Interactive Hierarchical Dimension Ordering, Spacing and Filtering for Exploration of High Dimensional Datasets

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

Abstract

Large numbers of dimensions not only cause clutter in multidimensional visualizations, but also make it difficult for users to navigate the data space. Effective dimension management, such as dimension ordering, spacing and filtering, is critical for visual exploration of such datasets. Dimension ordering and spacing explicitly reveal dimension relationships in arrangement-sensitive multidimensional visualization techniques, such as Parallel Coordinates, Star Glyphs, and Pixel-Oriented techniques. They facilitate the visual discovery of patterns within the data. Dimension filtering hides some of the dimensions to reduce clutter while preserving the major information of the dataset.

In this paper, we propose an interactive hierarchical dimension ordering, spacing and filtering approach, called DOSFA. DOSFA is based on dimension hierarchies derived from similarities among dimensions. It is a scalable multi-resolution approach making dimensional management a tractable task. On the one hand, it automatically generates default settings for dimension ordering, spacing and filtering. On the other hand, it allows users to efficiently control all aspects of this dimension management process via visual interaction tools for dimension hierarchy manipulation. A case study visualizing a dataset containing over 200 dimensions reveals how our proposed approach greatly improves the effectiveness of high dimensional visualization techniques.

Keywords: dimension ordering, dimension spacing, dimension filtering, multidimensional visualization, high dimensional datasets

1 Introduction

High dimensional datasets are becoming commonplace in applications such as digital libraries, bioinformatics, simulations, process monitoring, and surveys. They bring an important issue to existing multi-dimensional visualization techniques - dimension management. Without effective dimension management, such as dimension ordering, spacing and filtering, high dimensional visualizations can be cluttered and difficult for users to navigate the data space. For example, a 200 dimensional data set means 40000 plots for Scatterplot Matrices [7], 200 axes for Parallel Coordinates [13, 22], 200 subwindows for Recursive Pattern [15], and 200 arms for Star Glyphs [19].

The order of dimensions is crucial for the effectiveness of a large number of visualization techniques [2]. For example, in many multidimensional visualization techniques, such as Parallel Coordinates [13, 22], Star Glyphs [19], Circle Segments [3] and Recursive Pattern [15], the dimensions have to be positioned in some one- or two- dimensional arrangement on the screen. This chosen arrangement of dimensions can have a major impact on the expressiveness of the visualization because the relationships among adjacent dimensions are easier to detect than relationships among dimensions positioned far from each other. Another example is attribute mapping. In visualizations such as Chernoff Faces [6], Worlds within Worlds [9], and Dimensional Stacking [17], more important dimensions need to be mapped to more preattentive visual attributes, such as more important features of the face in Chernoff Faces, axes appearing earlier in Worlds within Worlds, and outer dimensions in Dimensional Stacking. Dimension ordering targets these problems, and aims to improve the effectiveness of the visualization by giving reasonable orders to the dimensions so that users can easily detect relationships or pay more attention to more important dimensions.

In several multidimensional visualization techniques, such as Parallel Coordinates [13, 22] and Star Glyphs [19], uniform spacing/angles are placed by default between two adjacent axes in the display. We conjecture that non-uniform spacing could be used to convey information about dimensions, such as similarity between adjacent dimensions or structure of the dimension space.

Dimension filtering removes some of the dimensions from the display. It is essential for visualizing high dimensional datasets. For datasets contains several hundreds or more dimensions, none of the existing multidimensional visualization techniques can map all the dimensions at the same time without cluttering the display. Popular dimension reduction approaches, such as Principal Component Analysis [14], Multidimensional Scaling [18], and Kohonen's Self Organizing Maps [16, 10], condense the hundreds or thousands of dimensions into a few dimensions. However, those generated dimensions have little intuitive meaning to users and allow little user interaction. Dimension filtering is more intuitive to users in that the remaining dimensions are all original dimensions in the dataset so that they are all meaningful. Dimension filtering is also more flexible in that it allows users to interactively select or unselect dimensions to be filtered.

In this paper, we propose a general approach to dimension management for high dimensional visualization. Our solution is an interactive hierarchical dimension management approach called DOSFA (Dimension Ordering, Spacing, and Filtering Approach). It supports both automatic as well as interactive means of dimension ordering, spacing and filtering.

To reveal how DOSFA improves the effectiveness of high dimensional visualization techniques, we present a case study in this paper that visualizes the OHSUMED dataset, which contains the word-counts of a medical abstract collection [12]. It contains 215

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

dimensions and 298 data points.

The remainder of this paper is organized as follows. In Section 2, we review related work. In Section 3, we introduce the dimension hierarchy construction and navigation processes fundamental to our DOSFA approach. In Sections 4, 5 and 6 we present our semi-automatic dimension ordering, spacing and filtering approaches. In Section 7, we present our conclusions and future work.

2 Background

Dimension order is an important issue in visualization. Bertin gave some examples illustrating that permutations of dimensions and data items reveal patterns and improve the comprehension of visualizations [4]. Ankerst et. al. [2] pointed out the importance of dimension arrangement for order-sensitive multidimensional visualization techniques. They defined the concept of similarity between dimensions and discussed several similarity measures. They proposed the idea of rearranging the dimensions such that dimensions showing a similar behavior are positioned close to one another. They proved that this is an NP-complete problem equivalent to the traveling salesman problem, and used an automatic heuristic approach to generate a solution. Our dimension ordering approach is different from their approach in that we impose a hierarchical structure over the dimensions to reduce the complexity of the ordering problem and allow efficient user interactions.

Dimension order is also important for many other fields. For instance, the database primitive similarity join has been used to speed up applications such as similarity search, data analysis and data mining. Its computational overhead is mostly dedicated to the final distance computations between the feature dimensions. Böhm et. al.[5] proposed a generic technique to avoid and accelerate the distance calculations between those dimensions by carefully ordering them according to a probability model.

Manual dimension ordering and filtering are available in many multidimensional visualization systems. For example, Polaris [20] allows users to manually select and order the dimensions to be mapped to the display. Microsoft Excel allows users to change the order of the dimension by drag-and-drop operations. In XmdvTool [1], users manually filter dimensions and change the order of the dimensions from a reconfigurable list of dimensions. Although manual dimension ordering and filtering as found in such tools are sufficient for low dimensional datasets, they become cumbersome for high dimensional datasets.

Conveying information using spacing in the display has many applications. Many tree drawing algorithms [11, 8] use spacing to convey structural information about the tree. Ward [21] used the distance between adjacent glyphs to convey their relationship in an N dimensional space. In our approach, we propose to use the spacing between dimensions to indicate similarity between adjacent dimensions or structure of the dimension space.

The idea of using dimension hierarchies to facilitate dimension ordering, spacing and filtering is inspired by our previous work on Visual Hierarchy Dimension Reduction (VHDR) [24]. In VHDR, dimensions in a high dimensional dataset are grouped into a dimension hierarchy according to their similarities. Users select interesting dimension clusters and display them instead of the original dimensions in order to reduce the dimensionality of the display.

InterRing [23] (Figure 1 (a)-(d)) is an interactive radial space filling tree visualization tool we designed for visualizing a dimension hierarchy. InterRing is generated using the rules that deeper nodes of the hierarchy are drawn further from the center; child nodes are drawn within the arc subtended by their parents, and the sweep angle of a non-leaf node is equal to the aggregation of that of all its children. InterRing provides a rich set of interactive tools for panning/zooming, rolling-up/drilling-down, multi-focus distortion, modification, reordering, and selection.

3 Dimension Hierarchy

Our hierarchical dimension ordering, spacing and filtering approach is based on dimension hierarchies derived from similarities among dimensions. The problem of determining the similarity of dimensions was characterized by Ankerst et. al. [2] as follows: The database containing N data items with d-dimensions can be described as d arrays $A_i(0 <= i < d)$, each containing N real numbers $a_{i,k}$ $(0 <= k < N)$. A similarity measure S maps two such arrays to a real number. It also might be called a dissimilarity measure because large numbers mean high dissimilarity whereas zero means identity. Similarity can be defined in various ways. Often it is highly domain-dependent. Detailed information of similarity measures between two dimensions can be found in [2]. In this paper, we assume that $0 <= S <= 1$.

Currently, we use an efficient counting test to decide if the similarity measure between two dimensions is lower than a certain similarity measure threshold. Given a similarity measure threshold S, the idea of our approach is that if most data items in the dataset have lower or equal dissimilarities than S when evaluating two dimensions, then the similarity measure of these two dimensions passes the counting test and is lower than S. Those data items that have higher dissimilarities than S are defined as outliers. Given an acceptable outlier percentage O, the number of outliers must be lower than $O * N$ to pass the counting test. This approach is flexible in that users can change the outlier percentage in order to constrain or relax the similarity measure.

Using a dimension clustering approach (described in Section 3.1), we group similar dimensions into clusters and similar clusters into larger clusters, resulting in a dimension hierarchy. By organizing the original dimensions of a high dimensional datasets into a dimension hierarchy, we are able to switch the problem of ordering, spacing, and filtering of all the original dimensions to the problem of ordering, spacing, and filtering of the child nodes of each cluster in the dimension hierarchy. This then scales down the problems and reducing their complexity.

3.1 Dimension Hierarchy Construction

We construct dimension hierarchies using a dimension clustering approach. Generally speaking, as long as the similarity measure between two dimensions is defined, any one of several existing data clustering algorithms can be used to generate the dimension hierarchy; the similarity measure between two dimensions in dimension clustering corresponds to distance between two data items and dimensions in dimension clustering correspond to data items in data clustering. We briefly describe a bottom-up agglomerative dimension clustering algorithm we implemented as follows. For more detail information, refer to [24].

- **Iterative Clustering**: We use a bottom-up clustering approach with a user defined number of iterations I. The iterations are performed in the order of $iteration_0$, $iteration_1$, ..., $iteration_I$. The iterations correspond to a series of increasing similarity measure thresholds S_i $(0 <= i < I$, $S_i < S_j$ if $i < j$, $S_0 = 0$, $S_I = 1)$. These thresholds are

the minimum similarity measure required among the dimensions in a cluster formed during the iterations. In $iteration_i$, dimensions that have not formed any clusters and clusters formed from the previous iterations are considered. If any pair of them has a similarity measure s smaller than S_i, the pair is recorded as a *similar pair*. Then the dimension or cluster contained in the largest number of similar pairs is extracted as a new cluster center. All the other dimensions and clusters in similar pairs involved with it are put into this new cluster and the similar pairs are removed. Repeating the above approach will form more new clusters. An iteration ends when no similar pairs are left. It is ensured that all dimensions can be included into a root cluster since $S_I = 1$.

- **Representative Dimensions**: In order to calculate the similarity measure between two dimension clusters or a dimension cluster and a dimension, we use a representative dimension for each dimension cluster. The data array of the representative dimension is the average of the arrays of dimensions included in this dimension cluster. For a dimension cluster containing non-leaf nodes, the average can be calculated using the representative dimensions of the non-leaf nodes scaled by the number of dimensions included in the non-leaf nodes.

- **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. We select all data clusters with extents much smaller than the minimum similarity measure S_0. These data clusters contain all data items in the data set exactly once. We use these data clusters instead of the original data items in the data set to calculate the similarities among dimensions. During our counting tests for similarity measures, the entry of a data cluster is added into the count if it meets the criteria. For a very large data set, the number of data clusters used in this process would be much smaller than the number of original data items.

3.2 Dimension Hierarchy Navigation and Modification

Because dimension hierarchies are essential for our hierarchical dimension ordering, spacing and filtering approach, it is important to allow users to interactively investigate and modify the automatically generated dimension hierarchy. We use InterRing [23] to navigate and modify the dimension hierarchy. InterRing (Figure 1 (a)-(d)) provides a rich set of navigation operations to allow users to interactively gain overview and detail of the dimension hierarchy. These operations include:

drill-down/roll-up: the process of exposing/hiding sub-branches of the hierarchy;

pan, zoom, and rotation: the process of modifying the focus, scale, and orientation of the display.

distortion: the process of enlarging some objects in the display while maintaining the context of surrounding objects;

With the modification operation in InterRing, users are able to modify the given dimension hierarchy according to their domain knowledge. A simple drag-and-drop operation allows nodes or subtrees to be relocated to arbitrary non-terminal nodes in the hierarchy.

Figure 1 (a) shows the dimension hierarchy of the OHSUMED dataset visualized in InterRing. Figure 1 (b) is the same hierarchy after reordering (Section 4). All the leaf nodes are selected and their dimension names are shown. In Figure 1 (c), the reordered hierarchy is distorted in order to examine details of several nodes. In Figure 1 (d), the nodes outside the focus are hidden using a roll-up operation so that the focus node can be viewed more clearly.

4 Dimension Ordering

Different orders of dimensions in order-sensitive multidimensional visualizations can reveal different aspects of the datasets to users. For example, similarity-oriented order of the dimensions places dimensions with similar patterns next to each other. Through this order, users are able to detect interdependent dimensions. As another example, importance-oriented order of dimensions places dimensions that are more important to the users in more prevalent visualization positions, or maps them to more pre-attentive visual attributes, thus helping users concentrate on them. In the following subsections, we discuss these two dimension ordering techniques.

4.1 Similarity-Oriented Dimension Ordering

Similarity-oriented dimension ordering aims to minimize the sum of similarities of all adjacent dimensions in the display. It has been shown to be an NP-complete problem by Ankerst et. al. [2]. In our approach, we reduce the complexity of this problem using the dimension hierarchy. First, we order each cluster in the dimension hierarchy. For a non-leaf node, we use its representative dimension in the ordering of its parent node. Then, the order of the dimensions is decided in a depth-first traversal of the dimension hierarchy (non-leaf nodes are not counted in the order). Once the order of children of each cluster is decided, the order of all the dimensions is decided. Thus the problem of ordering all the original dimensions has be re-expressed as ordering the children of non-leaf nodes of the dimension hierarchy.

Because in most cases, the number of children of a cluster in the dimension hierarchy is much smaller than the number of leaf nodes of that hierarchy, the complexity of the ordering will be greatly reduced. For example, suppose we have N dimensions to be arranged in a 1-dimensional order, if we use optimal ordering, which require the calculation of all possible permutations of the dimensions, the similarity measure calculation, which is the most time consuming in the ordering approach, has to be applied N! times. While using a well-balanced M-tree, where M is much smaller than N, we only need to apply the similarity measure calculation for approximately $(N-1)/(M-1)*M!$ times, which is much less than N!. As a result, some algorithms that are not suitable for ordering a large number of dimensions, such as optimal ordering, could be used in our approach. Here are some ordering algorithms we implemented:

- **Optimal ordering**. Optimal ordering computes the sum of neighboring dimension similarities for every possible permutation within the dimensions to be ordered to find the permutation with the smallest sum.

- **Random Swapping**. Random swapping starts with an initial configuration and randomly chooses two dimensions to switch their positions. If the new arrangement has a smaller similarity sum, then it is kept and the old one is rejected; otherwise leave the old arrangement intact and go on swapping

another pair of dimensions. Keep doing this for a certain number of times. This is not an optimal approach for reordering dimensions, but it is more applicable for large number of dimensions.

An approximate dimension ordering approach is to directly use the depth-first traversal result of a dimension hierarchy without reordering it. The reason is that in dimension hierarchies, similarities among siblings within a cluster are controlled within certain ranges. In other words, the similarities among the children of a dimension cluster are similar. Thus with a good dimension hierarchy, we should get a reasonable order of the original dimensions even without reordering the hierarchy.

Figures 2 (a), 3 (a) show the OHSUMED dataset in Parallel Coordinates and Star Glyphs (a subset of the data items are shown) in their original dimension order. It can be seen that it is very hard to find any patterns of the dataset. Figure 1 (b) shows the dimension hierarchy of the OHSUMED dataset ordered according to similarity. Figures 2 (b) and 3 (b) show its corresponding Parallel Coordinates and Star Glyphs (the same subset of data items as in Figure 3 (a)) displays. It is obvious that Figures 2 (b) and 3 (b) reveal similar dimensions while Figures 2 (a) and 3 (a) do not. The benefits of dimension ordering are not limited to revealing similar dimensions. It is clear that in Figure 2 (b), it is much easier to detect the overall trends of the dataset than in Figure 2 (a). For the Star Glyphs displays, it is much easier to find differences among the glyphs in Figure 3 (b) than in Figure 3 (a). For example, it is much easier to detect the difference between the third and fourth star glyphs in the last row in Figure 3 (b) than in 3 (a).

4.2 Importance-Oriented Dimension Ordering

The importance of each dimension is decided by a user's particular visualization task. In the following description, we assume that users are looking for variance of a dataset. Thus a dimension that contributes more to the variance of the dataset is more important than a dimension that contributes less to the variance. To this end we order the dimensions according to their contribution to the variance. Here again, we use the hierarchical approach to scale down the complexity of the problem.

We assume that similar dimensions have similar contributions to the dataset's variance because they contain similar information. Thus the problem of ordering all the dimensions can be switched to the problem of ordering each cluster in the dimension hierarchy. As in the similarity-oriented ordering, we first order each cluster in the dimension hierarchy. For a non-leaf node, we use its representative dimension in the ordering of its parent node. Then, the order of the dimensions is decided by the order of the dimensions in a depth-first traversal of the dimension hierarchy (non-leaf nodes are not counted in the order). To order a non-leaf node, we apply Principal Component Analysis (PCA) [14] on its children, and order them according to their contributions to the first several principal components.

With different user tasks, the importance of the dimensions can be decided in different ways from the above example. However, the general approach is similar; either order all the dimensions at the same time according to their importance, or order the clusters in the dimension hierarchy according to the importance of their children in order to reduce the complexity of the problem.

4.3 User Interaction

Relationships among dimensions may remain undetected by the automatic ordering approach. Since users are often experts on the data sets been visualized, they may be able to improve on the results of the automated ordering. Hence allowing users to interactively adjust the order of the dimensions is important.

In our approach, users have two ways to interactively change the order of the dimensions (reorder):

- manually change the order of the dimensions in the data displays. This is similar to many existing manual dimension reordering approaches, using either reconfigurable lists or drag-and-drop operations [20, 1].

- manually change the order of siblings using the "Modification" operation of InterRing. By changing the order of a dimension cluster, the order of all the similar dimensions contained in this cluster is changed correspondingly. This change will be propagated to the data display. This reordering approach is efficient even for a large number of dimensions. It provides the users with a manageable way for manual reordering through this multi-resolution approach.

5 Dimension Spacing

Dimension ordering reveals useful dimension relationship information to users. However, this information may not be accurate. For example, in the dimension order generated using similarity-oriented hierarchical ordering approach, two adjacent dimensions could be the ending dimension of one cluster and the starting dimension of another cluster. Thus their similarity is lower than the similarity of two adjacent dimensions within any of those clusters. For an order generated using a non-hierarchical approach, the similarities between dimensions may still be different. Users could not know these differences from the order of the dimensions. Thus dimension spacing, which explicitly convey dimension relationship information by varying the spacing between two adjacent axes or angles, is useful here. For unordered dimensions, dimension spacing is even more useful since the order of dimensions does not convey any dimension relationship information. In this section, we propose several promising dimension spacing approaches.

5.1 Automatic Approach

In multidimensional visualization techniques containing explicit axes, the default spacing between all the adjacent axes is equal, such as Parallel Coordinates [13, 22] and Star Glyphs [19]. However, the relationships between adjacent dimensions are generally not equal - some adjacent dimensions may have close relationships while others may not. We can explicitly convey this to users by varying the spacing between adjacent axes - a smaller distance or angle means a closer relationship so that closely related dimensions are placed closely together, while unrelated dimensions stand apart. By applying this spacing approach to the similarity-oriented ordered dimensions, we allow users to grasp relationships between the dimensions more intuitively as suggested by the Gestalt Law on proximity. A simple and general approach to spacing dimensions to reveal dimension relationships is to calculate the correlation factor of each pair of adjacent dimensions, and assign axes a distance or angle proportional to this factor.

Besides revealing relationships between adjacent dimensions, spacing can also be used to reveal the structure of the dimension space in our hierarchical approach. The algorithm of spacing according to the structure of the dimension hierarchies is simply to make the distance between two adjacent dimensions proportional

to the threshold used to form their first common ascendant. Thus a dimension will have a smaller distance or angle to its adjacent dimension if the adjacent dimension belongs to the same cluster than if it belongs to a different cluster. Also, if two adjacent dimensions belong to the same cluster, their distance or angle is smaller if the threshold used to form that cluster is smaller. Because the threshold used to form a cluster reflects similarity of its children in the dimension hierarchy, the spacing calculated using this algorithm also reveals the similarity of the adjacent dimensions along with the overall structure of the dimension space.

The Parallel Coordinates display of the OHSUMED dataset in Figure 2 (b) has been spaced according to the structure of the dimension hierarchy. It is easy to detect some closely related dimensions, and some dimensions that are fairly different from adjacent dimensions from this figure. Figure 2 (c) is a zoomed display of Figure 2 (b). It can be seen that in the OHSUMED dataset, which contains word-counts in a medical abstract collection, the word "glucose" appears in similar frequency to the word "insulin" in most articles.

5.2 Interactive Control

For high dimensional datasets, the displays might still be cluttered despite dimension spacing. Thus it is important to allow users to interactively enlarge or decrease the distance of adjacent dimensions so that they can examine as well as hide detail of interesting or uninteresting dimensions and relationships. In our approach, we provide three ways to allow users to interactively change the distances between dimensions.

- zooming in/out and panning. We allow users to zoom the x and/or y directions so that, for visualization techniques such as Parallel Coordinates, horizontal zooming will only affect the distances between axes.
- manual distortion. When the display is in distortion mode, users can left click between two axes to increase their spacing, or right click to decrease it.
- structure-based spacing distortion.

Zooming and panning keeps the original spacing of the dimensions, however, context is lost when users are examining details. Manual distortion preserves the context, however, the local spacing in the distorted area is lost. In addition, it can be tedious when users want to examine details of many dimensions at the same time. When the dimensionality of the dataset is high, it is also difficult to specify a distortion because the spacing between most dimensions is very small. *Structure-based spacing distortion* addresses these shortcomings.

Structure-based spacing distortion is linked to structure-based circular distortion of InterRing [23], the dimension hierarchy visualization. Structure-based circular distortion allows users to proportionally enlarge/decrease all descendants in clusters through simple drag-and-drop operations. Figure 1 (c) shows a distorted dimension hierarchy in InterRing. A cluster and all its descendants are enlarged by one drag-and-drop operation. We propagate this distortion to dimension spacing in data displays. Thus when a node in InterRing presenting a dimension is enlarged/decreased, the spacing around the dimension is also enlarged/decreased.

Particularly, we adjust the dimension spacing algorithm so that the spacing between two adjacent dimensions is decided by the product of two parameters: the spacing parameter and the distortion parameter. The spacing parameter is the distance or angle decided by the algorithm described in Section 5.1 that reflects the relationship between these two dimensions. The distortion parameter is specified by the distortion degree of the nodes representing these two dimensions in InterRing. We choose the larger of the two so that for an enlarged leaf node in InterRing, the spacing around its corresponding dimension is also enlarged proportionally.

Many advantages of structure-based distortion in InterRing have propagated to structure-based spacing distortion, such as:

- Spacing of a cluster of dimensions can be distorted using a single drag-and-drop operation.
- Details can be examined within their context.
- Local spacing inside the enlarged or decreased area has been preserved.
- Multiple foci can coexist. Users can enlarge or decrease several areas in the same display.

In Figure 1 (c), the dimension hierarchy is distorted using structure-based distortion. Figure 3 (c) shows part of the Star Glyphs display linked to Figure 1 (c) with structure-based spacing distortion. It is clear that structure-based spacing distortion helps users to see details within context.

6 Dimension Filtering

6.1 Automatic Approach

When the dimensionality is fairly large, even if we apply ordering and spacing of the dimensions to the display, it still may be crowded. Very similar dimensions may likely be cluttered together. In this case, we would like to filter some dimensions to reduce the clutter problem while at the same time retaining most of information in the dataset. This could be done manually, automatically, or semi-automatically. We propose a dimension filtering approach that automatically generates a default filtering result, while allowing users to interactively modify it. We claim that automatically generating a default result is important for high dimensional datasets, and user interactions are also necessary because improvements can usually be made on automatic approaches.

Our dimension filtering approach is also based on the dimension hierarchy. In this approach, we use a filtering criterion that is a combination of the dimension similarity and importance. We assume that if some dimensions are very similar to each other, then only one of them should be left in the display. We also assume that if some dimensions are fairly unimportant for a user's visualization task, then they should not be displayed. Thus a similarity threshold and an importance threshold are used in the filtering. The algorithm to select dimensions for display is an iterative approach starting from the root of the dimension hierarchy. Each iteration contains the following steps:

1. Check if the node's importance is smaller than the importance threshold; if yes, ignore the node and return. That is, the dimensions contained in unimportant nodes are ignored.
2. If it is not ignored and it is a leaf node, select it and return.
3. Otherwise check the threshold used to form the node. If it is larger than the similarity threshold, apply the iterative approach on its immediate descendants. Otherwise only apply the iterative approach on its most important immediate descendant.

Figures 2 (d), 3 (d) and 4 (b) give examples of dimension filtering. Comparing Figures 1 (b), 4 (b) with 2 (d), 3 (d) (remaining dimensions in Figures 2 (d) and 3 (d) keep the same order as in Figures 2 (b) and 3 (b)), we find that the filtering retains the major information of the dataset fairly well, while in the filtered displays, the number of dimensions is much more manageable compared to the unfiltered ones. In Figure 4 (b), the Scatterplot Matrix display of the OHSUMED dataset, there are so many plots that individual plots cannot be discerned without significant zooming.. While in Figure 4 (d), the number of plots has been greatly reduced.

6.2 Interactive Filtering

It is possible that the automatic filtering process might filter out some dimensions that a user is concerned about, or keep some dimensions that a user finds uninteresting. It is therefore important to allow users to interactively adjust filtering results.

The automatic filtering (Section 6.1) is a recursive process staring from the root of a hierarchy. We can apply this process to the root of any sub-branch in the dimension hierarchy instead of the root of the dimension hierarchy. In this way, we can apply filtering to a subset of the dimensions instead of all the dimensions.

In InterRing, each selected leaf node corresponds to a dimension displayed in the data visualization. We can add/delete a dimension to/from the display by selecting/unselecting its corresponding leaf node. Thus users can filter dimensions using a manual selection operation in InterRing, that is, clicking an unselected node to select it, or clicking a selected node to unselect it.

In InterRing, rolling-up/drilling-down operations allow users to hide/show sub-branches by clicking their root nodes. We also link them to the dimension filtering. Whenever some leaf nodes are hidden in InterRing, it means that users are not interested in them. Thus we also filter their corresponding dimensions from the data display.

Users can also manually delete a dimension from the data display by clicking to select it, then hitting the 'delete' button to delete it. Since InterRing provides an overview of the dimension space structure, while data display provides the context of the data items for manually filtering dimensions, the option of filtering dimensions from InterRing or the data display gives users significant flexibility.

7 Conclusion and Future Work

In this paper, we have proposed an interactive approach to dimension ordering, spacing and filtering for high dimensional datasets based on dimension hierarchies. Dimension ordering, spacing and filtering can significantly increase the effectiveness of multidimensional visualizations, but the processes are complex for high dimensional datasets. By grouping dimensions into a dimension hierarchy according to their similarity, we improved the manageability of dimensions in high dimensional data sets and reduced the complexity of the ordering, spacing and filtering tasks. In addition, our findings are that user interactions for dimension ordering, spacing and filtering are much easier to accomplish with dimension hierarchies.

In the future, we will formally evaluate our approach. We plan to compare the quality of our hierarchical ordering approach with other heuristic approaches. We want to evaluate how dimension ordering, spacing and filtering benefits high dimensional visualization applications, such as document visualization, through user studies. We intend to evaluate DOSFA's applicability to different multidimensional visualization techniques. We also want to study and improve the efficiency and effectiveness of our dimensional clustering algorithm, as it is the foundation of our DOSFA approach.

References

[1] http://davis.wpi.edu/ xmdv/.

[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] M. Ankerst, D. Keim, and H. Driegel. Circle segments: A technique for visually exploring large multidimensional data sets. *Proc. of Visualization '96*, 1996.

[4] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.

[5] C. Boehm, F. Krebs, and H.-P. Kriegel. Optimal dimension order: A generic technique for the similarity join. *4th Int. Conf. on Data Warehousing and Knowledge Discovery, pp. 135-149*, 2002.

[6] 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.

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

[8] P. Eades. Drawing the trees. *Bulletin of the Institute of Combinatorics and its Applications, p. 10-36*, 1992.

[9] S. Feiner and C. Beshers. Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. *Proc. UIST'90, p. 76-83*, 1990.

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

[11] G. Furnas. Generalized fisheye views. *Proc. of Computer-Human Interaction '86, p. 16-23*, 1986.

[12] W. Hersh, C. Buckley, T. Leone, and D. Hickman. OHSUMED: An interactive retrieval evaluation and new large text collection for research. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Performance Evaluation, pages 192–201, 1994.

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

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

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

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

[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] C. Stolte and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *InfoVis '00, p. 5-14*, 2000.

[21] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization, Vol 1, pp.194-210*, 2002.

[22] E. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association, Vol. 411(85), p. 664-675*, 1990.

[23] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. *InfoVis '02, p. 77-84*, 2002.

[24] J. Yang, M. O. Ward, E. A. Rundensteiner, and S. Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. *VisSym 2003, accepted*, 2003.

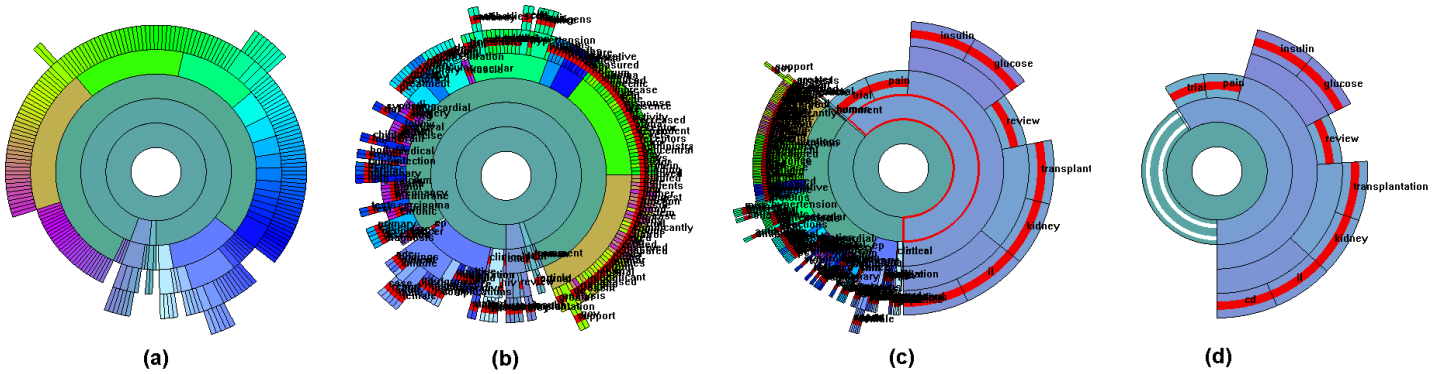

Figure 1: InterRing. (a): Dimension hierarchy of OHSUMED dataset in InterRing. (b): after reordering. (c): after distortion. (d): after roll-up operation.

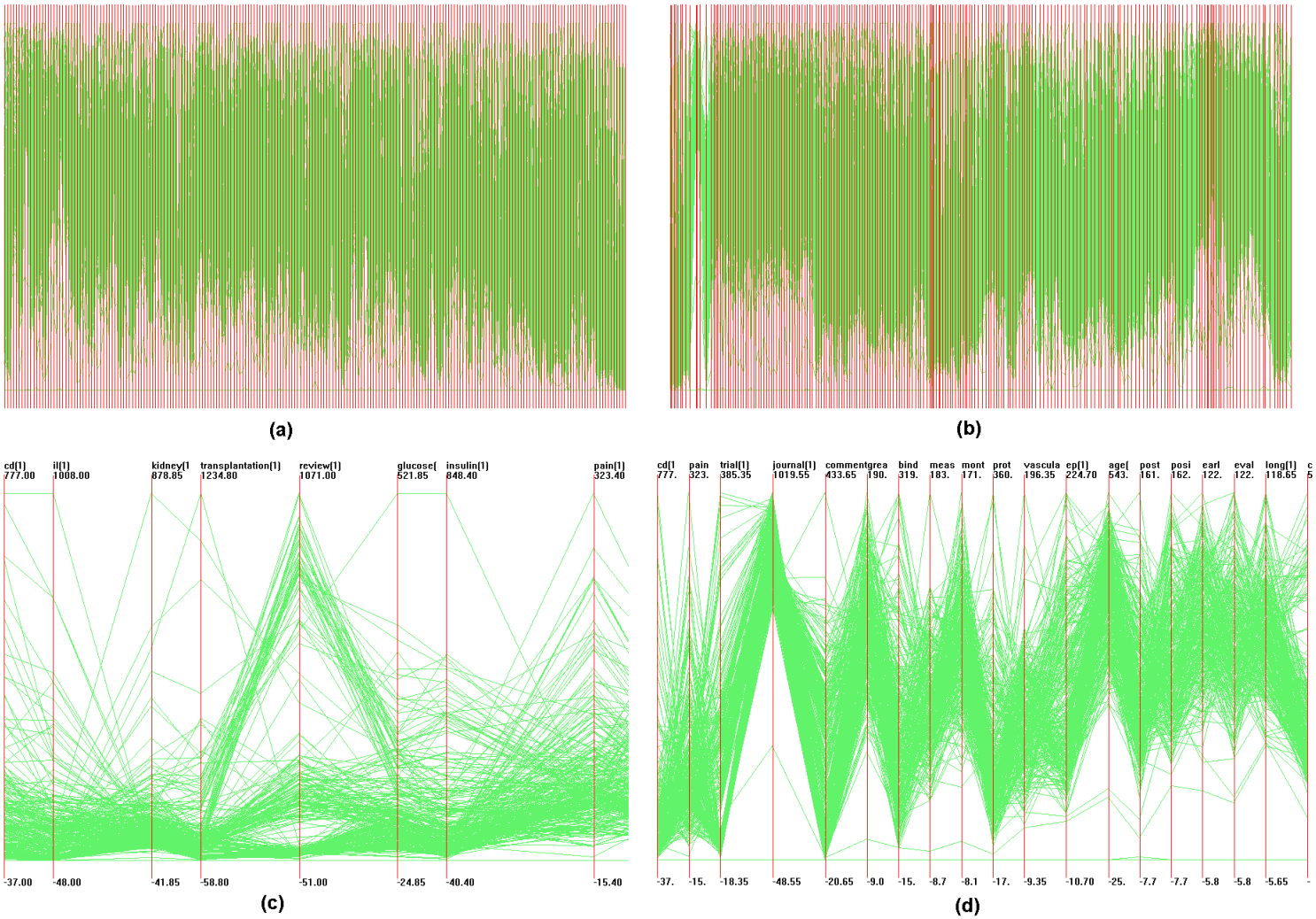

Figure 2: Parallel Coordinates. (a): OHSUMED dataset without DOSFA. (b): after ordering and spacing. (c): after zooming. (d): after filtering.

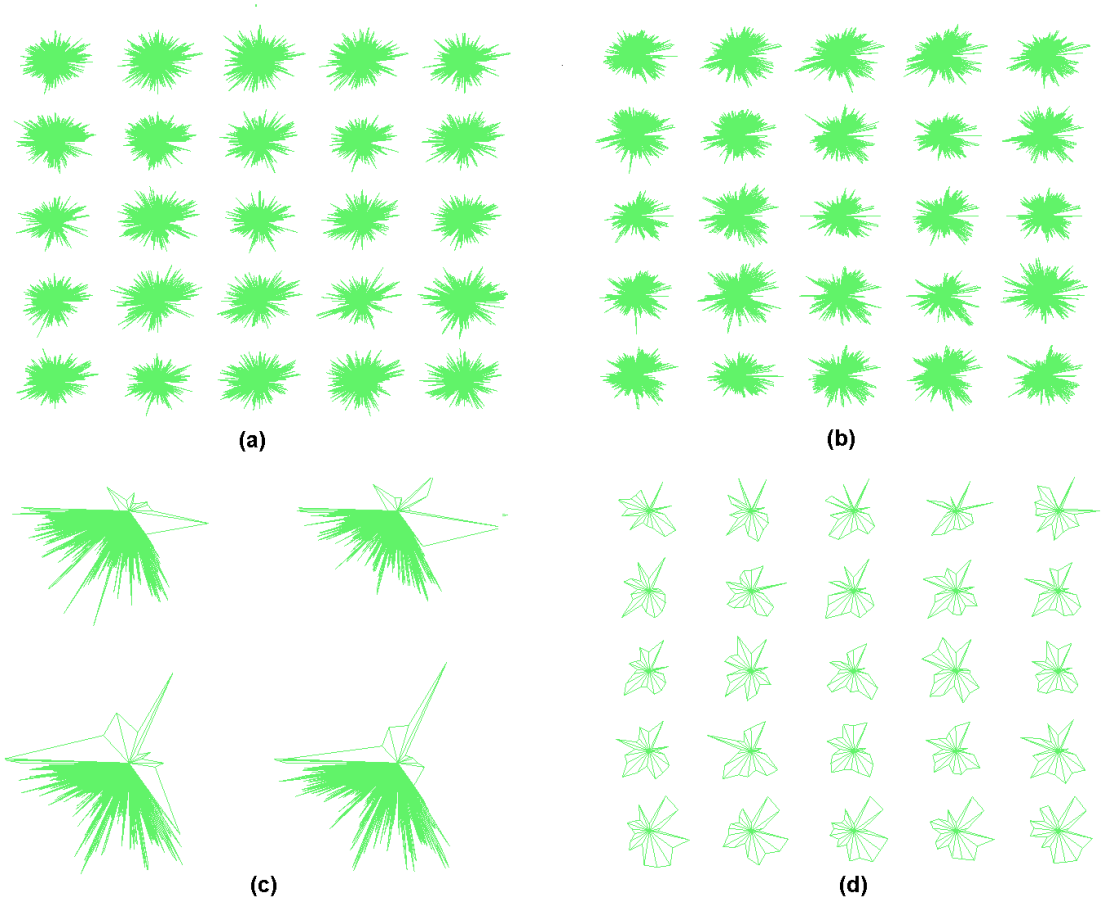

Figure 3: Star Glyphs. (a): OHSUMED dataset without DOSFA. (b): after ordering and spacing. (c): distorted star glyphs. (d): after filtering.

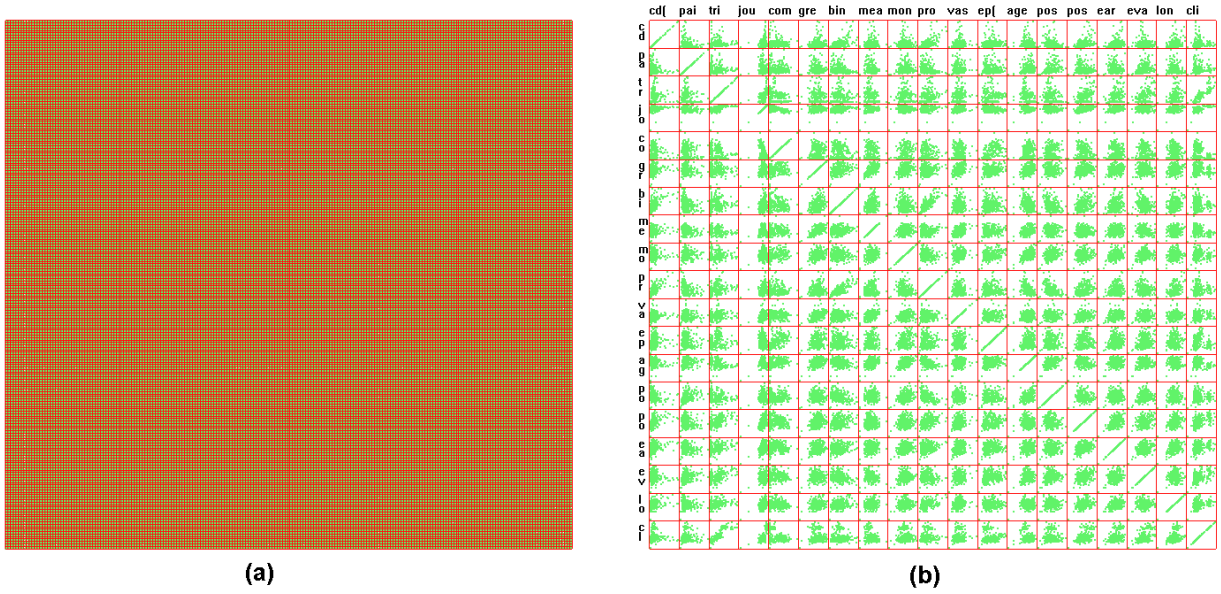

Figure 4: Scatterplot Matrices. (a): OHSUMED dataset without DOSFA. Individual plots cannot be discerned without significant zooming. (b): after filtering.