the selected clusters. The original data set is projected into this lower dimensional space.

To better convey dimension cluster characteristics, we provide extensions to glyph [1, 19, 4, 18], parallel coordinates [12, 21], and scatterplot matrices [5] to allow RDs to convey cluster dissimilarity information in the LD space. We discuss dissimilarity visualization in Section 7.2.

7.1 Data Projection

In VHDR, the system maintains a selected cluster list (SC list) and the current RD generation approach. A new LD space is constructed when the SC list is updated or the RD generation approach is changed. Hence we meet the problem of when and how to project the data set from the original high dimensional space to the LD space. We have explored two options:

- Option 1: Every time the SC list is updated or the RD generation approach is changed, project all the data items of the data set from the original high dimensional space to the new LD space, one by one, and store their projected values into a data structure in memory. When redrawing the multi-dimensional displays, directly read from this data structure as input for the multi-dimensional displays. We call this approach *pre-mapping*;

- Option 2: In every display redrawing event, for every data item, we read it from the original data set, map it to the lower dimensional space according to the SC list, and then draw the mapping result. We call this approach *online-mapping*.

The pre-mapping approach does the mapping once for every SC list or RD update, no matter how many times we redraw the displays. However, it requires memory to store the mapping results. Memory is a critical resource when displaying large data sets, since the original data set already occupies a large amount of memory. Moreover, when users update the SC list often (which can happen when users are in search of the most interesting lower dimensional spaces), this method's advantage in time savings will be diminished.

The online-mapping approach needs no extra memory. But on a first glance, it wastes time to recalculate the mapping result. However, compared to the time used to draw a data item on the screen, the time needed to calculate a mapping of that data item is negligible; we have observed no significant difference between the response times of the two mapping approaches. For this reason we adopted the online-mapping approach in our system.

7.2 Dissimilarity Visualization

Users are often concerned about the extent to which the dimensions in a dimension cluster are correlated to each other, since an RD is useful only when the dimensions within its cluster are reasonably correlated. We have extended glyph, parallel coordinates, and scatterplot matrix displays so they graphically provide the dissimilarity information of selected dimension clusters to users in the LD space. We call this *dissimilarity visualization*. We can perform dissimilarity visualization from two different viewpoints: from that of the individual data items, or from that of the whole data set. We name the former the "local degree of dissimilarity (LDOD)" and the latter the "global degree of dissimilarity (GDOD)". They are defined as follows:

- LDOD - the degree of dissimilarity for a single data item in a dimension cluster. We use a mean, a maximum, and a minimum value to describe it. The mean is the mapped image of the data item on the representative dimension. The minimum is the minimum value among the values of the data item on all the original dimensions belonging to the dimension cluster. The maximum is the maximum value among the values of the data item on all the original dimensions belonging to the dimension cluster. Note that all the dimensions have been normalized so values lie between 0 to 1.

- GDOD - the degree of dissimilarity for the entire data set in a dimension cluster. It is a scalar value and can be calculated according to the similarity measures between each pair of the dimensions in the cluster. We use a simplified approach, namely, we use directly the radius of a dimension cluster as its GDOD. A dimension cluster radius is initially assigned as the similarity threshold of the iteration in which the dimension cluster is formed in the VHDR automatic dimension cluster approach (Section 3). It can be affected by interactive modification to the hierarchy.

LDOD and GDOD information are useful to users. By studying the LDOD, users can discover outliers that have large LDODs while most data items have small LDODs in a dimension cluster, and can learn the overall dimension cluster correlation information by aggregating LDOD information of all the data items. By examining the GDOD, the overall dimension cluster correlation information will be immediately known, thus users can avoid using clusters with large dissimilarities to form the lower dimensional space. We provide LDOD and GDOD information to users visually so that they can gain a qualitative understanding of cluster characteristics.

Our current approach to visualize GDOD is named the Axis Width Method. In multidimensional visualizations that contain

axes, we make the width of the representative dimensions proportional to the GDOD of the dimension clusters they represent. A wider axis represents a dimension cluster with a larger GDOD. Currently, we have applied this method to parallel coordinates, scatterplot matrices, and star glyphs. In scatterplot matrices, GDOD is mapped to the width of the frames of the plots. Figure 4 (a) shows the Axis Width Method in parallel coordinates.

We have explored several different approaches to visualizing the LDOD:

- Approach 1: The Three-Axes Method for Representing LDOD. which is borrowed from the hierarchical parallel coordinates [7]. The basic idea of this method is to use two extra axes around a representative dimension to indicate the minimum and maximum of the corresponding dimension cluster for every data point. The three-axes method can be applied to parallel coordinates and star glyphs. For parallel coordinates, good correlation within a cluster would manifest itself as nearly horizontal lines through the 3 axes, while lines with steep slope indicate areas of poor correlation (see Figure 4 (b)).

- Approach 2: The Mean-Band Method for Representing LDOD, which is borrowed from hierarchical parallel coordinates [7]. A band is added to each data point ranging in width from the minimum to the maximum for each representative dimension. Narrow bands indicate a good correlation, while wide bands indicate a bad correlation. This method can be applied to parallel coordinates, scatterplot matrices, and star glyphs. However, it suffers from the overlaps introduced by the bands (see Figure 4 (c)).

- Approach 3: Diagonal Plots for Representing LDOD in Scatterplot Matrices. Having observed the fact that the diagonal plots (mapping a dimension against itself) in the scatterplot matrix convey little useful information, we map the minimum and maximum of the dimension cluster to the x and y coordinates of the diagonal plot of its representative dimension. Thus in the diagonal plots, if a point has an equal maximum and minimum, it will be represented as a point on the diagonal. On the contrary, if a point has a large LDOD, which means there is a large difference between maximum and minimum and thus a large difference between its x and y coordinates, it will lie a significant distance from the diagonal. Thus a diagonal plots with points spread out in the plot away from the diagonal indicates low correlation within that dimension cluster (see Figure 4 (d)).

8 Case Study

In the above sections, we have presented the VHDR approach as a sequence of steps. In practical use, VHDR is not so rigid. Users can go back to a previous step at any time and begin a new loop to form different LD spaces from different points of view based on their own strategy. For example, a user could explore the overall structure of a data set by forming an LD space using large dimension clusters. Then he could check the detail of some clusters by constructing another LD space using their leaf nodes.

In our case study, we use a 42 dimensional, 20,000 element data set derived from part of the unweighted PUMS census data from the Los Angeles and Long Beach areas for the years 1970, 1980, and 1990 [10]. We will refer to it as the Census dataset. Figure 5 (a) shows the parallel coordinates display of it. It is almost impossible to find any meaningful patterns from the display without dimension reduction.

Figure 6 (a) shows the automatically generated dimension hierarchy of the Census dataset. We explored the overall structure of the hierarchy, and found some clusters with large dissimilarities between each other through structure-based brushing and dissimilarity displays. We assigned a leaf node in each cluster as its RD. These RDs are “education”, “age”, “sex”, “weeks_worked_in_year”, and “income” respectively. We hoped that they could form a lower dimensional space that could reveal the main trend in this data set. In fact, we found many interesting data clusters from this space. For example, Figure 5 (b) shows a group of high-income males who have high education and work most of the year.

Then, we examined the details of a cluster we were interested in (see Figure 6 (b)). It seemed odd to us that some dimensions, such as “region_of_previous_residence” were put together with dimensions such as “income”. According to our experience, we felt that they were not related. Hence we remove the unrelated dimensions from that cluster (see Figure 6 (c)). Then we checked the lower dimensional space composed of all the leaf nodes of that cluster and found out that most, but not all, people of low income have low wage per hour and low capital gain. Several smaller clusters appear as distinct configurations of value ranges for each dimension (Figure 5 (c)).

9 Related Work

There are three major approaches to dimensionality reduction. Principal Component Analysis (PCA) [13] attempts to project data down to a few dimensions that account for most of the variance within the data. Multidimensional Scaling (MDS) [17] is an iterative non-linear optimization algorithm for projecting multidimensional data down to a reduced number of dimensions. Kohonen’s Self Organizing Maps (SOM) [16, 6] is an unsupervised learning method for reducing multidimensional data to 2D feature maps [3].

There are many visualization systems that make use of existing dimensionality reduction techniques [23, 3, 11]. Galaxies and ThemeScape [23] project high dimensional document vectors and their cluster centroids down into a two dimensional space, and then use scatterplots and landscapes to visualize them [22]. Bead [3] uses MDS to lay out high dimensional data in a two or three dimensional space and uses imageability features to visualize the data.

Recently, many new dimensionality reduction techniques have been proposed to process large data sets with relatively high dimensionality. For example, Random Mapping [14] projects the high dimensional data to a lower dimensional space using a random transformation matrix. Kaski [14] presented a case study of a dimension reduction from a 5781-dimensional space to a 90-dimensional space using Random Mapping. The Anchored Least Stress method [26, 22] combines PCA and MDS and makes use of the result of data clustering in the high dimensional space so that it can handle very large data sets. This work inspired us to make use of the data hierarchy in the dimension clustering process.

All the above approaches have the common drawback that their generated display spaces typically have no clear meaning for the users. In the VHDR approach, we reduce the dimensionality in an interactive manner so as to generate a meaningful low dimensional subspace.

Ankerst et al. [2] use similarity measures between dimensions

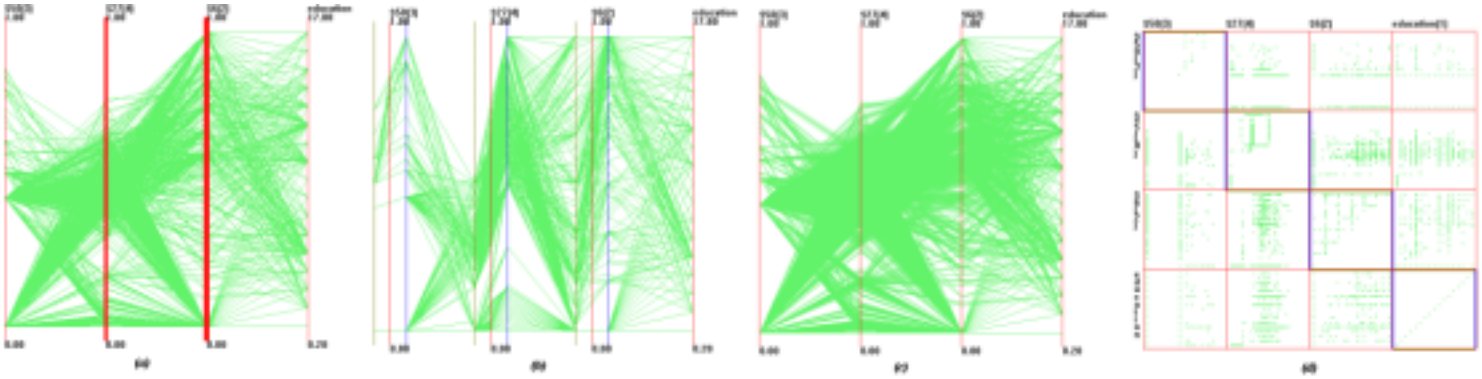

Figure 4: An LD space of the Census Dataset constructed of 4 dimension clusters. These clusters are composed of 3, 4, 2, and 1 original dimensions respectively from left to right of the displays. Figure (a), (b), (c) use parallel coordinates and Figure (d) is a scatterplot matrix. Figure (a) displays the GDODs using the Axis Width Method. Figure (b) displays the LDODs using the Three-Axes Method. Figure (c) displays the LDODs using the Mean-Band Method. The bands have been reduced to 6 percent of their original width. Figure (d) displays the LDODs using the Diagonal Plot Method. From all these figures, it can be noticed that the second cluster and the third cluster have lower correlations than the first and last cluster. Note the number of dimensions in a cluster is embedded in the cluster label.

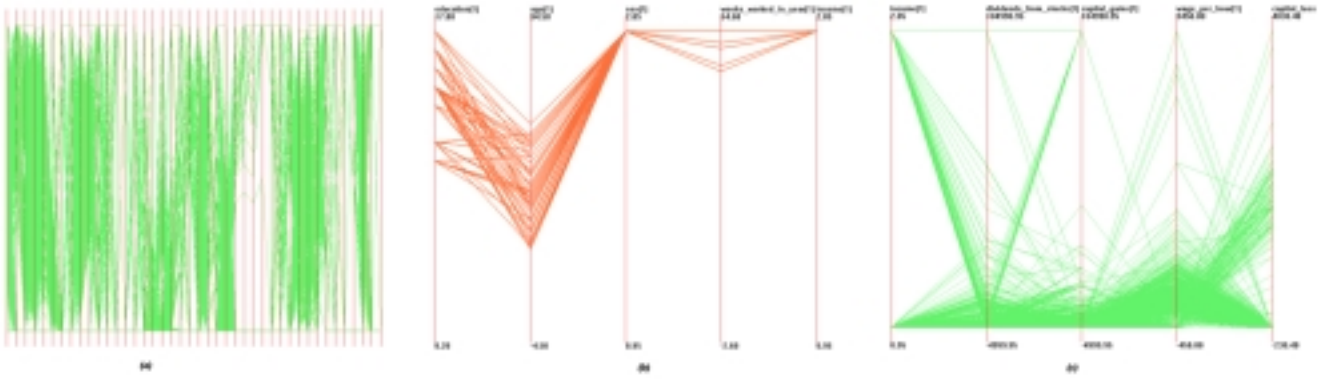

Figure 5: The Census dataset (42 dimensions, 20,000 data items) in parallel coordinates. Figure (a) shows the original high dimensional space. Figure (b) and (c) show two lower dimensional subspaces generated by VHDR.

to order and arrange dimensions in a multidimensional display. They arrange the dimensions so that dimensions showing similar behavior are positioned next to each other. Their work inspired us to use similarity among the dimensions to group them.

10 Conclusion and Future Work

In this paper, we present the VHDR framework, a visual interactive approach for dimension reduction. The main contribution of the VHDR approach is that it can generate lower dimensional spaces that are meaningful to users by allowing them to interactively take part in the dimension reduction process. Other contributions include the mechanisms for conveying cluster dissimilarity. We have implemented the VHDR framework and incorporated it into the XmdvTool software package (`http://davis.wpi.edu/~xmdv`), which will be released to the public domain as XmdvTool Version 6.0. We have applied it to several real data sets and found that it is effective in coping with high dimensional data sets.

In our future work, we will improve this approach in the following aspects:

- implementing and comparing different dimension clustering approaches;
- exploring a dissimilarity display method that could be applied to most, if not all, data visualization techniques in a consistent fashion;
- exploring automated techniques for generating or selecting interesting views of subsets of the hierarchy;
- evaluating the VHDR approach with user studies and experiments and improving the VHDR approach according to the results and feedback from users.

11 Acknowledgements

We gratefully acknowledge our colleagues in the XmdvTool group at WPI for their assistance in this research. Special thanks go to Prof. David Brown, who gave us many valuable suggestions for this work.

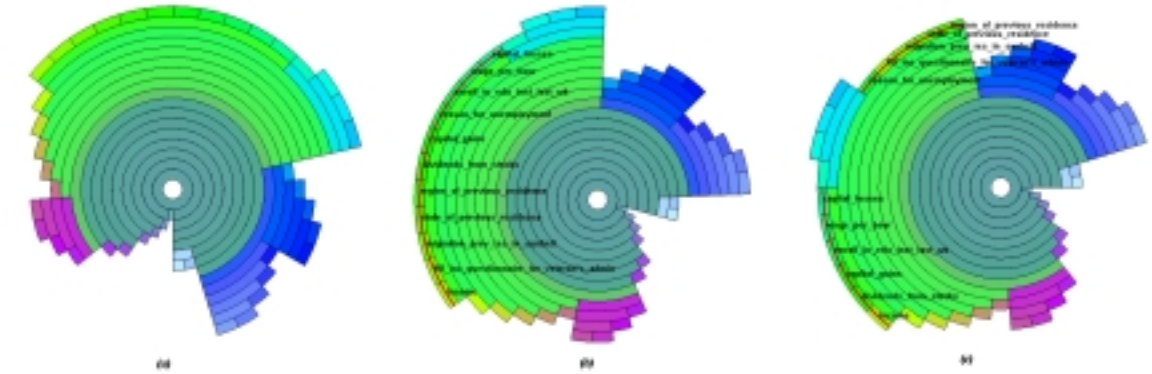

Figure 6: Dimension hierarchy of the Census dataset in InterRing. Figure (a) shows the automatically generated hierarchy. Figure (b) shows the detail of a cluster after brushing and rotation. Figures (c) shows the modified hierarchy after moving some dimensions from that cluster to elsewhere.

References

[1] D. Andrews. Plots of high dimensional data. *Biometrics, Vol. 28, p. 125-36*, 1972.

[2] M. Ankerst, S. Berchtold, and D. A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. *Proc. of IEEE Symposium on Information Visualization, InfoVis'98, p. 52-60*, 1998.

[3] D. Brodbeck, M. Chalmers, A. Lunzer, and P. Cotture. Domesticating bead: Adapting an information visualization system to a financial institution. *InfoVis'97, p. 73-80*, 1997.

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

[5] W. Cleveland and M. McGill. *Dynamic Graphics for Statistics*. Wadsworth, Inc., 1988.

[6] A. Flexer. On the use of self-organizing maps for clustering and visualization. *PKDD'99, p. 80-88*, 1999.

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

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

[9] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[10] S. Hettich and S. D. Bay. The uci kdd archive [http://kdd.ics.uci.edu]. *Irvine, CA: University of California, Department of Information and Computer Science*, 1999.

[11] B. Hetzler, P. Whitney, L. Martucci, and J. Thomas. Multi-faceted insight through interoperable visual information analysis paradigms. *InfoVis'98, p. 137-144*, 1998.

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

[13] J. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[14] S. Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. *Proc. IJCNN, p. 413-418*, 1998.

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

[16] T. Kohonen. *Self Organizing Maps*. Springer Verlag, 1995.

[17] A. Mead. Review of the development of multidimensional scaling methods. *The Statistician, Vol. 33, p. 27-35*, 1992.

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

[19] J. Siegel, E. Farrell, R. Goldwyn, and H. Friedman. The surgical implication of physiologic patterns in myocardial infarction shock. *Surgery Vol. 72, p. 126-41*, 1972.

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

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

[22] J. A. Wise. The ecological approach to text visualization. *JASIS, Vol. 50, No. 13, p. 1224-1233*, 1999.

[23] 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. of Information Visualization '1995, p. 51-58*, 1995.

[24] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical displays: A general framework for visualization and exploration of large multivariate data sets. *Computer & Graphics, to appear*, 2002.

[25] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. *Submitted to InfoVis '02*, 2002.

[26] J. York, S. Bohn, K. Pennock, and D. Lantrip. Clustering and dimensionality reduction in spire. *Proc. of the Symposium on Advanced Intelligence Processing and Analysis, p. 73*, 1995.