Arrangement of Low-dimenional Parallel Coordinate Plots for High-dimensional Data Visualization

Haruka Suematsu*[1], Zheng Yunzhu*[1], Takayuki Itoh*[1]
Ryohei Fujimaki*[2], Satoshi Morinaga*[3], Yoshinobu Kawahara*[4]
(*1)Ochanomizu University, (*2)NEC Laboratories America, (*3)NEC, (*4)Osaka University
{haruka, yunzhu, itot}@itolab.is.ocha.ac.jp, rfujimaki@sv.nec-labs.com
morinaga@cw.jp.nec.com, kawahara@ar.sanken.osaka-u.ac.jp

Abstract

Multi-dimensional data visualization is an important research topic that has been receiving increasing attention. Several techniques that use parallel coordinate plots have been proposed to represent all dimensions of data in a single display space. In addition, several other techniques that apply scatterplot matrices have been proposed to represent multi-dimensional data as a collection of low-dimensional data visualization spaces. Typically, when using the latter approach it is easier to understand relations among particular dimensions, but it is often difficult to observe relations between dimensions separated into different visualization spaces. This paper presents a framework for displaying an arrangement of low-dimensional data visualization spaces that are generated from high-dimensional datasets. Our proposed technique first divides the dimensions of the input datasets into groups of lower dimensions based on their correlations or other relationships. If the groups of lower dimensions can be visualized in independent rectangular spaces, our technique packs the set of low-dimensional data visualizations into a single display space. Because our technique places relevant low-dimensions closer together in the display space, it is easier to visually compare relevant sets of low-dimensional data visualizations. In this paper, we describe in detail how we implement our framework using parallel coordinate plots, and present several results demonstrating its effectiveness.

1 Introduction

Multi-dimensional data visualization is an important and active research field. According to survey papers in the field, several techniques have been proposed [18, 6]. The authors of [18] divided the available multi-dimensional data visualization techniques into three categories: two-variate displays, multivariate displays, and animation.

The typical two-variate display technique is the scatterplot. Because historically scatterplots have been widely used and are currently implemented in commercial spreadsheet software packages, they are particularly popular and users are familiar with them. The scatterplot matrix, which consists of multiple adjacent scatterplots, has also been widely used to represent the dimensions of high-dimensional datasets. Each scatterplot in a scatterplot matrix is identified by its row and column index. Even though ordinary users are familiar with scatterplot matrices, they have two major drawbacks. First, if the number of dimensions in a given dataset is very large, the individual scatterplots in a display space may be very small. Second, it is difficult to visually compare arbitrary pairs of scatterplots that are distantly placed in the display space.

Multivariate display techniques attempt to represent the distribution of all the dimensions in a given dataset on a single display space. Several multivariate display techniques are available, including icon- and glyph-based techniques, such as hierarchical axis [13], worlds within worlds [4], and parallel coordinate plots [7]. Recently, the parallel coordinate plots has been the subject of many studies and is widely used. However, this technique has several drawbacks. First, when the number of dimensions is very large it may require a large horizontal display space. Second, it is difficult to represent the correlation of a particular dimension with three or more dimensions.

In this paper, we present a technique to represent high-dimensional spaces using multiple low-dimensional visualization components. Our proposed technique overcomes the drawbacks of both the two-variate and multivariate data visualization techniques. As mentioned above, scatterplot-based techniques represent high-dimensional spaces as collections of scatterplots. However, the drawback of these techniques is that when all pairs of dimensions are displayed equally, the individual scatterplots displayed may be very small. Conversely, our proposed technique selectively displays meaningful sets of low-dimensional visualization components. Our technique first selects a pre-defined number of groups of dimensions based on their correlations. Then, it defines the distances or connectivity among the sets of dimensions. Next, it computes the ideal positions of the sets of dimensions based on their distance or connectivity values. Finally, it applies a rectangle packing algorithm and places the set of low-dimensional visualization components [8] [9] based on their ideal positions. Consequently, our technique places similarly looking low-dimensional visualization components closer together in the display spaces. Conversely, our proposed technique selectively displays meaningful sets of low-dimensional visualization components.

2 Related Work

2.1 Parallel Coordinate Plots

The parallel coordinate plots [7] display high-dimensional datasets as a set of polylines intersecting with parallel axes. Specifically, the existence of parallel polylines between two axes indicates positive correlation. Conversely, if several polylines cross the axes, it is indicative of negative correlation. Improving the parallel coordinate plots has been a very active research topic.

First, when coordinates are parallel, polylines may become cluttered. Several techniques have attempted to improve the readability of the results obtained by the parallel coordinate plots technique by applying clustering techniques or sampling the polylines [5] [10] [15] [20].

Second, the effectiveness of parallel coordinate plots strongly depends on the order of the dimensions. Although, various dimension ordering techniques have been proposed [14] [19], it is still often difficult to represent all correlations in one display space, especially when a particular dimension is strongly correlated with many other dimensions. Actually, parallel coordinate plots represent just a subset of all possible relationships between the dimensions.

Third, if the number of dimensions is very large, the parallel coordinate plots technique may require a display space that is horizontally large. This problem of parallel coordinate plots can be solved if we divide the high-dimensional data space into smaller meaningful subsets. Several works have explored the idea of dividing high-dimensional data spaces into multiple lower-dimensional data spaces and representing them by a set of parallel coordinate plots [1].

The technique presented in this paper addresses the second and third problems presented above.

2.2 Rectangle Packing

Our proposed technique applies a rectangle packing algorithm to visualize hierarchical data [8]. The rectangle packing algorithm represents a hierarchy as nested rectangles and leaf-nodes as painted icons. Moreover, it satisfies the following conditions:

Condition 1: It never overlaps the leaf-nodes and branch-nodes in a single hierarchy of other nodes.

Condition 2: It attempts to minimize the display area.

Condition 3: It attempts to minimize the aspect ratio and area of the rectangular subspaces.

Condition 4: It attempts to minimize the distances between the actual and ideal positions of the rectangular subspaces (when the ideal positions of the rectangular subspaces are provided).

First, the rectangle-packing algorithm specifies the order of the placement of rectangles. Then, it identifies several candidate positions that satisfy condition 1, for placing a rectangle. Next, for each candidate position, it computes penalty values, which represent to what extent each position satisfies conditions 2, 3, and 4, respectively. Finally, it places the rectangle at the best candidate position.

This rectangle packing algorithm has also been applied to a graph visualization technique [9]. The technique first applies hierarchical clustering to a given graph, and generates clusters of nodes based on both their assigned categories and connectivity. Then, it visualizes the hierarchy by applying a hybrid force-directed and space-filling layout. The force-directed layout minimizes distances between connected or similarly categorized nodes. Conversely, the space-filling layout applies the rectangle-packing algorithm to minimize the cluttering of nodes and maximize the utility of the display. Hence, the hybrid layout for graph visualization realizes simultaneously both features.

3 Framework of Low-dimensional Data Visualization Packing

In this section, we define the input datasets and outputs, and briefly describe the processing flow of the low-dimensional data visualization packing approach.

3.1 Data structure

In this paper, an n-dimensional input dataset Ds is defined as follows:

$$Ds = \{x_1, ..., x_N\}, \tag{1}$$

where x_i is the i-th plot, and N is the number of plots, as shown in Figure 1(Left)(1). As shown in Figure 1(Left)(2), our technique first divides the n dimensions into the following groups:

$$Gp = \{g_1, ..., g_G\}, \tag{2}$$

$$g_i = \{d_{i1}, ..., d_{iG_i}\}, \tag{3}$$

where g_i is the i-th group of the dimensions, G is the number of groups, d_{ij} is the j-th dimension of g_i, and G_i is the number of dimensions included in g_i. Figure 1(Left)(3) illustrates how our proposed technique displays a set of low-dimensional datasets Dp_1 to Dp_G, where Dp_i is a G_i-dimensional dataset containing N plots, x_1 to x_N, with a set of dimensions g_i.

3.2 Processing flow

Our proposed technique first divides the high-dimensional input dataset into a set of low-dimensional datasets. Let $S_i = \{x_1^{d_i}, ..., x_N^{d_i}\}$ be a set of scalar values, where $x_j^{d_i}$ is the value of the i-th dimension of the j-th plot. Our technique compares arbitrary pairs of values S_i and S_j, and if they are similar or correlative, categorizes the i-th and j-th dimensions into the same group. Next, it computes the similarities or distances between arbitrary pairs of values of the low-dimensional datasets Dp_i and Dp_j. Finally, it calculates the positions of the low-dimensional datasets and places similar ones closer together in the display space. The complete process is illustrated in Figure 1(Left)(4).

Our technique first calculates the ideal positions of the low-dimensional visualization components, and then applies the rectangle packing algorithm to adjust their positions. This two-step technique satisfies the following requirements: (1) it places similar low-dimensional visualizations closer together in the display space, and (2) it avoids overlaps, thus reducing wasted space in the display regions.

Our current implementation of computing ideal positions supports the following two methods:

- Dimension reduction based on the similarity distances between low-dimensional visualization components.
- Graph layout, where low-dimensional visualization components are connected based on their similarity measures.

In the following two sections, we describe in detail how we implement our technique using parallel coordinate plots.

4 Implementation as a Set of Parallel Coordinate Plots

In this section, we present the details of implementing high-dimensional data visualization by packing a set of parallel coordinate plots. First, we select a group of dimensions to be displayed by parallel coordinate plots. Next, we compute their ideal positions based on their correlations. Finally, we apply the rectangle packing algorithm and adjust their positions.

In the following sub-sections, we describe in detail the processing flow of our implementation.

4.1 Construction of Parallel Coordinate Plots

First, our implementation generates groups of dimensions based on conditional independence, and specifies the order of the dimensions.

4.1.1 Determination of conditional independence

To visualize high-dimensional datasets as groups of parallel coordinate plots, we impose the following requirements on the algorithm used to group dimensions:

- Highly-correlated dimensions are displayed in the same parallel coordinate plots.
- Poorly-correlated dimensions are displayed in different parallel coordinate plots.

This implementation applies ”conditional independence” to properly group the dimensions.

Conditional independence is determined among three disjoint subsets of dimensions [Remark 1] A, B and C (A, B, $C \in \{d_1, \ldots, d_n\}$). Specifically, we compute the mutual information $I(A, B|C)$, where A and B is an arbitrary pair of dimensions, and C is a given dimension. Given C, A and B are conditionally independent if $I(A, B|C)$ is close to zero.

Using the computed conditional independence values between arbitrary pairs of dimensions, we group the dimensions such that conditionally independent dimensions are categorized into different groups [2]. For example, let $\{d_1, d_2, d_3, d_4\}$ be four dimensions, where $I(d_1, d_4|d_3) = 0$, and $I(d_2, d_4|d_3) = 0$. In this case, we can conditionally connect non-independent pairs of dimensions and construct a graph consisting of two creeks, $\{d_1, d_2, d_3\}$ and $\{d_3, d_4\}$. This technique treats creeks as groups of dimensions, and visualizes groups as low-dimensional parallel coordinate plots, as shown in Figure 1(Right)(1).

In general, the problem if identifying the most likely (MLE) structure of conditional independence among variables is NP-complete [11]. However, under a weak assumption (fixed-width of the tree induced by the estimated conditional independence structure), we can use a polynomial-time algorithm based on the sub-modularity in conditional mutual information, which guarantees PAC (probably approximately correct) solutions (instead of MLE) [2].

4.1.2 Dimension reordering for parallel coordinate plots

In our current implementation, we use a greedy algorithm to specify the order of dimensions. Typically, it is preferable to draw strongly correlated dimensions in adjacent parallel coordinate plots. First, we compute the correlations of all possible pairs of dimensions and select the most correlated pair as the two dimensions drawn at the left end of the parallel coordinate plots. This process is repeated to select which dimensions to place to the right of the most recently placed dimensions.

This implementation can be improved by applying the existing dimension reordering algorithms. For example, Zhang et al. presented a dimension reordering algorithm that applies the traveling salesman problem to dimension graphs [19].

4.2 Placement of Parallel Coordinate plots

In the latter part of our proposed technique, we place the set of parallel coordinate plots onto the display space. First, we compute the ideal positions of the parallel coordinate plots on the display space based on their correlations.

4.2.1 Dimension reduction for computing ideal positions

We compute the ideal positions of the parallel coordinate plots by applying a dimension-reduction scheme. Let g_i and g_j be the two groups of dimensions to be displayed as parallel coordinate plots, where $g_i = \{d_{i1}, ..., d_{iG_i}\}$ and G_i is the number of dimensions in g_i.

The similarity distances between the two groups g_i and g_j are computed as follows:

$$D(g_i, g_j) = 1 - \frac{1}{|g_i||g_j| \sum |R_{g_i g_j}} \tag{4}$$

Here, $R_{g_i g_j}$ is the correlation coefficient between g_i and g_j, computed using the following equation:

$$R_{g_i g_j} = \frac{\sum_{a=1}^{Gi} \sum_{b=1}^{Gj} (d_{ia} - \overline{g_i})(d_{jb} - \overline{g_j})}{\sqrt{\sum_{a=1}^{Gi} (d_{ia} - \overline{g_i})^2} \sqrt{\sum_{b=1}^{Gj} (d_{jb} - \overline{g_j})^2}} \tag{5}$$

where G_i and G_j are the number of dimensions in g_i and g_j, respectively, and d_{ia} is the a-th dimension of g_i.

Next, we compute $D(g_i, g_j)$ for all possible pairs of groups of dimension and generate their distance matrix, as shown in Figure 1(Right)(2).

Finally, we apply a dimension reduction scheme (e.g., Isomap) to the distance matrix, and treat the first and second dimensions of each dimension groups as the ideal positions of the parallel coordinate plots on the display space.

4.2.2 Graph layout for computing ideal positions

Instead of generating a distance matrix, when $D(g_i, g_j)$ is sufficiently small we generate a graph structure by connecting pairs of dimension groups g_i and g_j. Next, we apply a force-directed graph layout technique and compute the two dimensional positions of the parallel coordinate plots.

4.2.3 Rectangle packing

After computing the ideal positions of the parallel coordinate plots by selectively applying dimension reduction or graph layout techniques, as shown in Figure 1(Right)(3), we calculate the final positions of the parallel coordinate plots by applying a rectangle packing algorithm, as shown in Figure 1(Right)(4).

5 Example

In this section, we present examples of visualizing datasets using our technique described above. We implemented in Python 2.7 the parallel coordinate plots construction (Section 4.1), and dimension reduction (Sections 4.2.1) algorithms. Moreover, we implemented the force-directed layout (Sections 4.2.2) and rectangle packing (Sections 4.2.3) algorithms using the Java Development Kit (JDK) 1.6.0.

In our experiments, we observed large differences in the quality and computation time between approaches using dimension reduction and force-directed graph layout. Dimension reduction required approximately ten minutes of computation time, while force-directed graph layout required only several seconds. Conversely, the visualization quality using dimension reduction was subjectively much better.

In the next sub-section, we present visualization results obtained using dimension reduction.

5.1 Example 1: Image segmentation data

In this sub-section, we present the visualization results obtained by applying our technique to the ”Segmentation” dataset published at the UCI machine learning repository [21]. This dataset contains 18 feature values for 210 blocks of images, generated by dividing each of 7 images into 30 blocks. We treated images as classes, feature values as dimensions, and blocks as polylines. Hence, the 18-dimensional dataset is displayed as 210 plots in 7 colors.

In Figure 2(Left), we present 20 parallel coordinate plots resulting from our experiment. Here, we draw the values for only two images as yellow and light green polylines. Moreover, in Figure 2(Right), we show a close-up view of 2 parallel coordinate plots in the visualization results. Integer values indicated below the axes of the figures are the IDS of the dimensions of the dataset.

These results indicate that dimension 9 is correlated with many other dimensions, including strong positive correlations with dimensions 10, 12, 16, and weak negative correlations with dimensions 7 and 8. Here, the meanings of the dimension IDs indicated in this example are as follows:

7: Average of the contrast of vertically adjacent pixels. Used for horizontal line detection.

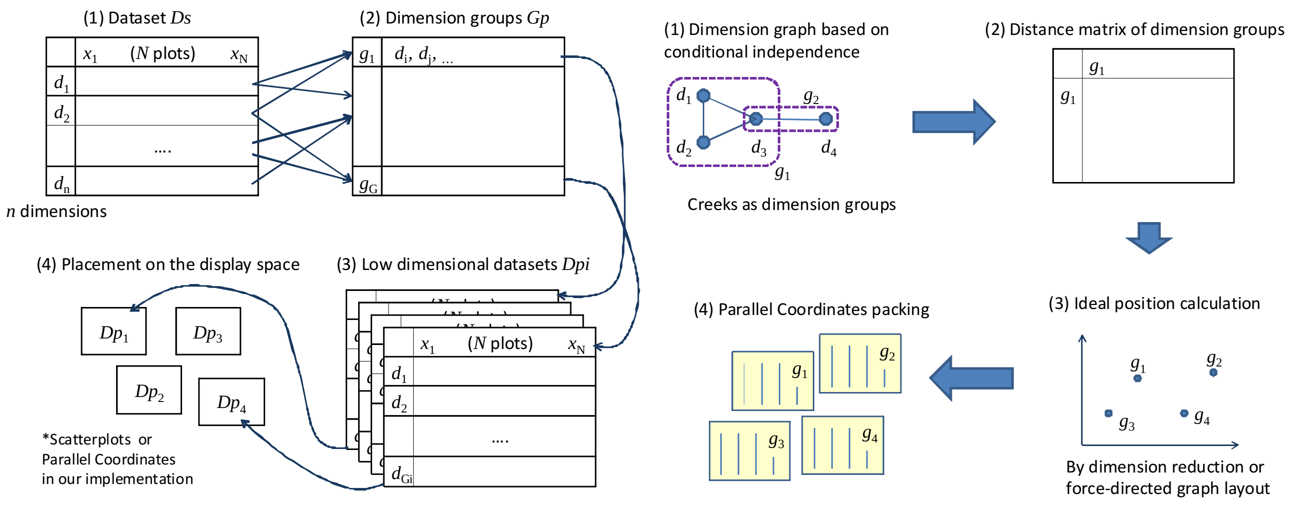
(1) Dataset *Ds*
x_1 (*N* plots) x_N
d_1
d_2
....
d_n
n dimensions
(2) Dimension groups *Gp*
g_1 d_i, d_j, ...
g_G
(4) Placement on the display space
(3) Low dimensional datasets *Dpi*
Dp_1
Dp_3
Dp_2
Dp_4
*Scatterplots or Parallel Coordinates in our implementation
x_1 (*N* plots) x_N
d_1
d_2
....
d_{Gi}
(1) Dimension graph based on conditional independence
d_1
d_2
d_3
d_4
g_1
g_2
Creeks as dimension groups
(2) Distance matrix of dimension groups
g_1
g_1
(3) Ideal position calculation
g_1
g_2
g_3
g_4
By dimension reduction or force-directed graph layout
(4) Parallel Coordinates packing
g_1
g_2
g_3
g_4

Figure 1: (Left) Illustration of the data structure and processing flow. (Right) Illustration of selecting and placing parallel coordinate plots.

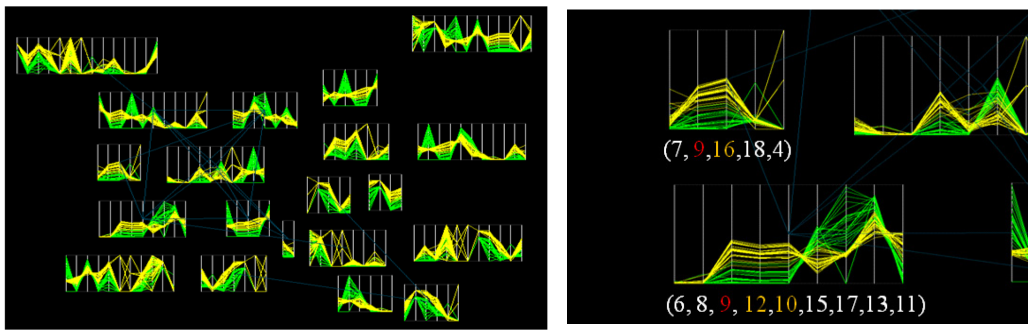

Figure 2: (Left) Visualization of an image segmentation dataset as 20 parallel coordinate plots. (Right) Visualization of an image segmentation dataset by a set of parallel coordinate plots components.

8: Standard deviation of the contrast of vertically adjacent pixels. Used for horizontal line detection.

9: Average over the region of (R + G + B)/3.

10: Average over the region of the R value.

12: Average over the region of the G value.

16: 3D nonlinear transformation of the RGB values using the algorithm of Foley and VanDam.

This type of correlation with multiple dimensions may be lost if we visualize the dataset using a single pair of parallel coordinate plots. However, our technique can adequately represent all correlations.

5.2 Example 2: Vehicle specification data

In this sub-section, we present visualization results obtained by applying our approach to a vehicle specification dataset constructed from websites containing on-line vehicle catalogs [22]. This dataset contains 70 feature values and various classes for 429 vehicles. Hence, the 70-dimensional dataset is displayed as 429 plots colored based on user-selected classes. Here, the feature values include sales-related values such as price, performance-related values such as displacement and fuel efficiency, and size-related values such as width and length. In addition, this dataset contains various average values of 5-point subjective user evaluations, including appearance, equipment, interior, engine power, cost performance, and the total.

In Figure 3(Left), we present an example of the visualization results . Here, the colors of polylines denote the countries of vehicle manufacturers, where yellow corresponds to France, blue corresponds to Germany, cyan corresponds to Japan, red corresponds to Sweden, and green corresponds to the USA. This result demonstrates that our technique extracts various meaningful subsets of correlated dimensions.

In Figure 3(Left), the dimensions of the parallel coordi-

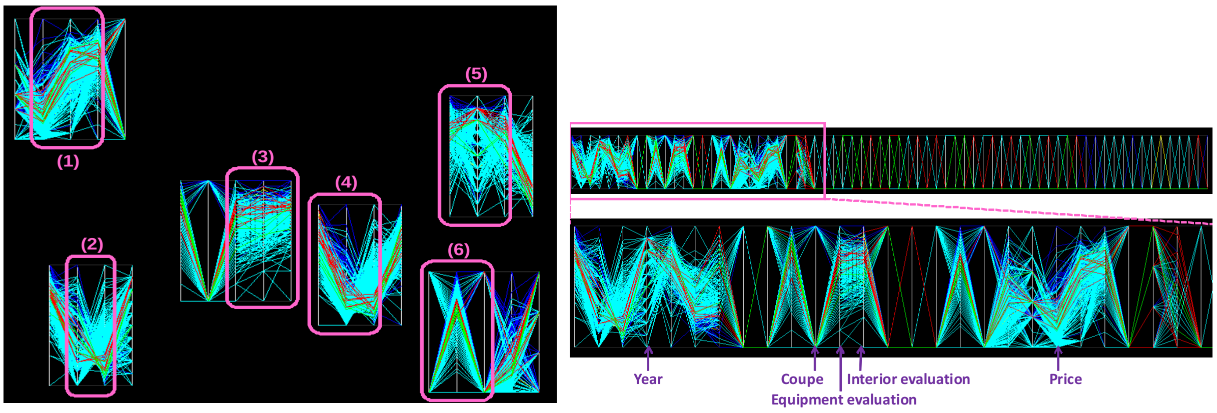

Figure 3: (Left) Visualization of a vehicle specification dataset by parallel coordinate plots components. Colors of polylines denote countries of vehicle manufacturers. (Right) Visualization of a vehicle specification dataset by a single parallel coordinate plots component.

nate enclosed be the rectangles indicated as (1) to (6) are as follows:

(1): ”Price,” ”outer length,” and ”cost performance evaluation.” The left two dimensions have positive correlation, while the right two dimensions have negative correlation. Long cars are not evaluated highly in this dataset.

(2): ”Displacement” and ”fuel efficiency,” which have negative correlation. Actually, it is quite common for vehicles with large displacement to have poor fuel efficiency.

(3): ”Equipment evaluation,” ”interior evaluation,” and ”total evaluation,” which have positive correlation. The plot suggests that the left two evaluations affect strongly the total evaluation.

(4): ”Interior evaluation,” ”price,” and ”fuel efficiency.” The left two dimensions have positive correlation, while the right two dimensions have negative correlation. In this dataset, vehicles with rich interiors or poor fuel efficiency are expensive.

(5): ”Engine power evaluation,” ”year,” and ”equipment evaluation.” Variable ”year” has negative correlations with the other two variables. This result, indicating that newer vehicles had worse evaluations was unexpected.

(6): ”Light car” (or not: binary values), ”engine power evaluation,” and ”coupe” (or not: binary value). Engine power evaluation has negative correlations with the other two valuables.

In Figure 3(Right), we visualize the same dataset using a single parallel coordinate plots. When all 70 dimensions are displayed in a single parallel coordinate plots, a horizontally large display space is required. Moreover, it is quite difficult to represent all correlations between arbitrary pairs of dimensions in a single parallel coordinate plots. In fact, displaying a dataset using a single parallel coordinate plots does not have many of the abovementioned features. For example, it is difficult to identify the correlation between ”year” and ”equipment evaluation,” which is well represented in Figure 3(Left)(5). Similarly, it is difficult to identify the correlation between ”interior evaluation” and ”price,” which is also well represented in Figure 3(Left)(4). This result demonstrates the advantage of our technique over the single parallel coordinate plots.

6 Conclusion

In this paper, we presented a technique for visualizing high-dimensional data as a set of well-selected and arranged low-dimensional data visualization components. Our technique selects meaningful groups of dimensions and visualizes them using parallel coordinate plots. Then, it computes the similarity between arbitrary pairs of dimension groups. Next, it computes the ideal positions of dimension groups based on their similarity distances. Finally, it adjusts the final positions of the parallel coordinate plots. We presented examples of applying our proposed technique to visualize image segmentation and vehicle specification datasets.

Our technique can represent various pairs or groups of correlated dimensions in the high-dimensional data spaces. However, it is not easy for users to discover dimensions that are ”unexpectedly independent.” In the example of the vehicle specification dataset, we expected that ”displacement” and ”fuel efficiency” are correlated with ”engine power evaluation.” However, results show that these dimensions are not well correlated. Moreover, the fact that these dimensions are not well correlated is not clearly represented in our visualization results. In future work, we will explore techniques that will enable users to discover such unexpectedly independent dimensions without any hassle.

Moreover, we would like to perform additional experiments in various applications. Specifically, we would like to apply our technique to larger datasets containing hundreds of dimensions and tens of thousands of plots to demonstrate its scalability. In addition, we would like to apply our technique to various fields of datasets to demonstrate its usability as a generic visual analytics tool.

Finally, we would like to numerically and subjectively evaluate our technique.

References

[1] M. ten Caat, N. M. Maurits, J. B. T. M. Roerdink, Tiled Parallel Coordinates for the Visualization of Time-Varying Multichannel EEG Data, *IEEE VGTC Symposium on Visualization*, 61-68, 2005.

[2] A. Chechetka, C. Guestrin, Efficient Principled Learning of Thin Junction Trees, *Advances in Neural Information Processing Systems*, 20, 273-280, 2008.

[3] N. Elmqvist, P. Dragicevic, J. Fekete, Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation, *IEEE transactions on Visualization and Computer Graphics*, 14(6), 1141-1148, 2008.

[4] S. Feiner, C.Beschers, Worlds within Worlds: Metaphors for Exploring n-dimensional Virtual Worlds, *ACM Symposium on User Interface Software and Technology (UIST '90)*, 76-83, 1990.

[5] Y. Fua, M. O. Ward, E. A. Rundensteiner, Hierarchical Parallel Coordinates for Exploration of Large Datasets, *IEEE Visualization*, 43-50, 1999.

[6] G. Grinstein, M. Trutschl, U. Cvek, High-Dimensional Visualizations, *KDD Workshop on Visual Data Mining*, 2001.

[7] A. Inselberg, B. Dimsdale: Parallel Coordinate: A Tool for Visualizing Multi-Dimensional Geometry, *IEEE Visualization*, 361-370, 1990.

[8] T. Itoh, H. Takakura, A. Sawada, K. Koyamada, Hierarchical Visualization of Network Intrusion Detection Data in the IP Address Space, *IEEE Computer Graphics and Applications*, 26(2), 40-47, 2006.

[9] T. Itoh, C. Muelder, K. Ma, J. Sese, A Hybrid Space-Filling and Force-Directed Layout Method for Visualizing Multiple-Category Graphs, *IEEE Pacific Visualization Symposium*, 121-128, 2009.

[10] J. Johansson, P. Ljung, M. Jean, M. Cooper, Revealing Structure within Clustered Parallel Coordinates Displays, *IEEE Information Visualization*, 125-132, 2005.

[11] D. Karger, N. Srebro, Learning Markov networks: Maximum bounded tree-width graphs, *the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, 392-401, 2001.

[12] G. Leban, I. Bratko, U. Petrovics, T. Curk, B. Zupan, VizRank: Finding Informative Data Projections in Functional Genomics by Machine Learning, *Bioinformatics Applications Note*, 21(3), 413-414, 2005.

[13] T. Mihalisin, E. Gawlinski, J. Timlin, J. Schwegler, Visualizing a Scalar Field on an n-dimensional Lattice, *IEEE Visualization*, 255-262, 1990.

[14] W. Peng, M. O. Ward, E. A. Rundensteiner, Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering, *IEEE Information Visualization*, 89-96, 2004.

[15] R. Rosenbaum, J. Zhi, B. Hamann, Progressive Parallel Coordinates, IEEE Pacific Visualization Symposium, 25-32, 2012.

[16] M. Sips, Selecting Good Views of High-Dimensional Data Using Class Consistency?C *Eurographics/IEEE-VGTC Symposium on Visualization*, 831-838, 2009.

[17] L. Wilkinson, A. Anand, R. Grossman, Graph-Theoretic Scagnostics, *IEEE Symposium on Information Visualization*, 157-164, 2005.

[18] P. C. Wong, R. D. Bergeron, 30 Years of Multidimensional Multivariate Visualization, Scientific Visualization: Overviews Methodologies and Techniques, *IEEE Computer Society Press*, 3-33, 1997.

[19] Z. Zhang, K. T. McDonnell, K. Mueller, A Network-Based Interface for the Exploration of High-Dimensional Data Spaces, *IEEE Pacific Visualization Symposium*, 17-24, 2012.

[20] H. Zhou, X. Yuan, H. Qu, W. Cui, B. Chen, Visual Clustering in Parallel Coordinates, Computer Graphics Forum, 27(3), 1047-1054, 2008.

[21] UCI Machine Learning Repository, Image Segmentation Data Set, http://archive.ics.uci.e.,du/ml/datasets/Image+Segmentation

[22] YAHOO! JAPAN Automobile, http://autos.yahoo.co.jp/