Vis Comput (2012) 28:541–551
DOI 10.1007/s00371-012-0713-0

ORIGINAL ARTICLE

Structure revealing techniques based on parallel coordinates plot

Xin Zhao · Arie Kaufman

Published online: 26 April 2012

Abstract Parallel coordinates plot (PCP) is an excellent tool for multivariate visualization and analysis, but it may fail to reveal inherent structures for complex and large datasets. Therefore, polyline clustering and coordinate sorting are inevitable for the accurate data exploration and analysis. In this paper, we propose a suite of novel clustering and dimension sorting techniques in PCP, to reveal and highlight hidden trend and correlation information of polylines. Spectrum theory is first introduced to specifically design clustering and sorting techniques for a clear view of clusters in PCP. We also provide an efficient correlation based sorting technique to optimize the ordering of coordinates to reveal correlated relations, and show how our view-range metrics, generated based on the aggregation constraints, can be used to make a clear view for easy data perception and analysis. Experimental results generated using our framework visually represent meaningful structures to guide the user, and improve the efficiency of the analysis, especially for the complex and noisy data.

Keywords Parallel coordinates plot · Dimension sorting optimization · Visual representation

1 Introduction

Complex structures must be extracted, shown, and analyzed for the multivariate data, but they can be difficult to visualize and graph. Parallel coordinates plot (PCP), first presented by Inselberg et al. [15], is a great high-dimension (N-D) visualization tool. In this approach, each dimension is drawn as a vertical line, therefore, each multidimensional point is visualized as a polyline that crosses each axis at its corresponding coordinates. This methodology facilitates the presentation of very complex N-D datasets in a single 2D image. Usually, in virtually every scientific field dealing with empirical data, people attempt to get a first impression on their data by trying to identify and analyze groups of similar behavior in the data, then clustering and dimension sorting are extremely important to help reveal trend and correlation information of major clusters. Aiming to create tools suitable for the analysis of large and complex datasets, we design new clustering algorithms and dimension sorting methods based on the spectrum theory, correlation properties, and view-range metrics in the format of parallel coordinates. Our experimental results demonstrate the potential visual abilities of our framework to reveal data clusters and structure aspects: information characteristics between adjacent axes and across the entire coordinates can be shown in diverse PCP forms for data analysis and exploration.

Our paper is organized as follows. Section 2 presents the related work and a summary of our contributions. Our design and implementation details are described in Sect. 3. Section 4 analyzes experimental results obtained by our framework. In Sect. 5, we draw conclusions and discuss the future work.

2 Related work

Many research scientists have addressed the N-D data visualization and visual clustering in PCP. However, basic PCP

X. Zhao (✉) · A. Kaufman
Computer Science Department, Stony Brook University,
Stony Brook, NY, 11794, USA
e-mail: xinzhao@cs.stonybrook.edu

A. Kaufman
e-mail: ari@cs.stonybrook.edu

tools for representing classes, such as multicolored brushing [25], contain a major shortcoming: strong crossing and overlappings of a large number of polylines severely hamper the user's ability to identify patterns in the data from its visual representation. Tackling with this problem, many clustering methods, such as grand tour [26] and binning algorithm [21], have been proposed to group nearby N-D polylines into a single representative cluster. Colors and opacities are utilized to show polyline memberships in each individual cluster. Some approaches focus on multiresolution techniques and impose the hierarchical brush or filter. Fua et al. [10] have provided a multiresolution view of the data via hierarchical clustering, and used a variation in PCP to convey the aggregation information for clusters. However, the band for each calculated cluster is not shown to be aggregated and clear. Later, they have designed an interactive brushing technique for minimizing clutter, which permits the user to manually omit portions of the data during rendering [11]. However, it is time-consuming for complicated datasets. For the aggregation of polylines, Zhou et al. [27] have proposed an optimization scheme designed to minimize the curvature of polyline edges and maximize the parallelism of adjacent edges through an energy function. However, they only consider the line or point distributions without a clear correlation detection between adjacent axes, and fail to consider the ordering of entire coordinates.

Density-based approaches, such as [1], are very helpful in achieving a distribution view of the data and can reveal feature information that may be obscured by overlapping polylines. Johansson et al. [16] have employed clustering to reveal the inherent structure of the data, and displayed the structure with high-precision texture through transfer functions on the density plot. Later, they have further applied depth cues and density features to explore temporal datasets [17]. However, the overlapping resulting from crossings of polyline segments may lead to the overaggravated cluttering, or the inappropriate dimension sorting order may fail to reveal nice clear bands. Therefore, reforming positions of coordinates provides an alternative method to reduce the clutter of bundles and reveal hidden correlations. Novotny [22] has shown that the user appreciates the effort to gather related polylines for prominent views and supported that the ordering technique provides solutions for a clear visualization. Peng et al. [23] have restructured datasets in an automatic or semiautomatic manner to minimize clutter for the multidimensional data visualization using a dimension reordering technique.

2.1 Contributions

The goal of our framework is to provide novel clustering and dimension sorting techniques for the multivariate data to reveal clusters and correlations in PCP. Our framework well-implements the following tasks: (1) examine correlations with explicitly mathematical definitions between variables represented by neighboring axes; (2) aggregate polylines to reduce visual clutters; and (3) represent each individual cluster for a easy understanding of data properties or relationships. The specific contributions and benefits of our framework are as follows:

- A spectrum based clustering method first applied in PCP, conveys major trends across the entire axes and helps to eliminate outliers and noises.
- A new correlation based dimension sorting technique with the corrgram, accurately and efficiently recognizes and reveals various predefined correlations between adjacent axes in PCP.
- A new metric, view-range matrix, specifically defined for dimension sorting in PCP, generates the optimal coordinate order to maximally increase the aggregation of polylines.
- A novel dimension sorting approach using spectral clustering theory, reveals a clear view of clusters by maximally extending the measure distance between clusters.

3 Design theory and algorithm

In this section, we focus on the design details: clustering of polylines, dimension sorting, and visual representation.

3.1 Theory and algorithm of spectral clustering

Spectral clustering has solid theory foundations. Donath and Hoffman [5] have first suggested to construct graph partitions based on eigenvectors of adjacency matrix. Fiedler [7] has further suggested to use eigenvectors to partition a graph. Since then, spectral clustering has been discovered, rediscovered, and extended in different areas. The success of spectral clustering is mainly based on the fact that it does not make strong assumptions on the form of clusters and then be easily applied to various cases. Two most common objective functions used to build graph partitions from eigenvectors of the adjacency matrix are Ratiocut [12] and the normalized cut Ncut [24].

Spectral clustering algorithm The main tools for spectral clustering are graph Laplacian matrices. There exists a whole field dedicated to the study of those matrices, called spectral graph theory [3]. Given a dataset consisting of n data points $x_1, \ldots, x_n$, which can be arbitrary dimensions, we measure their pairwise similarities $s_{ij} = s(x_i, x_j)$ by some similarity function which is symmetric and nonnegative, and then denote the corresponding similarity matrix by $S = (s_{ij})_{i,j=1,\ldots,n}$. Parameter k is the total number of clusters we want to construct and W is the weighted adjacency matrix of the constructed similarity graph. With respect to the theory of Laplacian eigensystem, the Laplacian of matrix is defined as $L = D - W$, where D is a diagonal degree matrix and its diagonal entries are given by summation of the rows of W. The normalized Laplacian is defined as

$$L = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

Then we compute the first k eigenvectors $u_1, \ldots, u_k$ of L, and let $U \in R^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns. Next, for $i = 1, \ldots, n$, let $y_i \in R_k$ be the vector corresponding to the ith row of U, and then cluster the points $(y_i)_{i=1,\ldots,n}$ in R_k with the k-means algorithm into clusters $C_1, \ldots, C_k$. Finally, we have the clustering result with clusters $A_1, \ldots, A_k$, where $A_i = \{j | y_j \in C_i\}$. The main idea of this algorithm is to change the representation of the abstract data points x_i to points $y_i \in R^k$. Due to the properties of the graph Laplacians, this change of representation is useful to enhance the cluster-properties in the data, so clusters can be trivially detected in the new representation.

Spectral clustering merits Results obtained by spectral clustering often outperform the traditional approaches because of the application of graph Laplacians: there are no issues of getting stuck in local minima or restarting the algorithm for several times with different initializations [3]. Moreover, spectral clustering can be easily adapted in PCP, and be solved efficiently by standard linear algebra methods. In this paper, we first apply the spectrum theory in PCP as a new clustering and dimension sorting technique, to convey major clusters and trends.

3.2 Data preparation

All the datasets and visualization display used in this paper are courtesy of the XmdvTool [25]. Because the plot of polylines is based on the interpolation of consecutive pairs of variables, data scaling is necessary in order to reveal meaningful structures. Thus, normalization is an important data preparation process in our framework.

3.3 Clusterings of polylines

As the number of data record increases, the identification ability of potentially clusters of data items is seriously damaged because of the overlapping of polylines in PCP. Therefore, clustering is extremely important to enhance the presentation of useful information and to avoid displaying irrelevant one.

3.3.1 Dendrogram based k-means clustering

One of the most popular clustering methods is the k-means clustering, but it has major issues—the strong dependency on the initial setting of the cluster number k and the random initialization of centroids. To solve these problems, we build a similarity matrix and examine interior relations to construct a dendrogram, which can automatically find the best cluster number k for various datasets. We adapt the cluster similarity parameter S proposed in [6]:

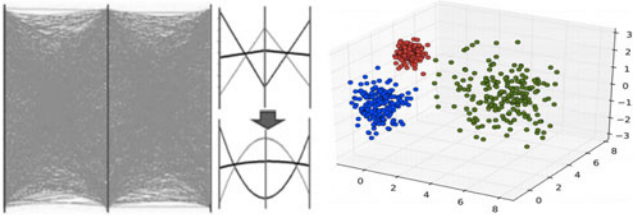
$$S(C_1, C_2) = \sum \cos(d_1, d_2) / \bigl(size(C_1) \times size(C_2)\bigr),$$

where d_1 and d_2 are the average distance between elements of cluster C_1 and C_2. We follow these steps: first, set the threshold w_h; next, compute similarity between all pairs of clusters (each point is a cluster at the initial step), merge the most similar two clusters together and update the similarity matrix M; then repeat merging and updating until all values in M are smaller than w_h; and automatically output the best cluster number k at last. In general, large w_h leads to more clusters. We experimentally find that values in the range $0.4 \leq w_h \leq 0.6$ generate reasonable results.

3.3.2 Spectrum based clustering

We first propose an approach of clustering data using spectral synthesis, for clutter reduction and cluster detection in PCP. Our design enables an alternative PCP clustering view and provides great potentials in exploring and revealing underlying patterns.

The equally spaced vertical lines are the axes of the parallel coordinate system and the polygonal line segment is a plotted point. Note that spectra have a very natural representation in PCP with a parallel axis for each spectrum wavelength. The theory of spectrum based clustering is inspired by a basic spectral synthesis theory that any curve can be reconstructed by trigonometric functions (with parameters of amplitude, frequency, and phase-shift). After deforming polylines into polycurves for each data point, as shown in Fig. 1a, our framework decomposes each polycurve S into a series components: $S = \sum_{j=1}^{n} T_j$, where n



(a) (b)

Fig. 1 Spectrum based clustering design. (**a**) Illustration of spectrum generation from polylines. (**b**) The k-means ($k = 3$) clustering result in the parameter space formed by the first component: $vs_1 = (A_1, \omega_1, \varphi_1)$

is the total number of components (larger n can generate more accurate curve reconstruction by preserving fine details), and $T_j(t) = A \times Tri(\omega t + \varphi)$ is a series trigonometric functions defined by the amplitude A, angular frequency ω, and phase φ. Because the selection standard for each component is to minimize the sum of squared errors at current step, the ordering of components is naturally fixed during the computation. According to our experiments, in general, at most top three components can visually reconstruct an accurate polycurve for clustering. We define a trigonometric function based vector v to describe the selected first m components for each polycurve: $v = (vs_1, \ldots, vs_m)$, where $vs_i = (A_i, \omega_i, \varphi_i)$ and $m \leq 3$ for most cases. Therefore, as shown in Fig. 1b, polylines are transformed into corresponding points in a new parameter space formed by the first component, and we apply k-means clustering algorithm to group points together. More components can be further used for the hierarchically clustering refinement. An advantage of our design is that our method mainly focuses on the shape of each trend across the entire axes rather than the distribution of data points at the coordinates (e.g., traditional k-means clustering in the high-dimensional domain), which may help to reveal hidden structures and trend associations in multivariate datasets (e.g., Fig. 8).

3.4 Dimension sorting of parallel coordinates

Although PCP offers simple visualization and interactions for the user to explore the high-dimensional data, dimension sorting is inevitable and critical to create the illuminating result for typical data analysis. How to find the best ordering of coordinates is an important question in PCP. There are various methods trying to solve this problem, such as random swap or greedy algorithm with minimal outliers [22], and screen-space metrics [4]. In this section, we propose three dimension sorting optimization algorithms based on matrices of correlation, view-range, and spectral clustering, respectively. In addition, pseudocolored maps with respect to the variable ordering are rendered to facilitate the perception of structures in PCP.

3.4.1 Visual display of dimension sorting optimization

In order to have the optimization of dimension sorting, we first build the user preferred matrices to describe relations between any two attributes of the input dataset. Then, for the exploratory visual display, we follow two general principles: (1) the multi-color display: render the value of a matrix with different colors to depict its sign and magnitude, with the purpose to improve the ability of visual perception; and (2) the effect-ordered data display [9]: reorder variables in a matrix, so that "similar" variables are positioned adjacently for depicting patterns of relations among variables in matrices directly, particularly when the number of variables is moderately large.

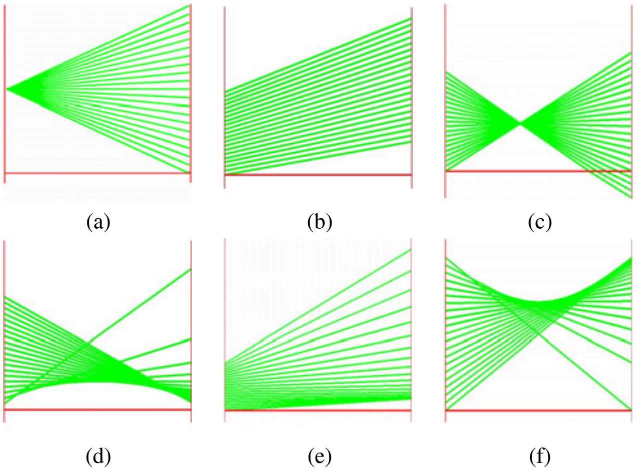


Fig. 2 Major correlation patterns between adjacent axes [14]: (**a**) constant, (**b–c**) direct and inverse linear, (**d**) hyperbola, (**e**) parabola, and (**f**) ellipse styles, are shown in PCP

3.4.2 Correlation based dimension sorting

We first propose a correlation based dimension sorting using some predefined correlation types, as shown in Fig. 2. Moreover, the application of the corrgram, an exploratory display for correlation matrices [8], makes it easy to reduce the clutter of bundles and reveal hidden correlations. For large or complex datasets, the corrgram may have advantages especially for exploratory purposes, because it shows all correlations, rather than just a low-dimensional summary. However, the traditional corrgram only works for the linear correlation. For more general applications, we modify it to detect and score various nonlinear correlations as follows: for the point sets (x, y), we redefine $x' = f(x)$, where f is the precalculated correlation functions (e.g., hyperbola or parabola), and then calculate the Pearson correlation coefficient of new point sets (x', y). Based on coefficients among variables, the corrgram forms a pseudocolor map by defining $+1$ direction as a perfect positive (increasing) relationship while -1 direction as a perfect decreasing (negative) relationship, and 0 as undefined correlation (random noise) for both linear and non-linear correlations. Figure 3 shows two advantages of our modified corrgram: it shows accurate correlations between any two axes even under a very noisy situation, and it qualitatively represents the level of noise. We further define two sorting criteria for different user preferences: searching for a dimension sorting vector s of n dimensions, which satisfies the following equations: $\max(\sum_{i=1}^{i=n-1} S_{s_i s_{i+1}})$, named *sorting by value* to only highlight the direct correlations, or $\max(\sum_{i=1}^{i=n-1} |S_{s_i s_{i+1}}|)$, called *sorting by magnitude* to emphasize both direct and inverse correlations, where $S_{s_i s_{i+1}}$ is the coefficient score between two adjacent coordinates.

This design can be easily extended as a hierarchical application to find subcorrelations. For complicated datasets

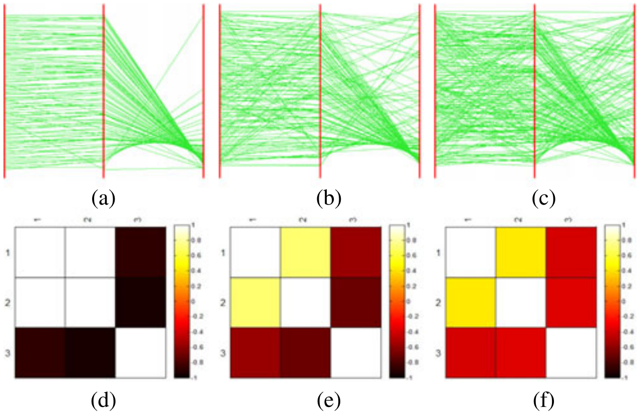


Fig. 3 Properties of our modified corrgrams. (**a**)–(**c**) A 100-item and three dimensional synthetic dataset with different levels of noise, (**a**) 5 %, (**b**) 50 %, (**c**) 100 %. There is a direct linear correlation between first two coordinates, and a parabola correlation between the last two coordinates. (**d**)–(**f**) Corresponding pseudocolor maps (corrgrams). Scores proportionally head to zero with the increasing noise level

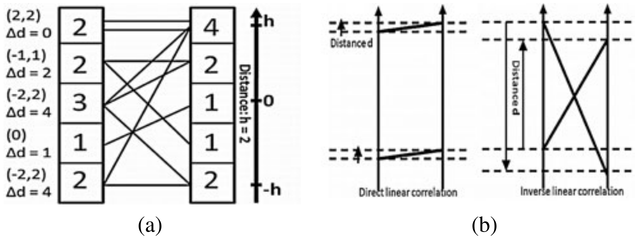


Fig. 4 Illustration of the generation of view-range matrix. (**a**) Definition of view-point matrix. View-range is calculated and recorded for each bin (a grid with the total polyline number inside itself) between adjacent axes. (**b**) A simple case to show that smaller sum value of vertical distance (direct linear correlation) has better parallelism than the larger one (inverse linear correlation)

with a mixture of several individual clusters (each individual cluster with a unique correlation), there might be no clear correlation view detected by our method if using entire data items. Therefore, if all the values shown in the corrgram are in the weak correlation range (e.g., between the range $[-0.3, 0.3]$), our system will automatically apply our dendrogram based k-means clustering algorithm to generate clusters. Then the same dimension sorting method is applied for each cluster, and the user can interactively select and analyze a cluster of interest at one time.

3.4.3 *View-range matrix based dimension sorting*

Hauser et al. [13] have provided a bar chart-style rendering of a histogram on each axis in PCP. Inspired by them, we present a view-range matrix to find the best coordinate ordering with the maximally local aggregation of polylines.

As shown in Fig. 4a, for each bin, a vector with two distance elements: $vr(d_1, d_2)$, is calculated and recorded to describe the view-range between adjacent axes. The distance element d, for each axis, is defined to record the slope of polylines between axes, measured as the vertical distance. The steepest line going up or down covers the entire height of the coordinate, and can thus have a vertical distance in the range $[-h, h]$ with a fixed view point (value 0 at the middle point of the selected coordinate). Therefore, for each $vr(d_1, d_2)$, we have the view range $-h \leq d_1, d_2 \leq h$, and the total view range of $2h + 1$ bins. Thus, we can do a reasonable job of placing the variable axes in a well-defined optimal unidimensional order through building the view-range matrix with the following criterion: any two axes can form a axis pair, and for each axis pair, we define *the length sum of view-range* d_t as follows:

$$d_t = \sum_{\text{bin}=0}^{\text{bin}=2h} \Delta d_{\text{bin}} = \sum_{\text{bin}=0}^{\text{bin}=2h} |d_1 - d_2|_{\text{bin}},$$

which describes the gathering situation using the sum of the view-range length of all bins. A small d_t means good gathering of polylines, so the minimization of d_t will generate the maximal aggregation of polylines between axis pairs. We resort variables using this criterion to find the best sorting order.

Our framework also supports to find the optimal dimension ordering with the minimal length sum of view-range between adjacent coordinates under the bin-level. Therefore, all the polylines are maximally gathered between neighboring axes for each bin, which can reveal and highlight hidden structures in a subview. Moreover, because the view-range matrix is calculated based on bins at coordinates and the bin number can be flexible selected by the user, it is very suitable for the aggregation optimization of datasets with complex components. By setting the distance unit h according to different requirements, our framework can flexibly provide multi-level hierarchical views for datasets, as shown in Fig. 4b. If we minimize the length of each bin into the pixel-level, each polyline (or overlapped polylines) will form a single bin. The length of view-range d_t will be the sum of absolute vertical distance between end points of each polyline, which can be used to calculate angles between the given axes [4]. Therefore, our algorithm supports to find the best dimension ordering with the optimal parallelism to the horizontal axis of PCP, which may help reduce the visual clutter.

After the generation of the view-range matrix, we adapt similar display style as the corrgram to visualize all possible combinations of the length sum of view-range (d_t) using different colors and to find the best ordering, named viewgram (Fig. 5Left). Our method, based on the distance associated within the total $2h + 1$ bins, considers the local aggregation to generate the optimal visualization results. However, the length sum of view-range (d_t) is easily affected by outliers in

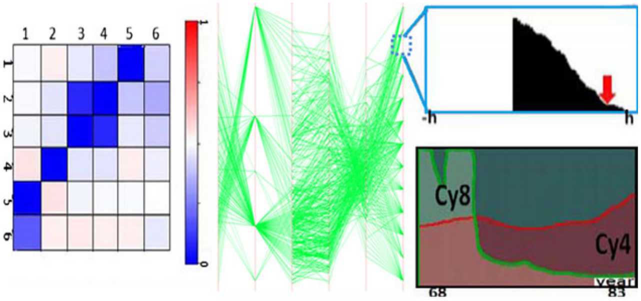


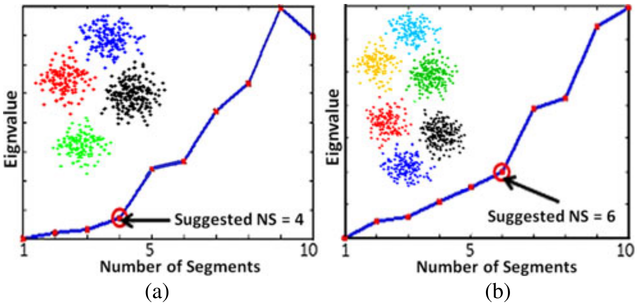
Fig. 5 Viewgram-style visualization of the view-range matrix, generated from the car dataset. (*Left*) The viewgram shows all combinations of the length sum of view-range (d_t). (*Right*) Subhistogram (*top*) of a bin of interest shows the distribution of view-range. The tail (*red arrow*) in the subhistogram can be automatically removed to narrow down the view-range, based on a cutting parameter. Multiple 1D subhistograms (*bottom*) reveal the sublevel relations between Cylinders and Year

each bin, thus, subhistograms can be used to remove outliers and to refine the view-range matrix by eliminating the noise, as shown in Fig. 5Right. Meanwhile, for a selected axis pair, subhistograms reveal the distribution of view-range inside a bin of interest, which can be used to explore different distribution types between adjacent axes with similar view-range length. Thus, our framework makes it easy to reveal sublevel relations of interest when there are no obvious overall correlations between adjacent axes. For example, Fig. 5 shows relations between variables Cylinders and Year of the car dataset, detected and highlighted by 1D sub-histograms: from 60's to 80's, the production of cars with 4 cylinders is very stable with slightly increasing, while the production of cars with 8 cylinders keeps dropping after mid-1970s.

3.4.4 Spectral clustering based dimension sorting

Spectral clustering is a good method that satisfies (1) invariance clustering during coordinates sorting; (2) close cluster segments to human perception, and (3) robustness to numerical noise in the dataset. Therefore, we develop a robust spectral clustering based dimension sorting scheme with three main steps: first, similarity graphs are appropriately designed to generate accurate clustering results; second, the number of the cluster/segments is estimated based on the analysis of the behavior of the Laplacian spectrum, which has a high consistency toward human perception; third, a spectral clustering based dimension sorting scheme helps to reveal a clear cluster view.

Similarity graphs design First step is to transform a given set $x_1, \dots, x_n$ of data points with pairwise similarities s_{ij} into a graph. With the goal of modeling the local neighborhood relationships between data points, we constructing a *fully connected graph* as similarity graph: simply connect all points with positive similarity with each other, and then weight all edges by s_{ij}. A Gaussian similarity function, $s(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$, with the parameter σ controlling the width of the neighborhoods, is use to represent and model the local neighborhood relationships. Other popular graph design, such as ε-neighborhood graph and k-nearest neighbor graph can be easily embedded into our framework. Although choosing a good similarity graph is not trivial and spectral clustering may be unstable under some choices of parameters for neighborhood graphs, our design can produce good results for all the applied cases in our paper.

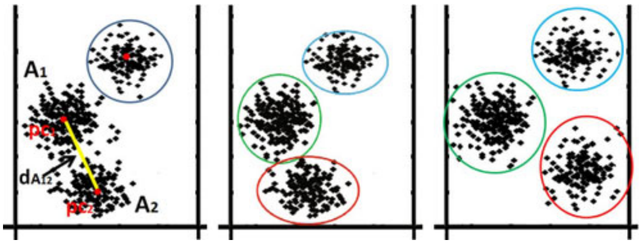


(a) (b)

Fig. 6 Eigencurves for estimating perceptually consistent number of cluster segmentations for different data distribution demos. Eigencurves plot the relationship between eigenvalues and the number of clusters. The first point on the curve with rapid eigenvalue change suggests the number of clusters for perceptually clustering. The points highlighted in *red circles* on the curves suggest the number of clusters for each scatter plot of data. We can see from the figure that the suggested NS matches the preferred number of clusters by human perception. Note that, NS denotes number of clusters

Estimation of k-number Choosing the number k of clusters is a general problem for spectral clustering. We describe a simple but effective approach based on the analysis of the behavior of the Laplacian spectrum. The goal is to choose the number k such that all eigenvalues $\lambda_1, \dots, \lambda_k$ are very small, but λ_{k+1} is relatively large. Let $0 = \lambda_1 \leq \lambda_2 \dots \lambda_n$ be the Laplacian spectrum of the model, where λ_i is the *ith* eigenvalue and $Dif(i, j) = \lambda_j - \lambda_i$. Then we have the following observation: the number i is likely to coincide with the number of components in a natural segmentation when there is a dramatic increase in $Dif(i, i+1)$ for the first time. We provide the following clustering experiments for verifying the observation. We define the curve plotting of the eigenvalue versus the number of cluster segments as an eigencurve. Figure 6 displays eigencurves for cluster segments of different data distributions.

Optimization of dimension sorting After getting the clustering result for the input dataset, the center of each cluster, which is defined as the average representative data point within each segment, is discovered by a center hunting algorithm: for clusters $A_1, \dots, A_k$ generated from spectral clustering, we find each cluster center pc_i of A_i using the average Eulerian distance, $pc_i = \frac{1}{N_i} \sum_{m=1,\dots,N_i} p_m$, where N_i



(a) (b) (c)

Fig. 7 Illustration of the spectral clustering based dimension sorting. (**a**) Illustrative image of distance calculation. *Red points* are center points of clusters (pc). *Yellow line* highlights the distance used to compute *the average distance between clusters* $d_{C_{\mathrm{bt}}}$, while blue circle shows all data points used to calculate *the average distance within each cluster* $d_{C_{\mathrm{in}}}$. The optimization results of the total cluster-distance (d_s) between adjacent axes, (**b**) without and (**c**) with the weighting scheme ($w_{\mathrm{bt}} = 3$, $w_{\mathrm{in}} = 0.5$). By comparison, using different weighting values, our framework finds an optimization result with more cluster separation ($w_{\mathrm{bt}} = 3$) but less gathering within the cluster ($w_{\mathrm{in}} = 0.5$). Clusters are highlighted using different colors

is the total number of data points inside the cluster A_i, and p_m is the data point with $m = 1, \dots, N_i$.

With the goal of maximally (1) extending the distance between clusters and (2) aggregating data points within each cluster, the following equation is used for the optimization of dimension sorting in PCP: for all the classified data points plotted in the scatter plot between any two coordinates, we calculate *the total cluster-distance* d_s,

$$d_s = d_{C_{\mathrm{bt}}} - d_{C_{\mathrm{in}}},$$

where $d_{C_{\mathrm{bt}}}$ is *the average distance between clusters*, while $d_{C_{\mathrm{in}}}$ is *the average distance within each cluster*.

As shown in Fig. 7a, $d_{C_{\mathrm{bt}}}$ is defined as

$$d_{C_{\mathrm{bt}}} = \frac{1}{C_k^2} \sum_{i,j=1,\dots,k} d_{A_{ij}} = \frac{1}{C_k^2} \sum_{i,j=1,\dots,k} \| pc_i - pc_j \|,$$

while $d_{C_{\mathrm{in}}}$ is computed using

$$d_{C_{\mathrm{in}}} = \frac{1}{N_i} \sum_{m=1,\dots,N_i} \| p_m - pc_i \|,$$

where C_k^2 is the total combinations of k clusters, and pc_i, pc_j are the center points of clusters A_i and A_j.

We have further designed weighting parameters to modify the distance measurement equation to generate diverse optimization sorting results as follows:

$$d_s = w_{\mathrm{bt}} d_{C_{\mathrm{bt}}} - w_{\mathrm{in}} d_{C_{\mathrm{in}}},$$

where weighting parameters w_{bt}, $w_{\mathrm{in}} \geq 0$, are used to control the weighting proportion of $d_{C_{\mathrm{bt}}}$ and $d_{C_{\mathrm{in}}}$, respectively. Parameter settings with specified properties, are able to control the optimization styles of dimension reordering, which will generate different sorting results to satisfy various requirements of the user. We show experimentally that our weighting scheme is useful in practice to generate various optimally sorting results, as shown in Figs. 7b and 7c.

Large cluster-distance d_s between adjacent axes means good gathering of polylines inside each cluster and maximal separation between clusters, so the maximization of $\sum_n d_s$ will optimally generate the local aggregation in each cluster and the global separation between clusters with the final dimension ordering result n. After the generation of the spectral clustering based matrix using the cluster-distance (d_s) between adjacent coordinates, we adapt the same display as the corrgram to visualize all possible combinations of the total cluster-distance (d_s) using different colors, named specgram. Then the specgram is used to find the best dimension sorting order n: the search are propagated across the axes from left to right to determine the ordering of coordinates, using the greedy algorithm.

3.5 Implementation

In order to maximize the system efficiency, our framework (1) implements the clustering, the optimal coordinates ordering, and distribution view (defined in Sect. 4) as the offline procedure (couple seconds for most cases), (2) allows the user to interactively adjust polylines and set parameters, and (3) accelerates texture rendering of polylines in real time using graphics hardware. Mathematical calculations are implemented using Matlab and C++ on CPU, while interactive operations and real-time rendering are implemented using OpenGL and Cg libraries on GPU.

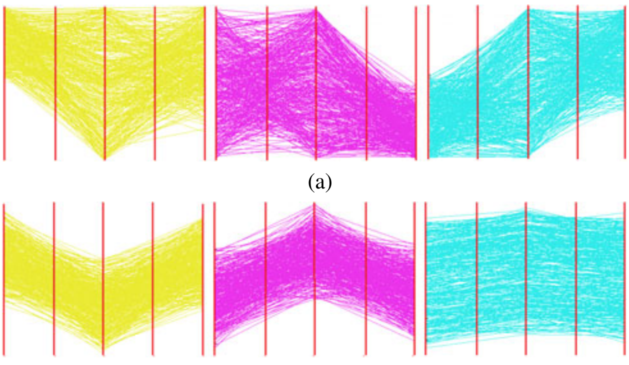
In general, finding an optimal ordering of axes for parallel coordinates is equivalent to the traveling salesman problem, and thus NP-complete [18]. During the implementation, for each dataset, we take the exhaustive permutation calculation between axes for the generation of correlation, view-range and cluster-distance matrices as precomputation. The computation of specified matrix for graph view, is very fast and efficient [8]. The generation of both view-range and cluster-distance matrices, according to the definition, is a linear efficiency between each axis pair. The exhaustive permutation of coordinates ordering is only performed once, and the computational time mainly depends on the item number and dimensionality of datasets. Using these computed constraint matrices, our system is able to efficiently optimize the dimension ordering to create a good display for the analyst.

Another time-consuming step is the creation of distribution view, which relies on the pixel resolution. Following the experimentation in [19], a prefixed viewport of 512 × 384 pixels with a neighborhood radius of 12 pixels is set, which provides good results with full resolution for most datasets. For the visualization, the computational time to create plot textures as visual views mainly depends on the viewport size, item number and dimensionality of the data. For example, it only takes approximately 320 ms to create a polyline texture for a five-dimensional dataset containing 3,800 data items in our prefixed viewport.

Applying the user designed transfer function to an appointed texture works as a specified mapping from an N-D vector to the specified color and opacity of each pixel. The information forms a texture stored in the framebuffer, which will be recomputed and redrawn on the screen in real time when the transfer function changes. Although some calculation, such as the generation of constraint matrices and distribution view, could be easily accelerated by shifting them to GPU for the parallel computation, our strategy is fast enough for interactive operations and efficient display.

4 Experimental results

In this section, we test some datasets to demonstrate the design merit of our framework (because of the page limitation, more experimental results are shown in the additional material). Aiming to obtain good visual-representations of large and complex datasets, our framework shows a clear quantization level of the distribution of polylines (distribution view) for the accurate visual perception with less clutter (e.g., Fig. 9b). Inspired by [1], different cluster densities are visualized with different opacities, where a high opacity corresponds with a major cluster containing large data points. The user can flexibly set colors and opacities through our framework.


(a)

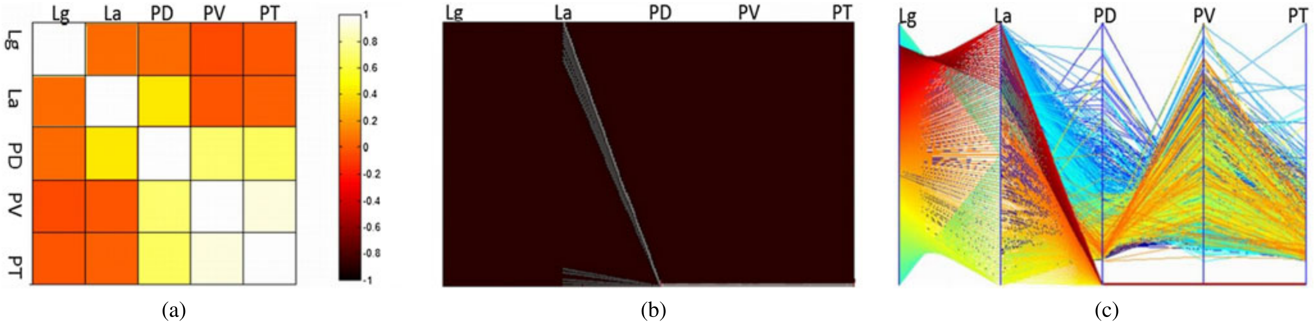
(b)

Fig. 8 Spectrum based clustering for the 1523-item, five-dimensional Nasdaq dataset with variables Open price, and Close price for today and the next three days. Clustering results using (**a**) the k-means clustering and (**b**) our spectrum based clustering methods, both shown in the format of parallel coordinates. Clusters are illustrated by different colors

4.1 Experimental results of clustering

Our spectrum based clustering method can accurately detect and classify interior trends across the entire axes, revealing helpful information for the user. For example, for the Nasdaq dataset, by comparison with the clustering directly according to the item values (e.g., k-means clustering based on the price differences), our spectrum based clustering method accurately detects three major trends of interest: up-down trend, down-up trend, and smooth-trend, providing much useful investment information for the analyst, as shown in Fig. 8. Trend information is more important than the clusters of stock price because it reflects the price change of stocks with time, rather than the simple capital value. This function is suitable for various time-series datasets to reveal data variation over time [2].

Clustering comparison In order to show the merit of spectral clustering by comparison with other clustering algorithms, especially for the high-dimensional data, we use distance measures, including clustering error (CE) and variation of information (VI) [20]. CE, as an intuitive way to compare clusterings, is to calculate the clustering error after an optimal matching of clusters. VI is a recently proposed clustering criterion based on information theoretic concepts [20]. It measures the amount of information that we gain and lose when going from one clustering method to another clustering technique. We compare our spectral



(a)

(b)

(c)

Fig. 9 Correlation based dimension sorting for the venus dataset. Results of the (**a**) corrgram, (**b**) distribution view, and (**c**) PCP with dendrogram based k-means clustering. The *red color* in (**c**) highlights the major trends shown in (**b**)

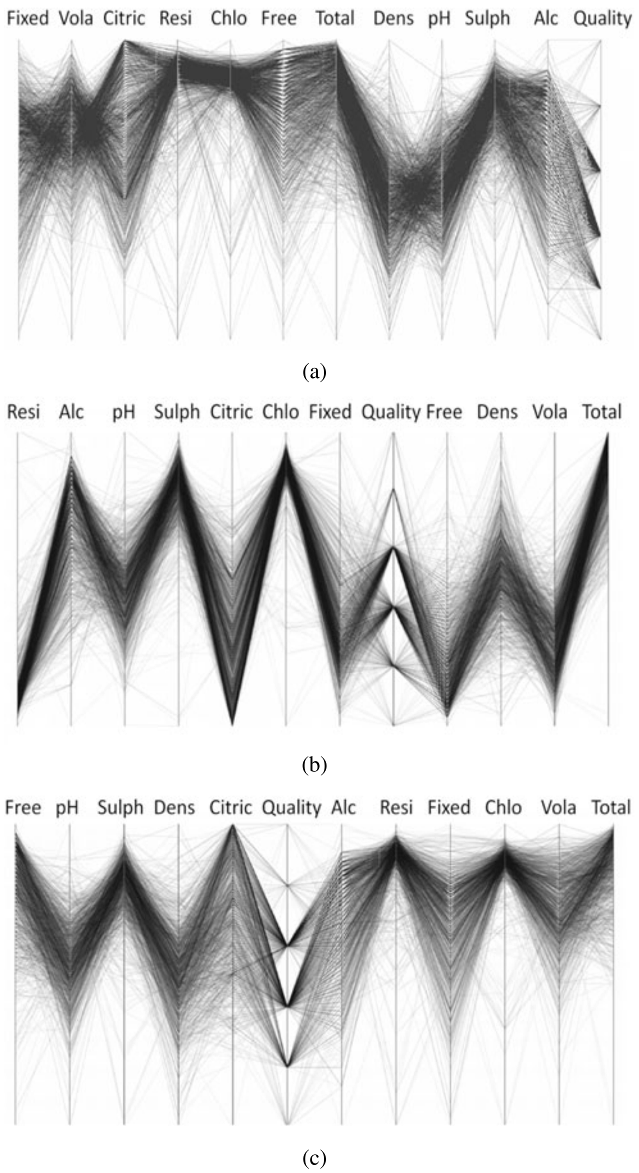
clustering design with both k-means and our modified dendrogram based k-means clusterings, using CE and VI. Our spectral clustering design (1) has good external cluster validation (comparing the clustering result to its corresponding true clustering) and nice stability based cluster validation (the stability of each cluster); and (2) keeps scale invariant with less outlier effects.

4.2 Experimental results of dimension sorting

Dimension sorting methods are built on the optimization of various constraints, which naturally guarantee the best result under certain conditions with specified constraints.

Correlation based dimension sorting We first demonstrate the ability of our framework for the visual representation of large and complex dataset by using the 8784-item, five dimensional (Latitude, Longitude, Velocity, Density, and Temperature) venus dataset. The corrgram (Fig. 9a) clearly shows that (1) Latitude and Longitude are less correlated because of the random sampling; (2) Velocity, Density, and Temperature form a relatively homogeneous grouping with high positive correlations, while Velocity and Temperature have the highest correlation among all the correlations; and (3) only Density has a weak positive correlation with the position especially with Latitude. The quantization view of the distribution of polylines (Fig. 9b) shows that our optimal correlation based dimension sorting method reveals some clear bands generated by the aggregation of polylines, which an arbitrary ordering of coordinates fails to reveal. The clustering view in PCP (Fig. 9c) further supports the observation of major trends, and reveals geographical sampling positions.

View-range matrix based dimension sorting We test our design for the wine dataset, which contains a total of 4898 items. For each sample, there are a total of 11 physiochemical features (Fixed acidity, Volatile acidity, Citric acid, Sugar residue, Chlorides, Free sulfur dioxide, Total sulfur dioxide, Density, pH, Sulphates, Alcohol) along with the Quality. By comparison with original dataset (Fig. 10a), some intriguing relations are shown with highly gathered polylines (between the most relevant factors) using our design. As shown in Fig. 10b, using the view-range matrix with a user selected histogram number, it is clear that alcohol can be increased or decreased by monitoring the sugar concentration, and an increase in Sulphates may be related to the generation of Citric acid and pH, which is a very important factor to improve the wine aroma. Some strong correlations are directly detected from the view-range matrix with the pixel-level, as shown in Fig. 10c. For instance, an increase in the Alcohol tends to result in a higher quality wine based on the Quality and Alcohol correlation, and the residual sugar in wine could be controlled by adjusting the sugar fermentation environment (carried out by yeasts) following Sugar and Fixed acidity correlation. All the explored relevant-factors or correlations can be used to improve the wine quality.



(a)

(b)

(c)

Fig. 10 Results of dimension sorting using the view-range matrix. (**a**) The original wine dataset. (**b**) The dimension sorting result using the view-range matrix with a user selected histogram number (coarse level). (**c**) The dimension sorting result using the view-range matrix with the pixel-level (fine level, each bin has a single item point of the polyline)

Spectral clustering based dimension sorting For a five dimensional (MPG, Cylinders, Horsepower, Weight, and Acceleration) car dataset with 392-item, we apply the spectral clustering based dimension sorting to accurately reveal and clearly visualize major trends of clusters. Figure 11b is the optimization result of spectral clustering based dimension

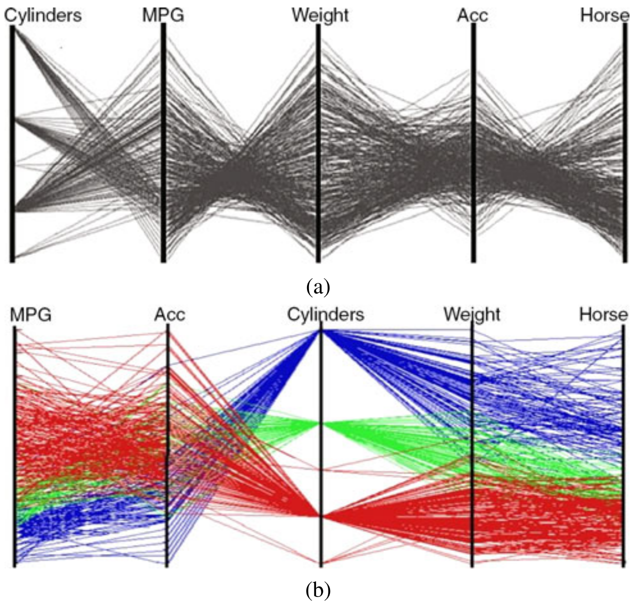


Fig. 11 PCP views of the car dataset using (**a**) a random ordering and (**b**) our spectral clustering based dimension sorting. Each cluster is specifically colored

sorting in PCP, which clearly shows three major clusters. By comparison with a random ordering result (Fig. 11a), our method greatly reduces the clutter and gives the user an immediate sense of correlations: (1) Cylinders, Weight, and Horsepower are most positively correlated, because a heavier car would have a larger engine providing considerable housepower; (2) Acceleration and Cylinders have strong negative correlations only for high values of Cylinders, and little or no correlations for lower values; and (3) MPG has weak positive correlations with Acceleration. In addition, the spectral clustering can also give the user a clear view of distributions and sizes of each cluster (e.g., the red cluster has the largest data items).

All the experimental results show the usability of our framework, with the new defined clustering and dimension ordering approaches, for the accurate and efficient exploration and analysis of various datasets.

5 Conclusions and future work

In this paper, we have proposed several novel clustering and dimension sorting techniques in the format of parallel coordinates, for conveying meaningful aspect/features of multi-dimensional data. Our framework improves the ability of structure revealing in PCP, through the clustering of polylines, optimization of dimension sorting and visual representation. For the clustering, a dendrogram based k-means clustering can automatically select the best cluster number and reach a consistent clustering result. A new spectrum based clustering method is an alternative way to show meaningful and continuous trend/cluster patterns across all axes. Using it, an analyst can easily manipulate each cluster and obtain an optimal bundle distribution view of the data across all coordinates. We propose the optimization of dimension sorting using various metrics to minimize the visual clutter. Therefore, similar variables are positioned adjacently while dissimilar ones are separated, making patterns of interest reveal. These techniques can efficiently and accurately assist the user to distinguish reveal hidden structures in PCP, especially for complex cases. All design methods have fast and easy implementations. Final results demonstrate the visual abilities and merits of our framework.

Further work includes several improvements of our framework: (1) GPU and parallel acceleration for computationally costly steps, such as matrix calculation, because we only calculate pair of axes; and (2) more user-friendly interface widgets to assist the customized design. In addition, appropriated case studies may further generate new research directions.

Acknowledgements This paper has been supported by NSF grants IIS0916235, CCF0702699, and CNS0959979.

References

1. Artero, A.O., De Oliveira, M., Levkowitz, H.: Uncovering clusters in crowded parallel coordinates visualizations. In: IEEE Symposium on Information Visualization, pp. 81–88 (2004)
2. Bach, F., Jordan, M.: Learning spectral clustering. Adv. Neural Inf. Process. Syst. **16**, 305–312 (2004)
3. Chung, F.: Spectral graph theory. In: Conference Board of the Mathematical Sciences, pp. 88–95 (1997)
4. Dasgupta, A., Kosara, R.: Pargnostics: screen-space metrics for parallel coordinates. IEEE Trans. Vis. Comput. Graph. **16**(6), 1017–1026 (2010)
5. Donath, W.E., Hoffman, A.J.: Lower bounds for the partitioning of graphs. IBM J. Res. Dev. **17**, 420–425 (1973)
6. Dubes, R.C., Jain, A.K.: Algorithms for Clustering Data. Prentice Hall, New York (1988)
7. Fiedler, M.: Algebraic connectivity of graphs. Czechoslov. Math. J., 298–305 (1973)
8. Friendly, M.: Corrgrams: exploratory displays for correlation matrices. Am. Stat., 316–324 (2002)
9. Friendly, M., Kwan, E.: Effect ordering for data displays. Comput. Stat. Data Anal. **37**, 47–53 (2002)
10. Fua, Y., Ware, M.O., Rundensteiner, E.A.: Hierarchical parallel coordinates for exploration of large datasets. IEEE Vis., 43–50 (1999)
11. Fua, Y., Ware, M.O., Rundensteiner, E.A.: Structure-based brushes: a mechanism for navigating hierarchically organized data and information spaces. IEEE Trans. Vis. Comput. Graph. **6**(2), 150–159 (2000)
12. Hagen, L., Kahng, A.: New spectral methods for ratio cut partitioning and clustering. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **11**(9), 1074–1085 (1992)
13. Hauser, H., Ledermann, F., Doleisch, H.: Angular brushing of extended parallel coordinates. In: IEEE Symposium on Information Visualization, pp. 127–131 (2002)

14. Inselberg, A.: Parallel Coordinates: Visual Multidimensional Geometry and Its Applications. Springer, New York (2009)
15. Inselberg, A., Dimsdale, B.: Parallel coordinates: a tool for visualizing multidimensional geometry. IEEE Vis., 361–378 (1990)
16. Johansson, J., Ljung, P., Jern, M., Cooper, M.: Revealing structure within clustered parallel coordinates displays. In: IEEE Symposium on Information Visualization, pp. 125–132 (2005)
17. Johansson, J., Ljung, P., Cooper, M.: Depth cues and density in temporal parallel coordinates. Comput. Graph. Forum, 35–42 (2007)
18. Keim, D.: Designing pixel-oriented visualization techniques: theory and applications. IEEE Trans. Vis. Comput. Graph. **6**, 59–78 (2000)
19. Mcdonnell, K.T., Mueller, K.: Illustrative parallel coordinates. Comput. Graph. Forum, 1031–1038 (2008)
20. Meila, M.: Comparing clusterings by the variation of information. In: Proceedings of the 16th Annual Conference on Computational Learning Theory, pp. 173–187 (2003)
21. Novotny, M.: Visually effective information visualization of large data. In: Central European Seminar on Computer Graphics (2004)
22. Novotny, M., Hauser, H.: Outlier-preserving focus+ context visualization in parallel coordinates. IEEE Trans. Vis. Comput. Graph. **12**(5), 893–900 (2006)
23. Peng, W., Ward, M.O., Rundensteiner, E.A.: Clutter reduction in multi-dimensional data visualization using dimension reordering. In: IEEE Symposium on Information Visualization, pp. 89–96 (2004)
24. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
25. The homepage of xmdvtool–multivariate data visualization tool. www.davis.wpi.edu/~xmdv/index.html (2012)
26. Wegman, E.J., Luo, Q.: High dimensional clustering using parallel coordinates and the grand tour. Comput. Sci. Stat. **28**, 352–360 (1997)
27. Zhou, H., Yuan, X., Qu, H., Cui, W., Chen, B.: Visual clustering in parallel coordinates. Comput. Graph. Forum **27**(3), 1047–1054 (2008)

Xin Zhao is a Ph.D. candidate and Research Assistant in the Department of Computer Science, Stony Brook University, Stony Brook, New York. Her research interests are visualization and computer graphics. She received B.S. in computer science from Zhongshan University, China (2007).

Arie Kaufman is a Distinguished Professor and Chair of the Computer Science Department, Chief Scientist of the Center of Excellence in Wireless and Information Technology (CEWIT), and the Director of the Center for Visual Computing (CVC) at Stony Brook University (aka State University of New York at Stony Brook). He is an IEEE Fellow, an ACM Fellow, and the recipient of IEEE Visualization Career Award (2005). He further received the IEEE Outstanding Contribution Award (1995), ACM Service Award (1998), IEEE CS Meritorious Service Award (1999), member of the European Academy of Sciences (since 2002), State of New York Entrepreneur Award (2002), IEEE Harold Wheeler Award (2004), and State of New York Innovative Research Award (2005). Kaufman was the founding Editor-in-Chief of *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 1995–1998. He has been the co-founder, papers/program co-chair, and member of the steering committee of *IEEE Visualization Conferences*; co-founder/chair of *Volume Graphics Workshops*; co-chair for *Eurographics/SIGGRAPH Graphics Hardware Workshops*, and the Papers/Program co-chair for *ACM Volume Visualization Symposia*. He previously chaired and is currently a director of IEEE CS Technical Committee on Visualization and Graphics. Kaufman has conducted research and consulted for over 40 years specializing in volume visualization, graphics architectures, algorithms, and languages, virtual reality, user interfaces, medical imaging, and their applications. He received a B.S. (1969) in Mathematics and Physics from the Hebrew University of Jerusalem, Israel, an MS (1973) in Computer Science from the Weizmann Institute of Science, Rehovot, Israel, and a Ph.D. (1977) in Computer Science from the Ben-Gurion University, Israel. For more information, see http://www.cs.sunysb.edu/~ari.