Interactive Local Clustering Operations for High Dimensional Data in Parallel Coordinates

Peihong Guo He Xiao Zuchao Wang Xiaoru Yuan*

Key Laboratory of Machine Perception (Ministry of Education), and School of EECS
Peking University, Beijing, P.R. China.

ABSTRACT

In this paper, we propose an approach of clustering data in parallel coordinates through interactive local operations. Different from many other methods in which clustering is globally applied to the whole dataset, our interactive scheme allows users to directly apply attractive and repulsive operators at regions of interests, taking advantages of an electricity interaction metaphor, for clutter reduction and cluster detection. Our design enables users to interact directly with the parallel coordinate plots and provides great flexibility in exploring and revealing underlying patterns. With instant feedback, our work allows users to dynamically adjust the clustering parameters to reach an optimum. We also supply the user with a graph indicating the logical relationship between clusters. Our experiments show that our scheme is more efficient than traditional methods in performing visual analysis tasks.

Keywords: parallel coordinates, high-dimensional data, clustering, local clustering operation, user interaction.

Index Terms: I.3.6 [Computer Graphics]: Methodology and techniques—Interaction Techniques; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces (GUI)

1 INTRODUCTION

Innovation in information technology and computing science is generating data with unprecedentedly large sizes and high dimensionality. With faster creation, collection and dissemination, overwhelming availability of data demands better algorithms and tools for visually conveying information into forms with higher comprehensibility.

Among many existing techniques for exploratory visualization of multidimensional datasets, parallel coordinates have been widely applied and studied. Introduced by Inselberg and Dimsdale [8, 10, 18, 9, 7] almost twenty years ago, parallel coordinate plots turn a large multidimensional dataset into a compact two-dimensional visual representation by displaying an *N*-dimensional data tuple as one polyline crossing parallel axes of each dimension. Data clutter in parallel coordinates is a severe problem due to line overdrawing on limited screen space. Due to the clutter effect and interference with crossing lines, the operation of data selection or clustering on parallel coordinates becomes a challenging problem when the data density is high.

In this work, we developed a scheme of using local operators to reduce clutter and detect clusters in parallel coordinates interactively. One of the difficulties in parallel coordinates that reduces the effectiveness of detecting underlying patterns in large data sets is the edge cluttering phenomenon as mentioned above, which occurs when too many edges lie in a limited screen region. Many techniques have been proposed to cluster the data and thus reduce visual cluttering [4, 2, 12, 14, 24]; however, most techniques are either automatic or semi-automatic and users are usually excluded from the course of visual exploration in the sense that they are not actively engaged in the identification of clusters. Another substantial problem is that these techniques can not always generate satisfactory clusters, so in these cases users probably wish to make some refinements to the clustering results. Although most existing techniques provide some adjustable parameters for tuning clustering results, they still lack the flexibility of identifying clusters as users wish. A similar problem exists in graph visualization. As the number of nodes gradually goes up, edge cluttering progressively interferes with users' exploration and interpretation of the graph. Wong *et. al.* suggested introducing EdgeLens [19] to improve the visual quality of complicated graphs by bending the edges in graphs with a magnet metaphor. However, the method of EdgeLens mainly focuses on graph visualization and does not fit well in the situation of parallel coordinate visualization. A recent method proposed by Yuan *et. al* integrates point representations [23] into parallel coordinates to tackle this problem, taking advantage of point representation's highly economic utilization of screen space.

To deal with the above mentioned problem, we propose to introduce some interactive operators to interact with parallel coordinates. By interacting with line segments in parallel coordinates, these operators enable the users to identify and interact freely with clusters in the dataset. The operator is a metaphor to point charges with either positive or negative electricity, and line segments in parallel coordinates are treated as positively charged elastic threads. The lines and the operators interact according to physics laws: operators with negative electricity attract the lines while positively charged operators repulse the lines. To extend this metaphor, we defined the operator as an attractive and repulsive operator which interacts with such lines that have properties specified by users. Further equipped with properties such as main axis direction, effective range and others, these operators perform very well in revealing clusters from highly cluttered parallel coordinates. Overall, we allow the user to directly operate on the parallel coordinates and make the line clustering more efficiently.

The specific benefits of our design and the contribution of this research are as follows:

- *Local Operation*: Since the operation is limited to a local area, it gives more flexibility to the clustering design. The user can focus only on the desired local region and find more interesting data clusters. The selection of the local focus can be aided by the user's knowledge. Other benefits of local operation include less computation and less complexities of the clustering algorithms.

- *Active user's role*: In our design, the user takes a more active role. He/she can interactively explore the data in the parallel coordinate plots by moving the operators and changing the parameters. The user is also able to decide specific areas that need further exploration on the basis of his/her observation.

*{peihong.guo,xiaohe.pku,xiaoru.yuan}@pku.edu.cn

The remainder of this paper is organized as follows. Section 2 provides a review of related works. An overview of our proposed visualization system with local operators is presented in Section 3, followed by details on the attractive and repulsive operators in Section 4 and user interface design in Section 5. After showing case experiments and performance data described in Section 6, conclusions and future work are presented in Section 7.

2 Related Works

Parallel coordinates [8, 10, 18, 7]) have been widely applied to visualize multidimensional datasets. A thorough coverage on the topic of parallel coordinates can be found in Inselberg's newly published book [9].

Many techniques have been developed to facilitate data analysis through parallel coordinates. The major exploration task in parallel coordinates is to find data clusters with certain common properties. On the other hand, one major challenge in parallel coordinates is the cluttering problem. Overlapped polylines can quickly saturate the display space and hide patterns in the dataset under investigation. The cluttering issue is highly related to the clustering task in parallel coordinates. In parallel coordinates, patterns are often difficult to detect due to the line cluttering. Solutions successfully remedying the cluttering problem can help find meaningful patterns in the parallel coordinates. Effective clustering algorithms can reduce the plot cluttering and alter the drawing of lines according to their clustering classification.

Clustering Many clustering methods have been developed for parallel coordinates. Fua et al. [4] constructed a hierarchical clustering method based on the measures of proximity between pairs of objects with a tree of nested clusters. With hierarchical clustering, a multi-resolutional view of the data can be achieved by dynamically selecting focus regions and levels of details. More recently, Zhou et al. [25] proposed geometry-based visual clustering to implicitly enhance the clustering in parallel coordinates by bundling the edges, and minimizing the edge curvatures and maximizing the parallelism of adjacent edges at the same time. When edges are bundled together, clutter can be reduced simultaneously. Further, clusters can be detected by superimposing semitransparent line segments on the screen to enhance important components [24]. Johansson et al. [12] used high-precision textures together with transfer functions to reveal structures within clustered parallel coordinates and highlight different aspects of the cluster characteristics. Illustrative rendering style, turning clustered polylines into continuous color stripes can also reduce the clutter and increase comprehensibility for parallel coordinates [13].

Clutter Reduction As parallel coordinates is prone to the clutter effect, much effort has been devoted to reducing the clutter for better data exploration. The cluttering depends on the alignment of the line segments between two adjacent axes. Reordering dimensions based on similarity can help minimize visual clutter [1, 15]. In addition to dimension ordering, Yang et al. [21] suggested dimension spacing and filtering to reduce clutter and explore high dimensional datasets with parallel coordinates. Artero et al. [2] reduced non-important information in parallel coordinates based on the computed frequency and density plots from the original datasets. Ellis and Dix [3] suggested random sampling as one effective way to visualize cluttered regions. The screen space quality method [11] reduces clutter, while preserves the significant features in the original datasets at the same time, by filtering out data items based on distance transformation for data abstraction.

Interaction Interactive exploration approaches are designed to navigate data both globally and locally to discover interesting patterns. Brushing is one of the most commonly applied interaction techniques in parallel coordinates. It has been designed as an effective tool to select or set focus on polyline groups with specific trends or properties between neighboring axes. Wang and Bergeron [20] designed wavelet brushing for browsing very large multidimensional multivariate datasets by introducing multiple layers of multiresolution approximations into Ward's multidimensional data brush [17]. Multidimensional brushing displays the brushed and non-brushed data at different resolutions. Siirtola [16] suggested direct manipulation of parallel coordinates through polyline averaging and showing correlation coefficients of polyline subsets to dynamically summarize a set of polylines and reveal data properties. Hauser et al. [6] proposed angular brushing of extended parallel coordinates to highlight data correlation between two neighboring axes by interactively specifying a subset of polylines with a certain range of slopes. Transfer functions can also be customized to dynamically highlight different aspects of the cluster data characteristics [12]. Novotny and Hauser [14] recently proposed an outlier-preserving focus+context algorithm which is based on 2-dimensional binning for each pair of adjacent axes. Sampling lens [3] techniques can be used to facilitate focus+context viewing of clutter regions.

It is clear that the tasks of clustering and clutter reduction can not be separated. Clustering with proper rendering enhancement can greatly reduce the line overlap in parallel coordinate plots; good clutter reduction techniques can help the discovery of important clusters in the datasets. In addition, interactivity is extremely important for data exploration and to reveal features. In this paper, we propose an interactive method of clustering that can also help reduce clutter during the exploration.

3 Overview

The approach we propose in this paper is different from many other approaches in that it allows users to locally focus on certain regions and specify clustering parameters dynamically. To enable such operations, two types of local operators are defined. The first type is the attractive operator. An attractive operator allows the user to select a location, define the strength of the attractive force and drag neighboring parallel coordinate lines to the operator center. Note the attraction does not apply to the whole dataset, but only to a local range specified by the user. The attraction force is also anisotropic. Only lines aligned to the defined directions are under clustering computation. The attractive operator can automatically cluster a subset of the data according to the user specification. At the same time, lines are bent to the operator center and relieve clutter in the selected region. The second type is the repulsive operator. The repulsive operator doesn't perform clustering. A repulsive operator blends lines outward from the operator center which is similar to the magic lens. This type of operator is mainly used for reducing cluttering. In the following part of the paper, we mainly focus on the attractive operator. The clustering results deliver feedback to the user immediately to allow direct manipulation.

The overview of the whole approach is illustrated in Figure 1 (a). Given a multidimensional data, the interface of the system allows the user to select regions to explore. In the left part of user interface, i.e. the parallel coordinate plot region, as shown in Figure 1 (b), the user can interact with the plots by defining attractive or repulsive operators with different locations and parameters through mouse interaction. The generated cluster(s) are then highlighted by the user's specific colors. At the same time, in the right part of the interface, the cluster graph, new graph nodes corresponding to the newly explored clusters are shown and the logical relationship among them is revealed by the graph node connection. Users can also directly operate on the cluster graph to manipulate the clustering results. In this part of the user interaction, highlighting, colorizing, unioning and deleting can be applied to the graph nodes. While the resulting graph and parallel coordinates with clusters undergoes user's analysis, the feedback given in realtime can result in new user exploration opportunities. Such interaction and manipulation can be repeated until a satisfactory clustering scheme is reached.

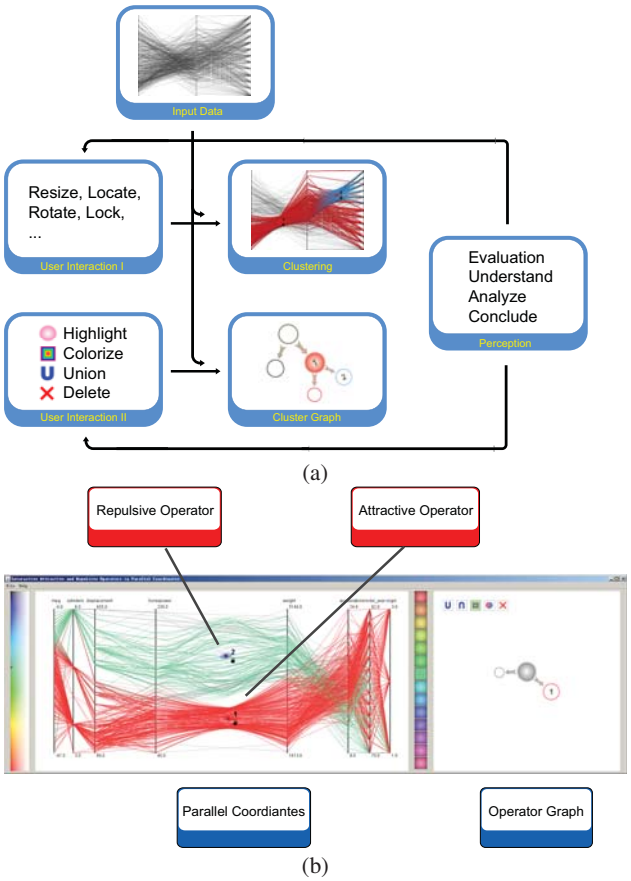


Figure 1: Overview of interactive local operation in parallel coordinates. (a) Flowchart of visual exploration with local operators in parallel coordinates; (b) repulsive and attractive operators in the parallel coordinate plot region together with the corresponding cluster graph region.

4 Attractive and Repulsive Operators

The local operators in our work is inspired by the physics laws between negative and positively charged items. Ji et al. [22] used a visual metaphor of dust and magnet for multivariate information visualization by interacting the data in the point format. In our case, the interaction is between the lines and more sophisticated clustering computation is included.

In our system, each operator, whether attractive or repulsive, is represented with an ellipse with an effective range and a main direction. Moreover, an angular tolerance and an interaction strength is assigned to each operator. Line segments lying in the effective radius and within the angular tolerance of a specific operator will be affected, either attracted or repulsed relative to the operator center. Nearer lines and lines with smaller angular difference towards the operator's main direction will strongly interact with the operator, while faraway lines or lines with a large angular difference will respond weakly. Lines lying outside of the operator's angular tolerance or its effective range will not be affected at all. By such setup, the operation is always limited to a local region. Only part of the dataset is under computation to improve the overall system performance.

In our scheme, users are free to put the operators anywhere in a parallel coordinate plot manually, and neighboring lines will be either attracted or repulsed according to our interaction model. Users are also able to move the operators around and pin the operators to

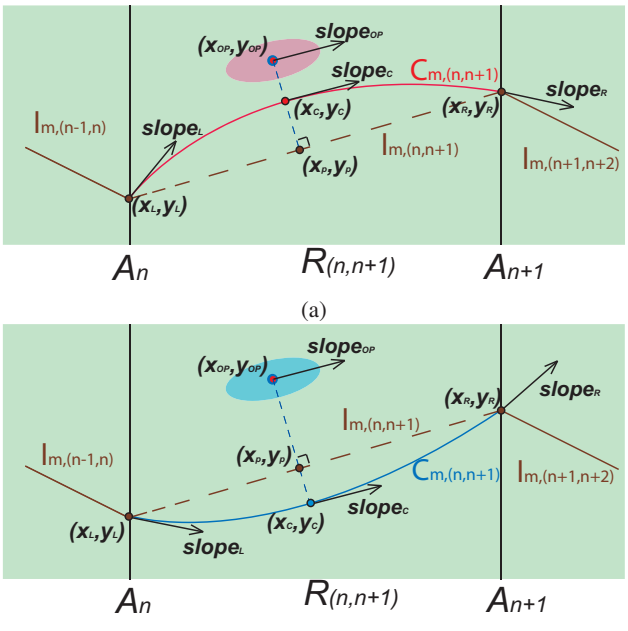


Figure 2: (a) Attractive and (b)repulsive operators. Point (x_c,y_c), (x_L,y_L) and (x_R,y_R), together with $slope_c$, $slope_L$ and $slope_R$, determine the curve $C_{m,n,n+1}$ corresponding to line segment $l_{m,(n,n+1)}$.

bundle affected line segments, which gives users great flexibility to interact with the dataset.

The physical model of interaction between the operator and line or point is as follows: A point charge q is fixed near a uniformly charged thread at a distance of h. Both ends of the thread are fixed. The thread will be either attracted or repulsed by the point charge depending on the electricity on the point charge. The thread will lengthen and bend before it finally reaches an equilibrium as the force exerted on it by the point charge.

However, the interaction between line segments and operator is very complicated and an exact simulation would be too costly in terms of computation. Furthermore, we wish these operators to be applicable to general situations where the interaction between operators and line segments may not be limited to an electrical interaction. Thus the following interaction scheme is taken.

The result of interaction is the formation of curves that represent affected line segments, in other words, simulating the interaction is in fact generating curves according to the information of both operators and related line segments. In our design, the line segments will bend and become curves when the curvature at both ends is changed while the positions remain fixed. Moreover, our scheme provides convenient control over both interaction strength and the resulting curves' shape. Splines are applied to smooth the curve shapes, and in our scheme Hermite splines are adopted as the default setting. In the spline plotting, only one control point is picked up for each line segment. The control point is selected to be between the position of the related operator and its projection on this line segment. Let (x_{op},y_{op}) be the position of the related operator, and (x_p,y_p) is defined as its projection on the line segment. The control point (x_c,y_c) can be determined by

$$\begin{cases} x_c = \alpha x_{op} + (1-\alpha)x_p \\ y_c = \alpha y_{op} + (1-\alpha)y_p \end{cases} \quad (1)$$

where α is the interaction strength factor. The sign of α is positive for attractive operators and negative for repulsive operators. High absolute values of α correspond to strong interaction while

low values correspond to the opposite. Once the control point is determined, the Hermite spline curve connecting both ends of the line segment and the control point in the middle can be generated.

For each curve, we need to specify its curvatures, or slopes, at the two ends of the line segment and the control point in the middle. The slopes have a great effect on the shape of resulting curve. Both smoothness and visual aesthetics must be taken into consideration while choosing the slopes. In our scheme, the slope at the left end, $slope_L$, right end, $slope_R$, and control point, $slope_c$, is chosen as:

$$\begin{cases} slope_L = \beta \cdot \dfrac{y_c - y_L}{x_c - x_L} \\ slope_R = \beta \cdot \dfrac{y_c - y_R}{x_c - x_R} \\ slope_c = slope_{op} \end{cases} \tag{2}$$

where β is the parameter controlling the curve's shape, $slope_{op}$ refers to the slope corresponding to the operator's main direction, while (x_L, y_L) and (x_R, y_R) are the positions of the left end and right end. In our experiments, we found that $1.2 \leq \beta \leq 1.5$ generates the best results.

The curve for each line segment is divided into two parts, the left part and the right part. The left part starts from the left end of the line segment and ends at the control point, while the right part starts from the control point and goes to the right end. For each part, the path of the curve (x_{curve}, y_{curve}) is defined by the following function.

$$\begin{cases} x_{curve} = t \cdot x_A + (1-t) \cdot x_B \\ y_{curve} = y_A \cdot f_A(x_{curve}) + y_B \cdot f_B(x_{curve}) \\ \qquad + slope_A \cdot g_A(x_{curve}) + slope_B \cdot g_B(x_{curve}) \end{cases} \tag{3}$$

where A and B refer to both ends of this part of curve, and $t \in [0,1]$. Base functions $f_A(x)$, $f_B(x)$, $g_A(x)$ and $g_B(x)$ are defined as follows.

$$\begin{cases} f_A(x) = (1 + 2\dfrac{x - x_A}{x_B - x_A})(\dfrac{x - x_B}{x_A - x_B})^2 \\ f_B(x) = (1 + 2\dfrac{x - x_B}{x_A - x_B})(\dfrac{x - x_A}{x_B - x_A})^2 \\ g_A(x) = (x - x_A)(\dfrac{x - x_B}{x_A - x_B})^2 \\ g_B(x) = (x - x_B)(\dfrac{x - x_A}{x_B - x_A})^2 \end{cases} \tag{4}$$

Our proposed scheme is illustrated in Figure 2.

As is described above, two parameters, α (see Equation 2) and β (see Equation 1), are provided for users to control both interaction strength and the resulting curves' shape. Since these parameters can be set to a value of the users choice, users can easily adapt this scheme to meet their demands by correlating α and β to some key properties in their specific situations.

In our implementation, we also feature both anisotropic and distance-dependent interaction. To achieve this goal, we formulate α as follows:

$$\alpha = exp(-\frac{d}{IR_{op}}) \cdot S_A \cdot F_S \tag{5}$$

where d is the distance between the operator and the related line segment, and IR_{op} is the effective radius of the operator. Angular similarity S_A, in inverse proportion to the angular difference, has a range of $[0,1]$, and represents angular similarity between the line segment and the operator which is unique for each line segment. Scale factor F_S has a default value of 1.0, and can be changed by the user to achieve the expected interaction strength. Note that d and S_A both depend on the properties of the line segment and the operator.

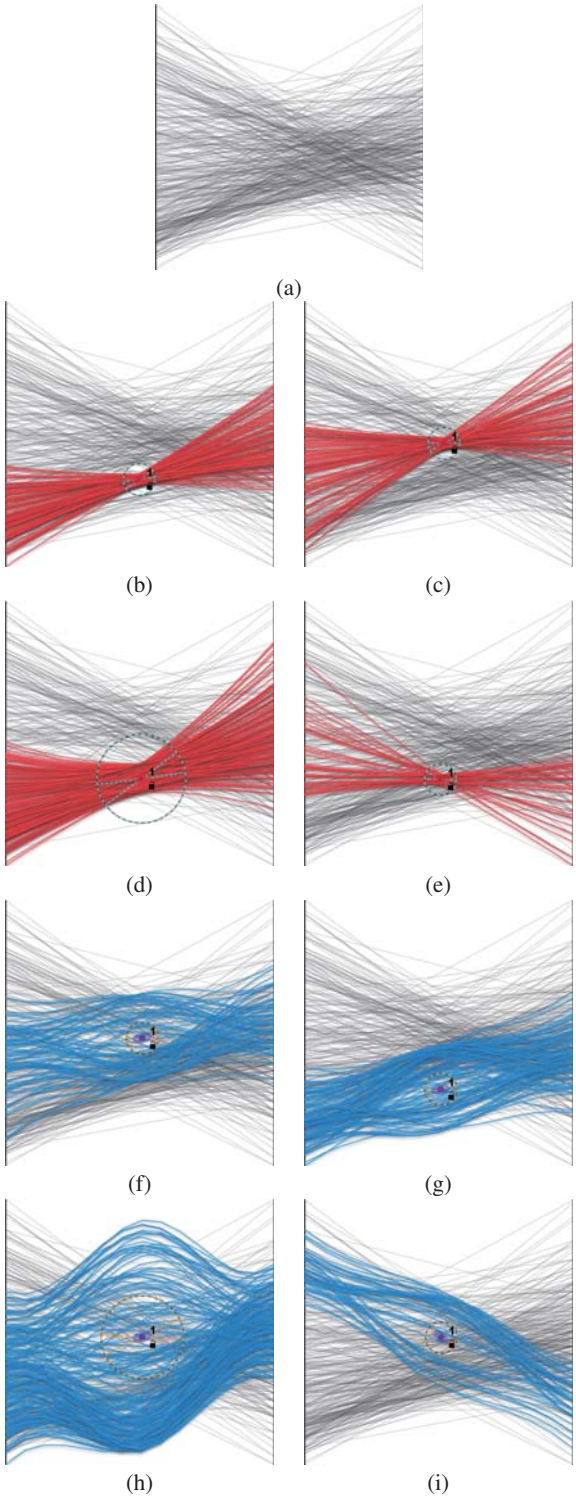


Figure 3: Operator interaction. (a) A segment of parallel coordinate plot under exploration; (b)-(e) user interaction with attractive operators; (f)-(i) user interaction with repulsive operators. [$F_S = 1.0,\ \beta = 1.2$]

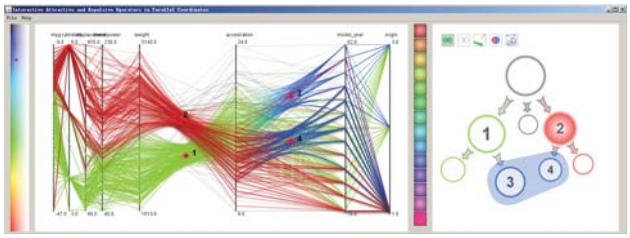


Figure 4: Hierarchical Clustering. The left part of the interface is the parallel coordinate plot. The right part of the interface is the cluster graph showing relationship between clusters defined by the user.

Typically this will result in varying values of α for different line segments and different operators.

5 User Interface Design

Our visualization system has a carefully designed interface to facilitate usability and improve data exploration performance.

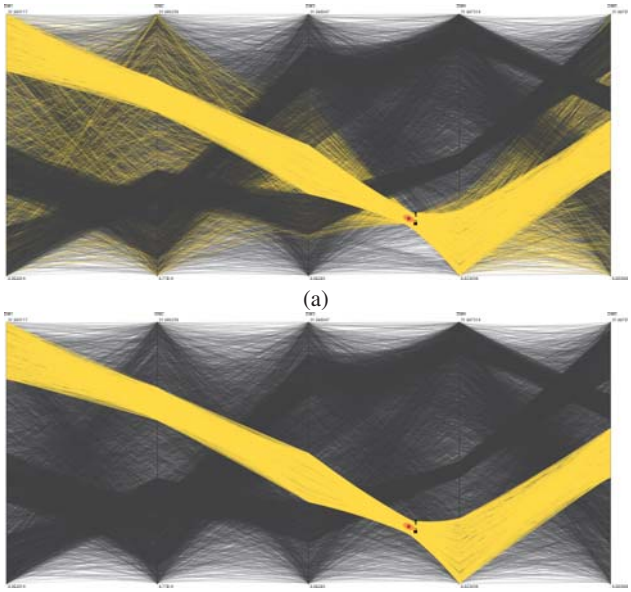
We design manipulations on the operator parameters that can be directly applied through mouse or keyboard interaction without resorting to numerical parameter input. As shown in Figure 3, many interactions can be applied. After defining an initial input of the operators as shown in Figure 3(b), the location of the operator can be directly moved by moving the mouse location (Figure 3(c)). The region of interest can also be resized by scrolling the mousing wheel as shown in Figure 3(d). Since the attractive operator is anisotropic in our work, only features with similar directions will be affected by the operator. The changing of the operator orientation is demonstrated in Figure 3(e).

Similar effects can be achieved for the repulsive operation. Operator center movement, region resizing, and orientation rotation are shown in Figure 3(g), (h) and (i) respectively. With the convenient input to adjust the operator parameters, our system help the user conduct data exploration smoothly and achieve desirable clustering results.

5.1 Hierarchical Clustering

It is common in data clustering that one cluster can be further classified into multiple sub-clusters and form a hierarchical structure. To facilitate such hierarchical clustering, we provide a dual view of user selection on clustering by showing a graph layout of the clusters.

As shown in Figure 4, in the parallel coordinate plot region, the user can hierarchically cluster the data under investigation. After the user identifies a cluster (cluster 1 in the figure, in green color), he/she can further allocate a subset of the cluster with another operator applied on the region which has been selected and obtain a new cluster (cluster 3 in blue color). At the same time, the graph on the right side shows the relationship between the clusters the user has explored. The size of each node indicates the number of items in each corresponding cluster. The color of each node is the same as that of the corresponding polyline cluster in the parallel coordinate plot. Illustrated in Figure 4, the user first identifies two clusters (cluster 1 and 2). Then, the user further applies a local attractive operator to filter out a sub-cluster of each each cluster obtained in the first stage. The cluster graph allows the user to directly manipulate the cluster(s) selected. The cluster can be jointed together or deleted directly in the cluster graph panel. The resulting effect is shown in the parallel coordinate plot area immediately.



(a)

(b)

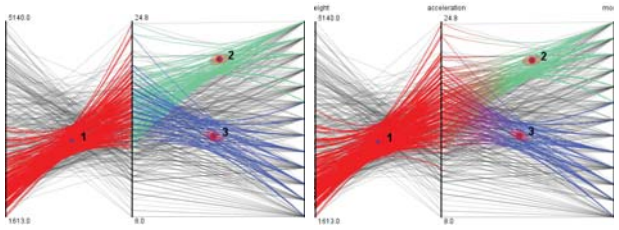
Figure 5: (a) Before optimization; (b) after 4 iterations of optimization with $\gamma = 2.0$.

5.2 Clustering Result Optimization

The locally defined operators provide users with an intuitive way of detecting clusters in the subspace formed by two neighboring dimensions. However, the resulting groups of data items may include some outliers, due to the limited information used during the classification of the dataset. We apply an interactive optimization on the specified raw clusters to remove the outliers thus guarantee more mathematically rigorous clustering results. Once the operator's parameters are specified by the users, the optimization process automatically refines the cluster. The users can manually perform further optimization repeatedly until satisfactory results are reached. The optimization is based on a simple outlier detection technique. Let μ be the mean value vector and σ be the standard deviation vector of the given group of data items. Data item that lies out of the range of $[\mu - \gamma\sigma, \mu + \gamma\sigma]$ are then considered as outlier and are excluded from the group. Note that γ is an adjustable parameter that determines the rigourousness of the optimization process, which has typical values ranging between 1.0 to 3.0. Figure 5 shows the affect on clustering results of the optimization process.

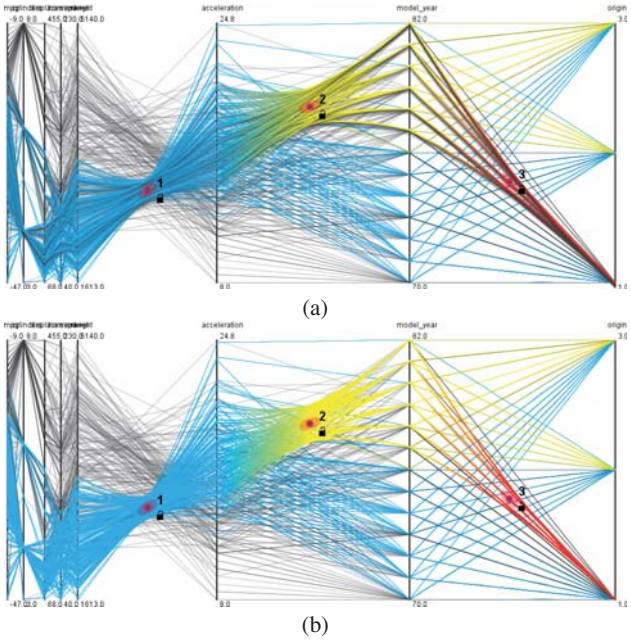
5.3 Color Continuity Design

In the hierarchical clustering, different colors are applied to differentiate clusters. As illustrated in Figure 6(a), it frequently occurrs that one cluster (cluster 1) is further subdivided into multiple subclusters (cluster 2 and 3). In a naive color scheme shown in Figure 6(a), three clusters are encoded with three different colors respectively. However, it is not intuitive to understand the hierarchical relationship among the clusters without careful tracing. We design a continuous color encoding scheme to facilitate the comprehensibility of the visualization for hierarchical clusters. As shown in Figure 6(b), the line color of cluster 1 is extended further to the right, resulting in a smooth color transition between cluster 1 and its subclusters. In this way, it is very easy to discern the hierarchical structures between clusters. Note that the curved edges can also help enable crossed axis tracing [5] in parallel coordinates and can be employed together with the color continuity design.



(a) (b)

Figure 6: Two neighboring parallel coordinate sessions with (a) discontinuous and (b) continuous colors for subgroup color encoding.



(a) (b)

Figure 7: (a) Illustrative drawing style; (b) traditional non-illustrative drawing style.

5.4 Illustrative Enhancement

To enhance the visual effects of the parallel coordinates, we apply illustrative styles to the selected polylines. As shown in the following images in Figure 7, the highlighted lines in Figure 7(a) are more comprehensible. In this case, we applied shadows to each highlighted line in the parallel coordinate plot. The shadows are of constant width and rendered right at the bottom of the curves. With shadows, the lines in the cluster which have similar orientation and that are very difficult to be distinguished (Figure 7(b)) in regular drawing, can discerned much more easily with our illustrative drawing style.

5.5 Clustering on Multi-touch Platform

It is very natural that in a collaborative visualization environment, inputs of the local clustering operation can be from multiple users simultaneously. To demonstrate the benefits of our system interacting with multiple users, we further implement our system onto a self designed and assembled multi-touch workstation which support direct-touch, bimanual, and multi-finger interaction. As shown in Figure 8, several users can interact with the system at the same time. Each user can specify a local operator and explore the region

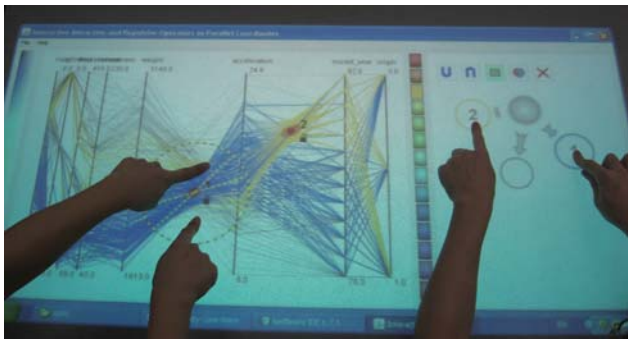


Figure 8: Collaborative visualization with a multi-touch display. Two users are exploring a high dimensional dataset with our interactive local operators and the accompanying cluster graph.

of interest. In the image shown, the user on the left side is manipulating the orientation of an attractive operator. At the same time, another user on the right is operating on the cluster graph and trying to bring two cluster nodes together. In our work, aggregated inputs from multiple users can be collected and displayed on the screen. The cluster graph also shows the clustering progress in the realtime.

6 Case Applications and Performance

In this section, our developed interactive local operators are applied to several widely studied data sets.

Our interactive local operation framework is first tested on a synthesized dataset with 7736 items in five dimensions, consisting of 4 clusters with 876, 752, 608, and 700 items respectively. Another 4800 noisy items are added to the data as illustrated in Figure 9(a). The clustering results with our proposed method are displayed in Figure 9(b)-(e). All four clusters are successfully extracted. As shown in the figures, the user can apply local attractive operators on the parallel coordinate plot, and identify clusters one by one. In Figure 9(b), the first cluster in the five-dimensional dataset is identified and colored in blue. Then the user further finds another cluster and colors it in green. The right panel of the graph shows there are two new nodes emerging from the original single node of the graph, and each node corresponds to one cluster. In Figure 9(d) and (e), two more clusters are detected.

To make a comparison, Figure 9(f) is a clustering result of hierarchical parallel coordinates [4]. Compared with the visual clustering [25] shown in Figure 9(g), our algorithm enables users to clearly extract desirable clusters correctly from other noisy background data items. Note that many outlier data exist in Figure 9(f), while our proposed method can successful avoid the introduction of unnecessary noise through local user defined interaction.

A remotely sensed dataset with 5 variables and 16,384 data items is also investigated by our method as illustrated in Figure 10. Four main clusters have been found through user interaction. Further, in the red colored cluster 1, two sub clusters are separated based on their local properties in the first two dimensions. Similar hierarchical clustering is also performed on the third cluster. The hierarchical clustering result is explicitly illustrated in the right graph panel of Figure 10. In addition, the user-aided exploration process does not only provide users with the flexibility of searching and grouping the data set as needed, but also helps users gain a better understanding of the underlying patterns in the dataset. Clutter is reduced naturally as clusters are identified and pinned by the user. Based on feedback from users, when there are more clusters explored in the data, the cluster graph appears to be more helpful. It also provides a quick shortcut for the user to temporally hide some

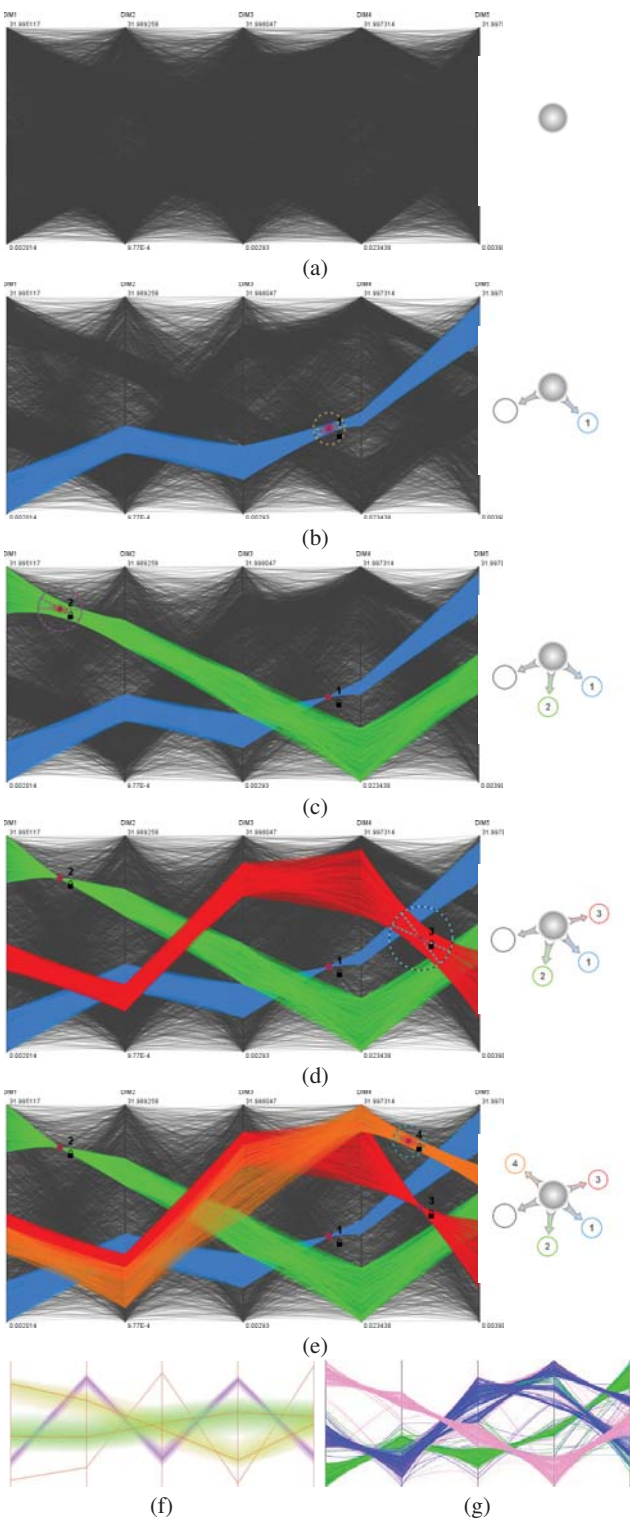


Figure 9: Visual exploration of a synthesized dataset with 7736 five-dimensional items. All 4 clusters are revealed by our methods. (a) Direct visualization; (b)-(e) clusters are detected one by one; (f) clustering by hierarchical parallel coordinates; (g) visual clustering; (g) is courtesy of Zhou et al. [25].

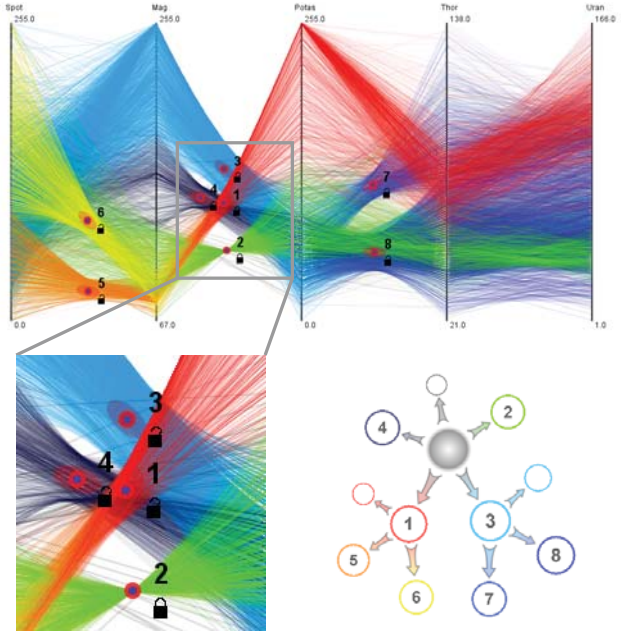


Figure 10: Clustering results of a 5 dimensional data with 16,384 items by our methods. Lower right corner: a graph showing the logical relation between the obtained clusters.

Data	Dim	Size	Cluster Size	Time/Frame	FPS
Car	8	406	143	0.33 ms	3030
			74	0.33 ms	3030
			126	0.27 ms	3703
Syn	5	7736	1243	57.2ms	17.5
			1097	60.6ms	16.5
			1711	50.8ms	19.7
RS	5	16384	3248	231.1ms	4.33
			1562	293.8ms	3.40
			2457	265.4ms	3.77

Table 1: Performance of user defined clustering operations. Syn stands for synthesized data set, and RS stands for the remote sensing data set.

nodes to facilitate new exploration in the data. Note for each cluster, the user can lock it so other operators can not affect lines in the locked cluster anymore. If the cluster is not locked, other clusters may share items with the existing ones.

We also conducted a study to evaluate the user performance with our software. The synthesized dataset illustrated in Figure 9 is used for our test. All together 9 students from different departments at Peking University participated.

Before working on the data, all participants were given a 5 minute introduction on the usage of the visualization tool. The participants were allowed to practice the tool for several minutes on an exercise dataset. Instructions were then given to each student to identify four clusters from the dataset by exploring the data with local operators. The results show that all participants were able to find out all of the four clusters with high accuracy in around 1 minute. The average accuracy of each identified cluster is around 90%. We noticed that the students who tried to resize the operator more frequently normally produced much better results. The accuracy level can reach almost 100%. Without much use of resizing, the partic-

ipants may get much lower accuracy. This indicates our proposed operators are quite promising in performing visual exploration tasks if appropriately used.

We implemented all above visualization algorithms on a Dell Precision T3400 desktop with Intel Core 2 Duo E7400 CPUs and 1GB Memory. The graphics card used is an NVIDIA Geforce GTX 275 with 768 MB DDR2 memory. The implementation is developed in the Java development environment. All visualization operations can be interactively performed. The frame rates of interaction on different datasets are listed in Table 1. Based on our observation, since the operation is mainly done in local environments, the performance is correlated with the number of items in the currently selected cluster. Since our system allows hierarchical data exploration, sub-clustering in an already defined cluster can only compute items of the parent cluster. In this way, much computation can be saved. It indicates that our approach is suitable for exploration on large datasets.

According to the measured timing, we can achieve reasonable frame rates on reasonably sized datasets. Since current implementation is in Java, coding in C/C++ with GPU acceleration can further improve the performance.

7 Conclusion

We developed local clustering operators to facilitate analysis and exploration in parallel coordinates. These operators provide users with great freedom and flexibility on user exploration with parallel coordinate plots. By interacting with line segments in a parallel coordinates plot, users are able to identify and reveal underlying patterns in a dataset efficiently and precisely. Cluttering is reduced naturally with the aid of our proposed operators. In general, our interactive methods can generate satisfactory clustering results compared with several existing clustering methods. The interaction also provides users with a convenient way to dynamically explore complex datasets. Knowledge from the user can also help identify potential clusters in the data under investigation. The accompanying graph provides additional information about relationships between clusters that have been identified. The user can directly conduct operations on clusters by selecting corresponding nodes in the graph. A prototype software of the proposed operators in parallel coordinates is available for downloading at `http://vis.pku.edu.cn/software`.

In the future, we plan to equip the operators with more powerful clustering algorithms, in addition to the geometry based method we currently applied in this paper. Interactivity is the key for effective user defined operations. When more sophisticated algorithms are applied, acceleration methods, such as GPU-based computation, should be considered. We also plan to investigate the effectiveness of our interaction mode under collaborative visualization environments.

Acknowledgements

We thank anonymous reviewers for their suggestive comments. This research is sponsored by National Natural Science Foundation of China Project 60903062, Beijing Natural Science Foundation 4092021, 973 Program (2009CB320903), Key Project of Chinese Ministry of Education (109001), and FSSP 2008109.The project is also supported by the startup funding from the National "985" Project Phase II at Peking University.

References

[1] M. Ankerst, S. Berchtold, and D. A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 52–60, Oct. 1998.

[2] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *Proceedings of IEEE Information Visualization*, pages 81–88, Oct. 2004.

[3] G. Ellis and A. Dix. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):717–724, Sept.-Oct. 2006.

[4] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of IEEE Visualization*, pages 43–50, Oct. 1999.

[5] M. Graham and J. Kennedy. Using curves to enhance parallel coordinate visualisations. In *Proceedings of the 7th International Conference on Information Visualization*, pages 10–16, Jul. 2003.

[6] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 127–130, 2002.

[7] A. Inselberg. Parallax: Software for multidimensional visualization and automatic classification. http://www.kdnuggets.com/software/parallax.html.

[8] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[9] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer New York, 2009.

[10] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of IEEE Visualization*, pages 361–378, Oct. 1990.

[11] J. Johansson and M. Cooper. A screen space quality method for data abstraction. *Computer Graphics Forum*, 27(3):1039–1046, 2008.

[12] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 125–132, Oct. 2005.

[13] K. T. McDonnell and K. Mueller. Illustrative parallel coordinates. *Computer Graphics Forum*, 27(3):1031–1038, 2008.

[14] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, Sept.-Oct. 2006.

[15] W. Peng, M. O. Ward, and E. A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 89–96, Oct. 2004.

[16] H. Siirtola. Direct manipulation of parallel coordinates. In *Proceedings of the 5th International Conference on Information Visualization*, pages 373–378, 2000.

[17] M. Ward. Xmdvtool: integrating multiple methods for visualizing multivariate data. In *Proceedings of the IEEE Conference on Visualization*, pages 326–333, Oct 1994.

[18] E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 411(85):664–675, 1990.

[19] N. Wong, S. Carpendale, and S. Greenberg. Edgelens: an interactive method for managing edge congestion in graphs. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 51–58, Oct. 2003.

[20] P. C. Wong and R. D. Bergeron. Multiresolution multidimensional wavelet brushing. In *Proceedings of IEEE Visualization*, pages 141–148, Nov. 1996.

[21] J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 105–112, Oct. 2003.

[22] J. S. Yi, R. Melton, J. Stasko, and J. A. Jacko. Dust Magnet: multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256, 2005.

[23] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, 15(6):1001–1008, Nov.-Dec. 2009.

[24] H. Zhou, W. Cui, H. Qu, Y. Wu, X. Yuan, and W. Zhou. Splatting the lines in parallel coordinates. *Computer Graphics Forum*, 28(3):759–766, 2009.

[25] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.