

# Visual Clustering in Parallel Coordinates

Hong Zhou<sup>1</sup>, Xiaoru Yuan<sup>2</sup>, Huamin Qu<sup>1</sup>, Weiwei Cui<sup>1</sup>, Baoquan Chen<sup>3</sup>

<sup>1</sup>Computer Science & Engineering Department, The Hong Kong University of Science and Technology, Hong Kong

<sup>2</sup>School of EECS & Key Laboratory of Machine Perception (Ministry of Education), Peking University, China

<sup>3</sup>Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China

---

## Abstract

*Parallel coordinates have been widely applied to visualize high-dimensional and multivariate data, discerning patterns within the data through visual clustering. However, the effectiveness of this technique on large data is reduced by edge clutter. In this paper, we present a novel framework to reduce edge clutter, consequently improving the effectiveness of visual clustering. We exploit curved edges and optimize the arrangement of these curved edges by minimizing their curvature and maximizing the parallelism of adjacent edges. The overall visual clustering is improved by adjusting the shape of the edges while keeping their relative order. The experiments on several representative datasets demonstrate the effectiveness of our approach.*

---

**Keywords:** Parallel coordinates, visual clustering, multivariate data visualization, clutter reduction.

## 1. Introduction

Understanding complex high-dimensional datasets is an important yet challenging problem. Among various techniques developed, parallel coordinates [ID90] have been widely adopted for the visualization of high-dimensional and multivariate datasets. By using parallel axes for dimensions, the parallel coordinates technique can represent N-dimensional data in a 2-dimensional space. Parallel coordinates can be considered as special node link diagrams, in which the nodes are on the parallel axes and the edges are those polylines linking the nodes on two neighboring axes.

Although parallel coordinates have been proven to be an effective tool, edge clutter prevents effective revealing of underlying patterns in large datasets. Many enhancements of parallel coordinates have been developed [ED06, AdL04, HLD02, JLC05]. Clustering is one of the most frequently used methods to reduce the visual clutter and improve the perceptibility of the patterns in multivariate datasets. The main cause of the visual clutter comes from too many polylines. Most of the existing clutter reduction efforts are mainly data centric, *i.e.*, data are clustered before they are plotted. Thus the revealed patterns are limited by the used data clustering algorithm. On the other hand, multidimensional data clustering itself is a very hard problem. Many

methods have been proposed, while most of them are only good for certain kinds of data. In addition, it is difficult for users to control the clustering level through setting parameters without expert knowledge.

Instead of using data-centric clustering, in this paper we propose to perform visual clustering by geometrically deforming and grouping polylines while they are being plotted. Our clustering is achieved by analyzing the geometric relationship between polylines rather than the data itself. An optimization scheme is designed to minimize the curvature of the edges and maximize the parallelism of adjacent edges through an energy function. By adjusting the weight of each term in the energy function, users are capable of controlling the level of visual clustering according to their preferences. Instead of assigning cluster tags to the data items, our visual clustering only groups potential cluster lines into visual bundles and allows users to identify the clusters by themselves. Users can then further explore the clustering results through color and opacity enhancement.

In this paper, we propose visual clustering for parallel coordinates as a novel technique to reduce visual clutter and enhance patterns. Curved lines are used to form visual bundles for clusters in parallel coordinates. Through this method, patterns will become visually better organized and have a higher chance to be revealed. We design an energy function and an optimization framework to perform visual clustering in parallel coordinates. Our energy function is for-

mulated under the intuitive observation that spatially close and visually parallel lines tend to be clustered together. To the best of our knowledge, this is the first time that such an energy formulation has been proposed for parallel coordinates. We further introduce a color and opacity specification tool and some animation schemes to enhance and reveal the patterns while preserving the context. We demonstrate that visual clustering in parallel coordinates has a wide range of applications, especially for datasets with fuzzy clusters or outliers. Compared with previous methods, visual clustering exploits the geometry channel of the visual representation to reveal or enhance the patterns and can thus free up the color and opacity channels for other important information.

## 2. Related Work

Since the early introduction by Inselberg [ID90], parallel coordinates have been widely used to visualize multivariate data. To deal with the visual clutter problem, many schemes have been proposed [ED07].

**Dimension reordering** Dimension ordering, spacing and filtering can improve the parallel coordinates layout and facilitate data exploration. Ankerst et al. [ABK98] clustered data dimensions according to their similarity, then data dimensions are rearranged such that dimensions showing a similar behavior are positioned next to each other. Yang et al. [YPWR03] proposed a hierarchical approach to improve the interactivity of dimension ordering, spacing and filtering. Peng et al. [PWR04] defined a visual clutter measure as the ratio of outlier points to the total data points. The optimized dimension order is then computed to minimize the proposed clutter measure. These methods are very useful, but our paper addresses a different problem, *i.e.*, how to further reduce clutter for parallel coordinates which may already have good dimension ordering and spacing.

**Clustering and filtering** Fua et al. [FWR99] used hierarchical clustering and proposed a variation on parallel coordinates to convey aggregation information for the resulting clusters. In their scheme, proximity-based coloring is used to assure that clusters from similar parts of the hierarchical structure are shown in similar colors. Novotny [Nov04] represented each cluster as a polygonal area and used different opacity values and textures to distinguish different clusters. In [AA04], striped envelope and ellipse plots have been proposed for exploring properties of subsets. Artero et al. [AdL04] tackled the overlapping problem by filtering out the information in parallel coordinates based on frequency or density information computed from the dataset. Ellis and Dix [ED06] used a sampling lens to facilitate focus+context viewing of particular regions. Compared with these clustering and filtering techniques, our method aims at presenting the entire information to users in a visually enhanced way so that users can more easily identify any potential patterns.

**Interaction techniques** To further assist the exploration, navigation tools are designed to support data localiza-

tion and subspace drilling operations. Wong and Bergeron [WB96] developed a wavelet brushing technique for high-dimensional data. Hauser et al. [HLD02] used angular brushing to pick out data subsets with specific trends between adjacent axes. Siirtola [Sii00] directly manipulated parallel coordinates by dynamically summarizing a set of polylines and interactively visualizing the correlation between polyline subsets. These brushing and interactive techniques are considered very effective tools in exploring the structures within the clusters, while our method can further provide an overview of the global structures and reveal the relationships between clusters.

**Visual enhancement** Various visual enhancement schemes have also been proposed for parallel coordinates. Johanson et al. [JLJC05] used high-precision textures and transfer functions to reveal clusters detected by numerical methods. Inspired by their work, we also use transfer functions, but our algorithm first geometrically enhances patterns before applying any color enhancement. Novotny and Hauser [NH06] presented a useful outlier-preserving focus+context visualization tool. Kosara et al. [BKH05, KBH06] developed various extensions to parallel coordinates to interactively explore categorical data. If free-form curves are used [The00], the space between two adjacent axes can be efficiently exploited to encode more information of other axes, which can help detect correlations among more than two dimensions. Graham and Kennedy [GK03] also used smooth curves to allow users to discern individual paths through the curve knots. However our method not only curves lines to reduce visual clutter but also adjusts their shapes to form visual clusters.

Other than the aforementioned methods, clutter reduction has also been applied to other visual representations like graphs [PWR04, WCG03, YPWR03]. For example, Holten [Hol06] bent and bundled edges in graphs for relationship exploration in data with known hierarchical information. This work further demonstrates the potential of using curves to enhance the patterns in the data.

## 3. The Visual Clustering Algorithm

This section describes our visual clustering algorithm for parallel coordinates based on energy minimization. We first introduce our framework to model visual interaction between lines. Then we describe in detail our energy function and the optimization process.

As discussed in Section 2, many researchers have worked on reducing visual clutter through filtering, clustering, and dimension reordering. We take an approach that directly considers the visual interaction among edges. Since it is difficult to directly measure the visual clutter of a given drawing of parallel coordinates by quantitatively estimating human perceptibility, we develop a simplified model to represent the interaction between the individual lines of the parallel coordinates.

We model the parallel coordinates as a system with force interaction between lines, where the force is defined towards reducing visual interference between edges. The status of the system can be described as the energy level of the whole system. By allowing edges to be curved and their shapes to be adjustable, visual clutter can be reduced. After computing the system status with minimized energy, the optimized configuration of the parallel coordinates can be obtained. Unlike previous approaches which use numerical methods to detect clusters, our algorithm achieves visual clustering without data classification.

In our model, the total energy of the edges can be divided into two major terms as represented by the following equation:

$$\mathbf{E} = \alpha_c \mathbf{E}_{curvature} + (1 - \alpha_c) \mathbf{E}_{gravitation} \quad (1)$$

where  $\mathbf{E}$  is the total energy of the whole system,  $\mathbf{E}_{curvature}$  is the energy term representing the bending of each line,  $\mathbf{E}_{gravitation}$  is the energy term representing the interactions between neighboring line pairs,  $\alpha_c$  and  $(1 - \alpha_c)$  are the weighting coefficients of the energy term  $\mathbf{E}_{curvature}$  and  $\mathbf{E}_{gravitation}$  respectively.

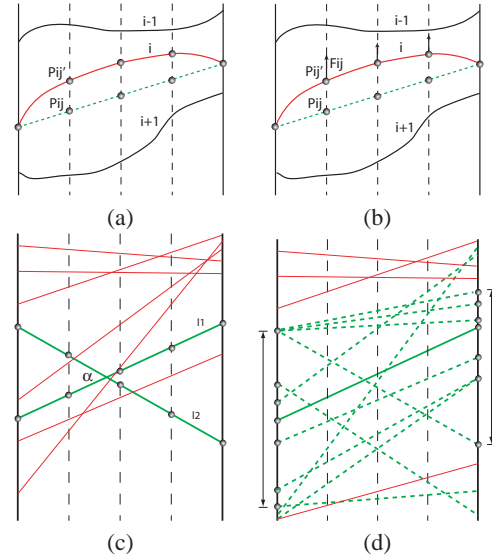
Next, we describe in detail each energy term and its corresponding visual clustering effect we intend to achieve through energy minimization. As a design guideline, we intend to construct a linear system so that it can be solved efficiently through linear programming. Therefore, both energy terms are formulated using linear functions.

### Curvature Energy Term:

In our model, we allow edges to be bent to reduce excessive intersections between lines. We set a curvature energy term that prevents each line from being bent too much. For a given drawing, we assume there are  $n$  data points (lines in the plot). The curvature energy term can be approximated as:

$$\mathbf{E}_{curvature} = \sum_{i=1}^n \sum_{j=1}^m |P'_{ij} - P_{ij}| \quad (2)$$

where  $P'_{ij}$  is a control point on a curved line  $i$  (the red line in Fig. 1(a)). Altogether there are  $m$  control points sampled for each line, and with these control points, the corresponding curve can be drawn by using any well-known curve (e.g., Bezier or Catmull-Rom curves). We can change the curve shape by moving the control points up or down. All control points with the same horizontal position form a column of control points. Point  $P_{ij}$  is the corresponding point on the straight line connecting the same two endpoints that  $P'_{ij}$  is associated with (the green line in Fig. 1(a)). Points  $P'_{ij}$  and  $P_{ij}$  share the same horizontal coordinates. The more bending the curve, the larger the curvature, and the longer the distance between the  $P'_{ij}$  and  $P_{ij}$  pair, therefore, the higher the



**Figure 1: Energy Terms:** (a) Curvature energy term; (b) Gravitation energy term; (c) Computing the force for the gravitation energy term; (d) The range of neighboring lines for gravitation interaction.  $m$  is set to be 3 for Eq. 2 and 3.

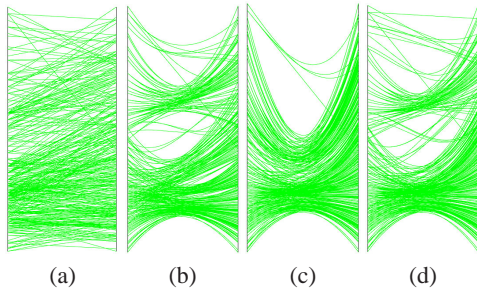
energy contribution from the curve. Note that our modeling of the curvature is a simplified one. It is more like a deviation energy which models the distance between the lines before and after the bending. In our system, we usually set  $m$  to be 2 or 3 which can form edge bundles efficiently without causing many zig-zag curves. Although directly measuring the curvature would be more accurate, it is computationally expensive, and furthermore, will result in a non-linear system.

### Gravitation Energy Term:

We use another term called gravitation energy to model the interaction between neighboring lines. To minimize excessive intersections between lines, it is desirable to have neighboring lines as parallel as possible and parallel lines pulled as close to each other as possible. Based on this intuitive observation together with extensive experiments, we design an empirical gravitation function. For the  $n$  data points (lines in the plot), our function is:

$$\mathbf{E}_{gravitation} = \sum_{i=1}^n \sum_{j=1}^m -F_{ij} \cdot (P'_{ij} - P_{ij}) + E_{ij} \quad (3)$$

where  $F_{ij}$  is the force computed based on the initial state of the neighboring edge arrangement and  $P'_{ij} - P_{ij}$  is the moving distance of the control point  $ij$  along the force direction as illustrated in Fig. 1(b). Note that the negative sign is used to guarantee that the energy will become lower when the control points move along the force directions.  $E_{ij}$  is defined to keep the relative vertical order of control point  $ij$  for



**Figure 2:** The effect of energy term weighting on visual clustering: (a) No visual clustering; (b)  $\alpha_c = 0$ ,  $q_\alpha = q_d = 15$ ; (c)  $\alpha_c = 0$ ,  $q_\alpha = q_d = 30$ ; (d)  $\alpha_c = 0.15$ ,  $q_\alpha = q_d = 30$ .

non-intersecting edges. If point  $ij$  is still “in order”,  $E_{ij} = 0$ , while if point  $ij$  crosses over the corresponding points of the nearest non-intersecting edges,  $E_{ij} = \text{infinite}$ . For each control point  $ij$ , the force  $F_{ij}$  is computed as the summation of its interactions with all the neighboring edges. Assume that we consider  $n'$  neighboring edges for line  $i$ . The force  $F_{ij}$  can then be computed as:

$$\mathbf{F}_{ij} = \sum_{k=1}^{n'} f_{(l_i, l_k)_j} \quad (4)$$

where  $l_i$  and  $l_k$  are the two lines forming a neighboring pair, and  $l_i$  represents the line to which the candidate control point  $ij$  belongs. The force of each line pair  $(l_i, l_k)$  at the  $j$ th sampling point is:

$$f_{(l_i, l_k)_j} = \frac{1}{\alpha_{l_i, l_k}^{q_\alpha}} \cdot \frac{1}{D_{(l_i, l_k)_j}^{q_d}} \quad (5)$$

The  $\alpha_{l_i, l_k}$  is the angle between the two lines, as illustrated in Fig. 1(c).  $D_{(l_i, l_k)_j}$  is the distance between sample point pairs of the two lines.

$$D_{(l_i, l_k)_j} = |P_{ij} - P_{kj}| \quad (6)$$

where  $P_{ij}$  is the  $j$ th sampling point of line  $i$ . Intuitively, this term intends to pull all line pairs as close to each other as possible, and as parallel to each other as possible, although the Curvature Energy Term will counter-balance this term. The exponents  $q_\alpha$  and  $q_d$  control the influence of  $\alpha_{l_i, l_k}$  and  $D_{(l_i, l_k)_j}$  respectively.

When computing the interaction between line pairs, we only consider a subset of neighboring lines for each line to minimize the computational cost. We first include all the lines that intersect with the line under consideration. Then we also consider the neighboring lines formed by  $n'$  nearest endpoints which are closed to each endpoint of the line.

As illustrated in Fig. 1(d), all lines interacting with the solid green line are dotted green lines. Note  $n'$  is 6 in this example. All lines that have no interaction with the green solid line are in red and will not participate in the computation.

We solve the above energy minimization problem with a linear programming solver *lp\_solve* [Sof] based on the revised simplex method and the Branch-and-bound method.

The effect of the visual clustering also depends on the relative weighting between the energy terms  $\mathbf{E}_{\text{curvature}}$  and  $\mathbf{E}_{\text{gravitation}}$ . In Fig. 2, we demonstrate the effect of the weighting factor  $\alpha_c$  on the final clustering. The coefficient  $\alpha_c$  controls the contribution from  $\mathbf{E}_{\text{curvature}}$  which tends to keep each line as straight as possible. Fig. 2(a) shows a plot before applying our energy minimization. From Fig. 2(b) to (d), the value of  $\alpha_c$  varies between 0 to 0.15, while  $q_\alpha$  and  $q_d$  vary between 15 to 30. We observe that without the contribution from the energy term  $\mathbf{E}_{\text{curvature}}$ , the level of clustering can be solely controlled by  $q_\alpha$  and  $q_d$  (Fig. 2(b) to (c)). By increasing  $\alpha_c$ , the clusters of curved edges can be distributed more evenly over the coordinate space, thereby achieving better visual clarity (Fig. 2(c) to (d)).

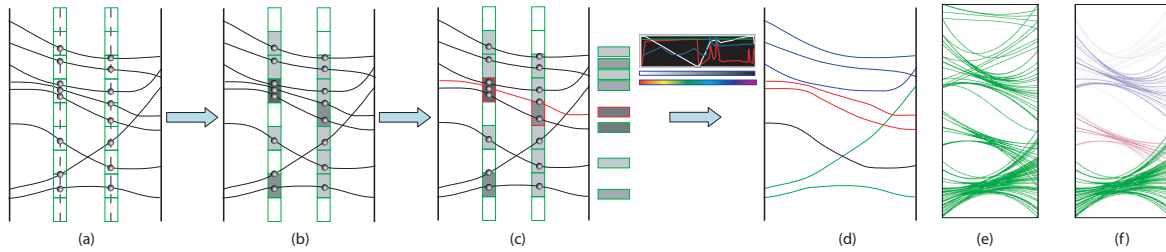
#### 4. Visualization and Interaction Techniques

In this section, we introduce two visualization techniques which can be used together with visual clustering to further increase the expressive power of parallel coordinates.

##### 4.1. Color and Opacity Enhancement

Applying alpha blending to parallel coordinates drawings can highlight different aspects of the data [JLJC05]. To further improve the visual effectiveness, we also assign color and opacity to the plot of our work. After the optimization, lines with similar properties are aggregated to form bundles. We can then assign opacity and color according to the local density of the lines.

The computation of local line density and the assignment of color and opacity are illustrated in Fig. 3. We compute the line density using a histogram method. Each vertical column of the control points is first divided into a number of bins (green rectangles in Fig. 3(a)). The number of bins depends on the total number of lines. We normally set the number of bins to 1/16 or 1/32 the number of lines. Then for each bin, we count the number of control points falling into it (see Fig. 3(b)). The number of points in each bin is then normalized to approximate the bin density. As illustrated in Fig. 3(c), for the red line, the density values of the bins containing the line’s control points are computed. The density values of the up and down two neighboring bins for each control point are also computed. The density value of a given control point is the convolution of the three bin densities with a Gaussian function based on the distance between the bin’s center and the control point. The averaged density



**Figure 3:** Applying color and opacity based on line density: (a) Density bins for each control point column; (b) Accumulation of line density for each bin; (c) Computation of the local density for each line by averaging the density values of all control points; (d) Applying color and opacity based on user specified transfer function; (e) Parallel coordinates before applying color and opacity; (f) Parallel coordinates plot after applying color and opacity.

value of all control points of a line is then used to represent the density of the line.

Afterwards, a user specified transfer function is utilized to map the average density value of each line to color and opacity for rendering. We design an interface similar to the transfer function specification in volume rendering to assign the color and opacity value based on density (Fig. 3(d)). Users can then interactively manipulate the transfer function and selectively enhance line bundles according to their local density values and the histogram curve showing how many lines fall into each density value bin. By assigning different colors to different density values, clusters can be easily differentiated from noises. It is possible that two distinct clusters with similar line density values may be assigned similar colors. But users can easily distinguish them based on their position differences. Fig. 3(e) and (f) show the plots before and after the application of the transfer function.

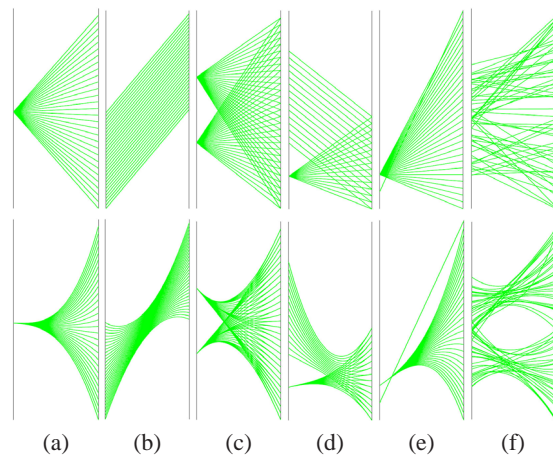
## 4.2. Animation

Animation can be exploited to facilitate the effective use of the visualization by keeping viewers oriented, facilitating clutter reduction [ED07], and enhancing comprehension of state transitions [HR07]. We take the advantage of animation techniques to reveal more information in the visual clustering process. By storing the intermediate results of the optimization process, then interpolating and displaying them consecutively, the animations reveal the interaction of edges and help users identify clusters by walking through the visual clustering process. In such animations, clusters which emerge early and become absorbed in further optimization iterations can be identified and then investigated based on users' requests.

## 5. Experimental Results

In this section, we demonstrate the effectiveness of our visual clustering framework through the experiments on several datasets.

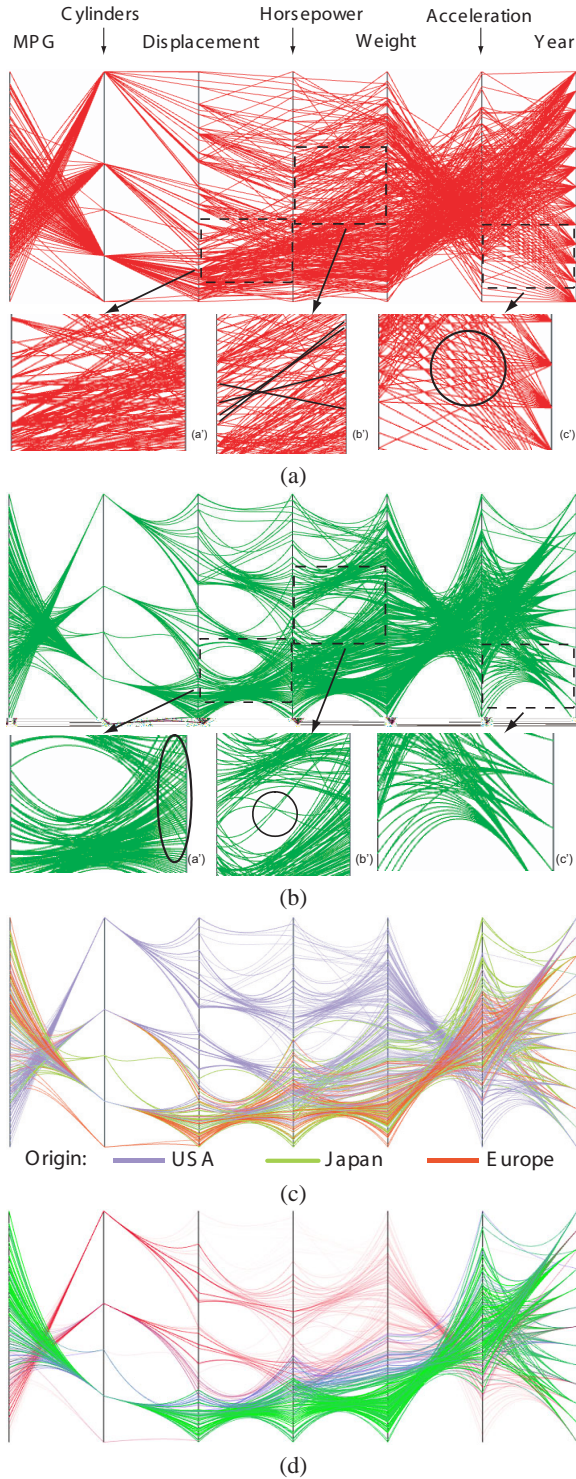
First we examined the effectiveness of our proposed



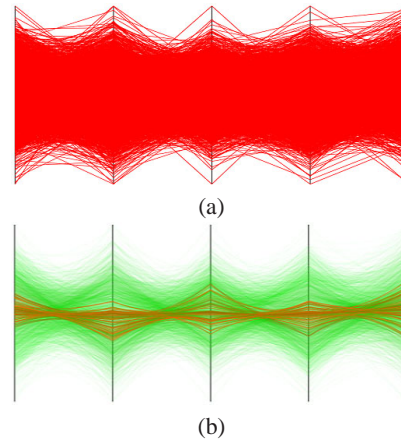
**Figure 4:** Visual clustering on synthesized datasets.

method on some synthesized datasets with simple patterns. In Fig. 4, seven examples are shown. The images on the top and bottom rows are plots of before and after applying visual clustering respectively. Our method successfully forms visual bundles for clusters in Fig. 4(a) to (d). Note there is one outlier edge in Fig. 4(e). Our visual clustering layout successfully distinguishes this outlier from the rest of the edges. Two clusters with fuzzy boundaries in Fig. 4(f) become more discernible after applying our algorithm.

Next we tested our algorithm on more sophisticated multivariate datasets. In Fig. 5, a car information dataset with seven variables, widely used in many previous publications [PWR04], is visualized. Fig. 5(b) is the visual clustering result over the original plot shown in Fig. 5(a). Our result successfully shows the major clusters and outliers even without the assistance of the color and opacity channels. Some interesting details are further illustrated in the sub-figures of Fig. 5(a) and (b). The interesting areas are highlighted with the black circles. In the region labeled with  $d'$ , the data values of the lines in the circled area become dis-



**Figure 5:** Experiments on a dataset with 7 variables and 392 data items: (a) Original plot; (b) After visual clustering optimization; (c) With color encoding the 8th variable, the origin of cars; (d) With one transfer function. Data from <http://lib.stat.cmu.edu/datasets/cars.data>.

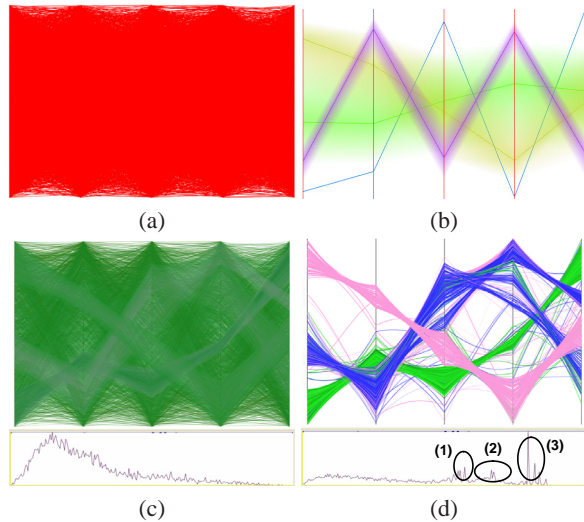


**Figure 6:** Experiments on a dataset with 5 variables and 3848 data items: (a) Original plot; (b) After visual clustering optimization and color and opacity enhancement. Data is from <http://rosuda.org/Mondrian/>.

cernable, because the lines are well organized and the intersections are grouped to leave clear spaces between lines. The four edges highlighted in the circled area in region  $b'$  are the fuzzy boundaries/outliers between the two neighboring clusters (See the black lines in Fig. 5(a)). Our algorithm isolates them from the major clusters. The lines in region  $c'$  likewise become discernable and beautifully grouped by our visual clustering method (See the corresponding circled area in Fig. 5(a)). As illustrated, the curved bundles can enhance the patterns within the data, and thus the color and opacity channels can be spared for other information. In Fig. 5(c), the origin of cars, the eighth variable in the dataset, is encoded by the color so that any possible correlation between this variable and all other variables can be conveniently examined. A transfer function can also be applied to reduce line density and separate different clusters (see Fig. 5(d)). In this example, we can see clearly that our geometry-based visual clustering scheme can effectively reduce visual clutter and enhance patterns.

For highly dense datasets, only using curved lines may not clearly reveal patterns, and it is in this manner that the color and opacity enhancement tool becomes indispensable. Fig. 6 illustrates our results on a five-dimensional Pollen dataset with 3848 entries. Fig. 6(a) is the parallel coordinates drawing without optimization, while Fig. 6(b) with optimization and enhancement. In Fig. 6(b), the only major cluster, the dark orange one in the center, is successfully detected by our scheme. This experiment demonstrates that our color and opacity enhancement method can be used together with visual clustering to reveal patterns in large datasets.

To further illustrate the effectiveness of our method, we worked with a synthesized dataset of 7736 five-dimensional items, with 2936 of its items distributed into four clusters



**Figure 7:** Experiments on a dataset with 5 variables and 7736 data items: (a) Original plot; (b) Hierarchical parallel coordinates result [FWR99]; (c) Applying a transfer function before visual clustering; (d) Applying a transfer function after visual clustering.

with 876, 752, 608 and 700 items respectively. The other 4800 items are noise generated randomly. Fig. 7(a) shows the original parallel coordinates without optimization. In Fig. 7(c) and (d), we apply the color and opacity enhancement scheme to the straight line drawing and the curved optimization result respectively. It is very hard to specify transfer functions in traditional parallel coordinates because the clusters are totally blended into the noise (see Fig. 7(c)). However, after applying visual clustering, we can immediately observe three major peaks in the histogram curve of Fig. 7(d): peak 1 corresponds to the pink cluster; peak 3 is for the green cluster; and peak 2 is for the two blue clusters. After checking with the data, we found that these four detected clusters are indeed the real patterns we planted in the data. Based on the information from the histogram curves shown in Fig. 7(d), appropriate transfer functions can be set to not only reveal a certain focusing cluster but also keep the context. Similar curves have also been obtained for the dataset used in Fig. 6. We further tested datasets consisting of purely random numbers and found that our histogram curves did not show any obvious major peaks and thus no artificial clusters were generated. Fig. 7(b) shows the result generated by the hierarchical parallel coordinates [FWR99]. Due to the dense noise, their algorithm cannot detect and show the four known clusters very well. This experiment demonstrates that the color and opacity enhancement becomes more effective after performing visual clustering.

We implemented our algorithm on a Dell Optiplex 745 desktop with Intel Core 2 Duo E6400 CPUs and 3GB Mem-

ory. The computation times of our visual clustering for datasets used in Fig. 5, Fig. 6, and Fig. 7 are 27s, 1714s, and 7797s respectively.

## 6. Discussions

From the experiments we can see that our visual clustering scheme has some clear advantages. Our scheme is visually pleasing and can greatly reduce visual clutter and enhance patterns. Compared with data-centric methods, our method does not filter out data when performing visual clustering so that the context can be preserved. Compared with other visual-centric methods, our technique exploits the geometry channel to expose patterns. Color and opacity enhancement will become more effective after visual clustering. In addition, our visual clustering scheme can be configured using different energy coefficients (see Fig. 2). However, we found that the default setting usually works well for most datasets. For example, all the experiments in this paper have been generated using the setting  $\alpha_c = 0.15$  and  $q_\alpha = q_d = 10$ .

Our method can enhance some patterns but it may also alter some otherwise obvious patterns and make them less discernible (see Fig. 4(b)). Thus users are suggested to use our system together with other parallel coordinates systems. In rare situations, our method may generate some artificial clusters so users should double check the results using the original parallel coordinates if in doubt.

One major disadvantage of our method is that solving the linear system may be time-consuming. The linear system can be constructed in  $O(n \lg(n))$  time complexity and the specification of color and opacity can be done in linear time while the used linear programming solver *lp\_solve* [Sof] has polynomial-time complexity. Therefore, it may take minutes or hours to generate the results for very large datasets. However, it is a one-time process that can be performed as a pre-processing. Once the layouts are generated, interactive visualization with different transfer functions can be achieved. As one of the most active projects to develop open source software for simplex-based linear programming, *lp\_solve* still may fail to solve some large feasible models. Therefore, for very large datasets, some data-centric clustering and filtering techniques can be used to reduce the data size to a manageable level before applying our method. Our current implementation is not highly optimized for speed, but if GPUs are exploited to solve the linear system, we believe that the computation time can be dramatically reduced.

## 7. Conclusions and Future work

In this paper, we have presented a novel framework to reduce edge clutter, consequently improving visual clustering for parallel coordinates. We exploit curved edges and adjust their shapes according to the curvature of each edge and its interaction with neighboring edges. We resort to an optimization method to optimize the configuration of the curved

edges by minimizing the energy of the system through linear programming. The overall visual clustering is achieved by geometrically bundling lines and forming patterns. We further enhance the visualization by varying color and opacity according to the local line density of the curves.

Our proposed work can be further improved in several ways. In our current implementation, the coherence along two sides of each coordinate axis has not yet been fully investigated. We plan to consider this coherence issue by enforcing curvature continuity across the axes for certain data entries. In our system, the interaction is mainly through the adjustment of transfer functions to selectively reveal features in large datasets. In the future, we plan to further augment our system with more sophisticated interaction tools such as brushing.

### Acknowledgments

We thank Amit Shesh, Nathan Gossett, and Jie Chen for their valuable suggestions. This work was supported by Hong Kong RGC grant CERG 618705 and 618706.

### References

- [AA04] ANDRIENKO G., ANDRIENKO N.: Parallel coordinates for exploring properties of subsets. In *Proc. of International Conf. on Coordinated & Multiple Views in Exploratory Visualization* (2004), pp. 93–104.
- [ABK98] ANKERST M., BERCHTOLD S., KEIM D. A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proc. of IEEE Symp. on Information Visualization* (1998), pp. 52–60.
- [AdL04] ARTERO A. O., DE OLIVEIRA M. C. F., LEVKOWITZ H.: Uncovering clusters in crowded parallel coordinates visualizations. In *Proc. of IEEE Symp. on Information Visualization* (2004), pp. 81–88.
- [BKH05] BENDIX F., KOSARA R., HAUSER H.: Parallel sets: visual analysis of categorical data. In *Proc. of IEEE Symp. on Information Visualization* (2005), pp. 133–140.
- [ED06] ELLIS G., DIX A.: Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Trans. on Vis. and Comp. Graph.* 12, 5 (2006), 717–724.
- [ED07] ELLIS G., DIS A.: A taxonomy of clutter reduction for information visualization. *IEEE Trans. on Vis. and Comp. Graph.* 13, 6 (2007), 1216–1223.
- [FWR99] FUA Y.-H., WARD M. O., RUNDENSTEINER E. A.: Hierarchical parallel coordinates for exploration of large datasets. In *Proc. of IEEE Visualization* (1999), pp. 43–50.
- [GK03] GRAHAM M., KENNEDY J.: Using curves to enhance parallel coordinate visualisations. In *Proc. of Intl. Conf. on Information Visualization* (2003), pp. 10–16.
- [HLD02] HAUSER H., LEDERMANN F., DOLEISCH H.: Angular brushing of extended parallel coordinates. In *Proc. of IEEE Symp. on Information Visualization* (2002), pp. 127–130.
- [Hol06] HOLTEN D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. on Vis. and Comp. Graph.* 12, 5 (2006), 741–748.
- [HR07] HEER J., ROBERTSON G.: Animated transitions in statistical data graphics. *IEEE Trans. on Vis. and Comp. Graph.* 13, 5 (2007), 1240–1247.
- [ID90] INSELBERG A., DIMSDALE B.: Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proc. of IEEE Visualization* (1990), pp. 361–378.
- [JLJC05] JOHANSSON J., LJUNG P., JERN M., COOPER M.: Revealing structure within clustered parallel coordinates displays. In *Proc. of IEEE Symp. on Information Visualization* (2005), pp. 125–132.
- [KBH06] KOSARA R., BENDIX F., HAUSER H.: Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Trans. on Vis. and Comp. Graph.* 12, 4 (2006), 558–568.
- [NH06] NOVOTNY M., HAUSER H.: Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Trans. on Vis. and Comp. Graph.* 12, 5 (2006), 893–900.
- [Nov04] NOVOTNY M.: Visually effective information visualization of large data. In *In Proc. of the 8th Central European Seminar on Computer Graphics* (2004).
- [PWR04] PENG W., WARD M. O., RUNDENSTEINER E. A.: Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proc. of IEEE Symp. on Information Visualization* (2004), pp. 89–96.
- [Sii00] SIIRTOLA H.: Direct manipulation of parallel coordinates. In *CHI'00 Ext. Abstr.* (2000), pp. 119–120.
- [Sof] SOFTWARE: <http://lpsolve.sourceforge.net>.
- [The00] THEISEL H.: Higher order parallel coordinates. In *Proc. of the Conference on Vision Modeling and Visualization* (2000), pp. 415–420.
- [WB96] WONG P. C., BERGERON R. D.: Multiresolution multidimensional wavelet brushing. In *Proc. of IEEE Visualization* (1996), pp. 141–148.
- [WCG03] WONG N., CARPENDALE S., GREENBERG S.: Edgelens: An interactive method for managing edge congestion in graphs. In *Proc. of IEEE Symp. on Information Visualization* (2003), pp. 51–58.
- [YPWR03] YANG J., PENG W., WARD M. O., RUNDENSTEINER E. A.: Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Proc. of IEEE Symp. on Information Visualization* (2003), pp. 105–112.