Parallel Coordinates with Data Labels

Hong Zhou
Shenzhen University
hzhou@szu.edu.cn

Panpan Xu
The Hong Kong University of Science and Technology
pxu@cse.ust.hk

Zhong Ming
Shenzhen University
mingz@szu.edu.cn

Huamin Qu
The Hong Kong University of Science and Technology
huamin@cse.ust.hk

ABSTRACT
Parallel coordinates have been widely used to analyze high-dimensional data. Numerous methods have been designed to provide overview patterns in parallel coordinate plots. However, detailed information is also important in data analysis. When several lines overlap or are close to one another, distinguishing detailed information of polyline crossings is difficult. In this paper, we present a novel approach to address the problem of polyline crossing ambiguity by using data labels. We place different labels along various polylines to give cues for differentiation of lines. We bend the lines and optimize the arrangement of curved lines to provide space for clear visible labels. An energy system that models attractive and repulsive forces of lines is used to guide the search for optimized line arrangement. The experiments on several real datasets demonstrate the effectiveness of our approach.

ACM Classification Keywords
H.4.0 Information Systems Applications: General; I.3.6 Computer Graphics: Methodology and Techniques

General Terms
Theory

Author Keywords
Parallel coordinates, multivariate visualization, labeling, label placement, information visualization

1. INTRODUCTION

Parallel coordinates [14] have been widely used to visualize multidimensional information or datasets. In parallel coordinates, multidimensional datasets can be easily displayed via a 2D mapping, wherein each dimension is drawn as a vertical axis and each data item is drawn as a polyline that connects its data values on parallel vertical axes. The design of 2D parallel axes allows the simultaneous display of multiple dimensions, and thus, high-dimensional datasets are visualized in a single image.

Many methods have been proposed to provide good overviews of parallel coordinates [11, 27, 19]. Visualizing a parallel coordinate plot (PCP) in overview involves displaying discernable high-level polyline patterns in the PCP. Under this criterion, precisely perceiving which segments are connected to form certain polylines is minimally important, and thus, showing detailed connectivity information in local areas of the PCP is frequently compromised. However, the objectives for visualizing a PCP in overview and in detailed view are different. In the latter, it requires to display clearly visible connection information between polyline segments. Identifying precisely which two segments are connected in one polyline is critical to allow users to trace a polyline and answer certain specific questions. Therefore, visualizing parallel coordinates in detailed view is essential.

To provide detailed views of PCPs, many approaches adopt zooming, colors [10, 33], or the curvature continuity of curves [9, 13] to show detailed information interactively. However, if two data items have the same value in a certain data dimension, then their corresponding polylines will cross at the same point on the dimension axis. Consequently, the zooming method still cannot help users obtain clear connection information between polylines segments by tracing polylines. We call this problem "polyline crossing ambiguity". An example of such case is shown in Fig. 1(a). Although using various colors to encode different polylines can help address this problem, colors are typically used to encode cluster information and reliably distinguishing more than 12 colors is difficult [30]. In addition, distorting lines to make them curve and using the curvature continuity of curves can help differentiate polylines. However, distinguishing different curves remains difficult in some cases, as shown in Fig. 1(b).

In this paper, we propose a novel technique to visualize

VINCI'14, August 05-08 2014, Sydney, NSW, Australia.
 ACM 978-1-4503-2765-7/14/08...$15.00
http://dx.doi.org/10.1145/2636240.2636854

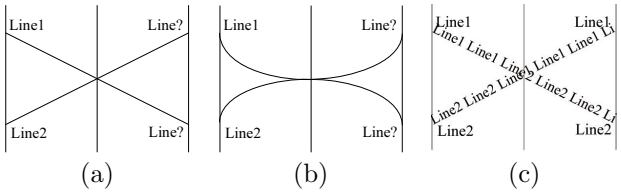

Figure 1. Polyline crossing ambiguity (a) is not solved by curves (b), but is solved by our methods (c).

parallel coordinates with data labels, wherein different labels can provide cues for differentiating polylines. We draw data labels as segments that connect data values on axes to form polylines. The label text starts from its data value on the first axis and is repeated until it reaches its data value on the last axis. Given that some lines may overlap or are near on another, we allow lines to curve and disperse to supply space for visible labels. We model parallel coordinates as a system with attractive and repulsive forces. Attractive forces prevent curves from bending too much, whereas repulsive forces disperse overlapping lines or those that are too close to one another. The optimized curve layout for the labels can be obtained by minimizing system energy via a linear programming solver. Compared with the NP-hard problem of drawing labels alongside lines [4], our approach is more efficient and effective because it replaces lines by labels. The contents of data labels range from names of data items (e.g., animal names) to the attribute values of data items or other information. The lengths of data labels range from short phrases to long sentences. Based on the contents of labels, users can easily trace polylines and obtain important labeling information for each polyline. The experiments on several real datasets demonstrate the effectiveness of our approach.

2. RELATED WORK

We aim to design a technique that will address the polyline crossing ambiguity problem. We first provide a summary of various parallel coordinates methods, and then discuss data labeling approaches for graphs and maps.

2.1 Parallel coordinates

Parallel coordinates have been widely used in multivariate visualization. Heinrich and Weiskopf [11] summarized the developments of parallel coordinates by focusing on visualization techniques. Meanwhile, we resummarize their work from the perspective of providing high-level or low-level patterns in PCPs in this section.

2.1.1 Overview methods

One of the major tasks in exploring parallel coordinates is to reveal important overall patterns in data. Some approaches [16, 21] first use the k-means algorithm [5] to extract data cluster information, and then apply various colors and opacities to visualize different data clusters. However, the appropriate value of k maybe difficult to set. Therefore, numerous clustering methods based on statistical information in PCP images have been proposed. For example, Novotny and Hauser [22] clustered data based on 2D-binning images for each pair of adjacent axes. Zhou et al. [33] visually clustered data based on the line-interaction energy computed from the PCP image. Hierarchical clustering methods have also been adopted to help users explore the multilevel clustering information [7, 24]. In addition, many filtering-based methods [3, 15] can reduce visual clutter caused by too many crossing or overlapping lines in PCPs, while preserving the significant features in the original data.

2.1.2 Detailed view methods

Revealing the overall structures of PCPs is essential in data analysis, whereas visualizing detailed views of PCPs is important to correctly perceive individual relations between data items. Therefore, many methods have been proposed to support interactive detailed-view explorations of PCPs. For example, brushing [10, 28] can be used to select certain groups of polylines. 2D-binning images [22], histograms [33, 8], scatterplots [32], and graphs [25] can also be used to find interesting patterns. These interaction methods typically highlight selected groups of polylines by using different colors. However, if the lines in the same group have the same color, the polyline crossing ambiguity problem within the group remains unsolved. If the lines in the same group have different colors, then users cannot distinguish similar colors when too many colors are used in the group. Our approach does not use colors to highlight polylines; therefore, it can be used with other interactive techniques.

Straight polylines can be curved to form edge bundles, and the curvature continuity can provide cues for the differentiation of polylines [9, 33]. In some cases, however, using curves still cannot help the differentiation of polylines, whereas our method is able to, as shown in Fig. 1(c).

2.2 Data labeling in graphs and maps

Placing text or symbol labels in graphs and maps is an important research area in information visualization. Labels are textual descriptions that convey information in graphical drawings; a number of tasks can be accomplished effortlessly by using labels [23]. Many automatic-label-placement techniques have been proposed and discussed for node labels [26, 20], edge labels [29], and both node and edge labels [18]. Label placement is typically an NP-hard problem [17]. A recent survey [6] classified labeling problems and summarized solutions. Unlike other label placement solutions that require additional display space for labels, Wong et al. [31] proposed a GreenArrow method that draws text labels to form edges between nodes in graphs. Our approach follows the concept of drawing text labels to form lines, but is specifically designed to address the polyline-crossing-ambiguity problem in parallel coordinates.

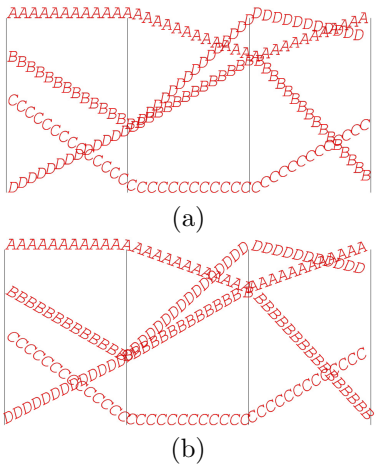

Figure 2. Parallel coordinate labels shown in placement schemes of (a) vertical labels and (b) labels perpendicular to polylines.

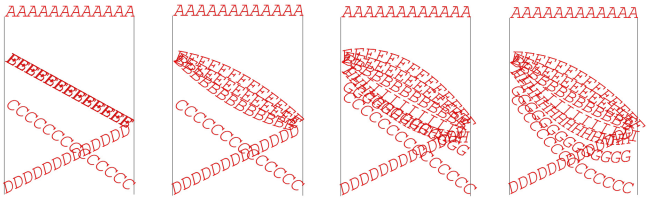

Figure 3. Placement examples: (a) overlapping labels, (b) curved labels, (c) curved labels that result in additional overlapping, and (d) dispersed labels with global optimization.

3. LABEL DESIGN

Many techniques have been proposed to reveal overall patterns and detailed information in parallel coordinates. In this paper, we introduce a novel method to address the polyline-crossing-ambiguity problem (see Fig. 1(a)) in parallel coordinates. We use different labels to provide cues for differentiation of polylines in parallel coordinates, and thus, assist users in discerning detailed information of data. We draw the label text of each polyline from its data value on the first axis and repeat the text until it reaches its data value on the last axis. The labels can be placed with different schemes. For example, a label can be vertical, perpendicular to its original polyline, or perpendicular to the dispersed curve. The contents of data labels can be the name of the data item or any attribute value of the data item. Users can easily trace polylines based on the contents of data labels, and thus, obtain detailed information on each polyline. The color of data labels can be used to encode other interesting information such as cluster information.

3.1 Placement schemes

Labels are textual descriptions that reveal information or clarify structures in graphical drawings. Therefore, labels are important in information visualization. However, given limited screen space, determining how to place labels automatically without overlapping is a difficult problem. Label placement is an NP-hard problem in graphs and maps [17]. Our method places labels along polylines in parallel coordinates; therefore, we do not have to consider the NP-hard problem of seeking space for the label layout. Our placement schemes in this section discuss methods for placing labels along polylines and arranging underlying polylines for better label placement.

Typically, solving a labeling problem involves searching and arranging space for good label assignment. Our placement problem is simple, because polylines are located in parallel coordinates, and our design involves placing labels along polylines without requring additional drawing space. Different datasets may have various features and different users may have various requirements; therefore, our labeling technique provides three placement schemes that users can select based on their scenarios. The first and second schemes, namely, vertical labels and labels perpendicular to polylines, rotate labels but do not move underlying polylines. The third scheme, namely, labels perpendicular to the dispersed curves, bend polylines to curve them and disperse curves for improved label appearance.

3.1.1 Vertical labels

In vertical label mode, all characters are drawn along the underlying polylines regardless of how the polylines are rotated. Fig. 2(a) shows an example of such drawing.

3.1.2 Labels perpendicular to polylines

In this mode, the characters in a label are rotated to make them perpendicular to their corresponding polyline as shown in Fig. 2(b). The moving directions of polylines are more discernable in this mode; thus, we use this mode as the default value for the rest examples of this paper.

3.1.3 Labels perpendicular to the dispersed curves

Polylines can be easily differentiated one by one based on our drawn data labels (Fig. 2). However, when a PCP has several overlapping polylines, the corresponding labels also overlap, thus preventing the distinguishing of polylines, as shown in Fig. 3(a). If only a small number of polylines overlap or are near one another, a straightforward method is to bend these polylines and arrange them in order with intervals for visible labels (Fig. 3(b)). However, this straightforward method is unsuitable for some general cases. For example, in Fig. 3(c), a bunch of nearby polylines (i.e., polylines *"B"*, *"E"*, *"F"*, *"I"*, and *"H"*) are detected with a threshold value. Because the threshold value is not large enough, another nearby polyline *"G"* is not considered as a part of this group of nearby polylines. The polyline *"G"* is finally overwhelmed by other labels. Regardless of the value of the threshold, this type of polyline *"G"* may still exist. To address this problem and improve the

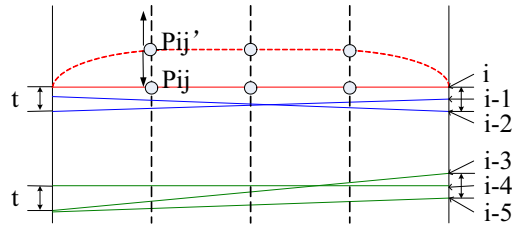

Figure 4. The attractive and repulsive forces.

label layout, we propose an effective placement scheme with labels perpendicular to the dispersed curves. The positions of the dispersed curves are computed under an optimization process that can globally optimize the layout of all the labels. After applying our new scheme, the previously hidden polyline *"G"* (see Fig. 3(c)) becomes visible in Fig. 3(d). In addition, our approach bends polyline *"C"* which is below polyline *"G"* to make the label of polyline *"C"* clear and not overlapped by upper polyline *"G"*.

To display the labels perpendicular to the dispersed curves, we need to first search for the layout of the dispersed curves, and then draw labels along the curves. We model the parallel coordinate plot as a system with attractive and repulsive forces. Attractive forces prevent the curves from bending too much, whereas repulsive forces disperse overlapping curves or those that are too close one another. Minimizing the system energy can help obtain the optimized curve layout for the labels of PCPs.

In our proposed PCP system, the total energy of the polylines consists of three energy terms as follows:

$$\begin{aligned}\mathbf{E}_{total} =& c_{attraction}\mathbf{E}_{attraction} + \\ & c_{repulsion}\mathbf{E}_{repulsion} + \\ & c_{others}\mathbf{E}_{others}\end{aligned} \tag{1}$$

where $\mathbf{E}_{total}$ is the total energy of the PCP, $\mathbf{E}_{attraction}$ is the energy term that describes the attractive forces from the original positions of the lines before distortion, $\mathbf{E}_{repulsion}$ is the energy term that describes the repulsive forces between neighboring lines, and $\mathbf{E}_{others}$ is the energy term for other purposes. $c_{attraction}$, $c_{repulsion}$, and c_{others} are the weighting coefficients of the corresponding energy terms. The forces are considered in each column (i.e., the adjacent two axes in a PCP), so we discuss our energy model only for one column in the rest of this section.

We intend to use efficient linear programming as a design guideline to minimize the energy $\mathbf{E}_{total}$. Therefore, all the following energy terms are designed to be linear functions.

Attractive force term

In the energy system, the overlapping and nearby lines are detected and bent to leave space for visible labels. We set the attractive force term that prevents each line from being bent too much. For one column in a PCP, we assume that n lines (i.e., data items in the dataset) exist. The $\mathbf{E}_{attraction}$ term is modeled as follows:

$$\mathbf{E}_{attraction} = \sum_{i=1}^{n}\sum_{j=1}^{s} ||P'_{ij} - P_{ij}|| \tag{2}$$

where P_{ij} is a point on the straight line i (the red line in Fig. 4), and P'_{ij} is the corresponding control point on the curved line i (the dotted red curve in Fig. 4). We set the constraint that P_{ij} and P'_{ij} have the same horizontal coordinates. s control points are sampled for each line, and the curve shapes are specified by the positions of the control points. In this paper, s is set to be 3, and third-degree Bezier curve is used to draw the curves. The value of s can be changed, and other drawing algorithms, such as Hermite and Catmull Rom curves, can be also used in our system.

Repulsive force term

We design the repulsive force term to disperse neighboring lines to leave the space for labeling. For n lines (i.e., data items in the dataset), the sets of neighboring lines are detected with a threshold distance value t. We set t to be the height of the used font size. In a set of neighboring lines, the distance of any pair of endpoints or control points with the same horizontal coordinates is no more than the value of t. For example, in Fig. 4, line i, $i+1$, and $i+2$ are in such a set, and line $i+3$, $i+4$, and $i+5$ are in another set. We assume that A sets exist, and each set has a_k lines, where $1 \le k \le A$, and the kth set is indicated as S_k. Note that A may be 0, and a line may not be in any set or be in two sets (i.e., a set of lines above it and a set of lines below it). The $\mathbf{E}_{repulsion}$ term is modeled as follows:

$$\begin{aligned}&\mathbf{E}_{repulsion} = -\sum_{k=1}^{A}\sum_{j=1}^{s}\sum_{i<m, P'_{ij}<P'_{mj}}^{a_k} ||P'_{ij} - P'_{mj}|| \\ &where\ ||P'_{1j} - P'_{2j}|| = ||P'_{2j} - P'_{3j}|| = ... \\ &\quad = ||P'_{a_k-2j} - P'_{a_k-1j}|| = ||P'_{a_k-1j} - P'_{a_kj}|| \le t, \\ &\forall (r,u) \in R, if\ \exists k, r, u \in S_k, then\ \forall j, P'_{rj} < P'_{uj}, \\ &\qquad else\ no\ such\ k, then\ \forall j, P'_{uj} - P'_{rj} \ge t\end{aligned} \tag{3}$$

Note that the negative sign indicate that the minimization of $\mathbf{E}_{repulsion}$ becomes the maximization of the distance between P'_{ij} and P'_{mj}. However, the repulsive force is designed to disperse nearby lines to obtain enough space for visible labels. Thus, our dispersing will stop if the space is enough. To achieve this effect, we set the

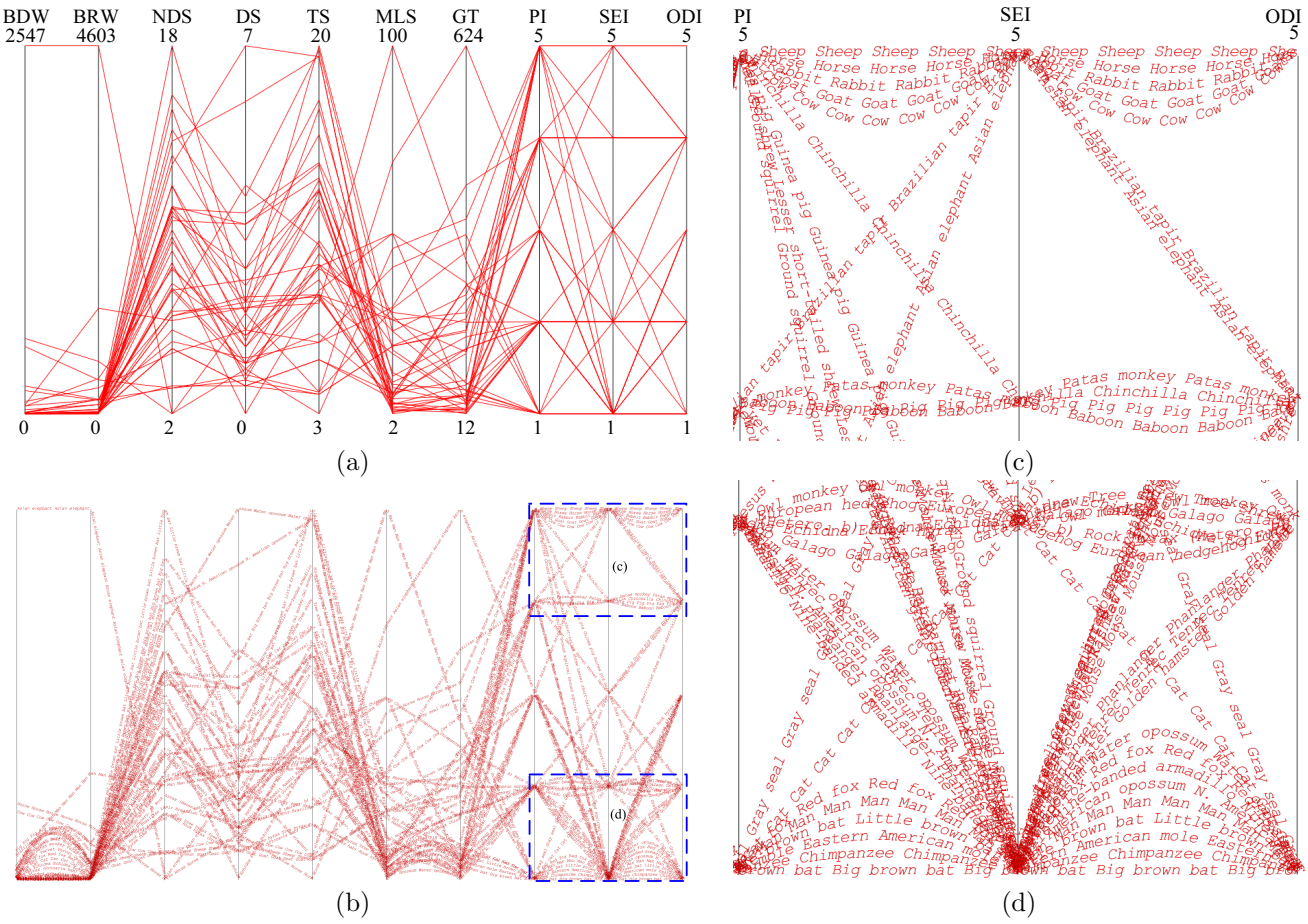

Figure 5. Experiments on the "sleep in mammals" dataset: (a) original plot; (b) with labeling; (c) with the names of mammals that have the most danger from other animals shown; (d) with the names of mammals that have the least danger from other animals shown.

constraint of $||P'_{ij} - P'_{mj}|| \leq t$ and the distances of each pair of adjacent points are equal. In addition, when a set of neighboring lines is bent, non-intersecting lines within this set cannot become intersected, and the other lines that are not in this set but are above or below this set should not be freshly overlapped by the new curves in this set(see Fig. 3(c) and (d)). Therefore, we detect all the pairs of adjacent lines (r, u), where lines r and u are not intersected, and $\forall j$, $1 \leq j \leq s$, then $P'_{rj} < P'_{uj}$. We use R to indicate this set of pairs. In Fig. 4, all the pairs of adjacent lines (r, u) are (i,i-1), (i,i-2), (i-1,i-3), (i-1,i-4), (i-2,i-3), (i-2,i-4), (i-3,i-5), (i-4,i-5).

Other terms

The $\mathbf{E}_{others}$ energy term extends our system to be suitable for additional effects. For example, the gravitation energy term [33] and attracting electrostatic force [12] can be used as our $\mathbf{E}_{others}$ to achieve the clustering effect.

Finally, our energy model $\mathbf{E}_{total}$ is minimized with a linear programming solver *lp_solve* [1], which is based on the revised simplex method and the branch-and-bound method.

3.2 Content schemes

The contents of the data labels are flexible. We can set the labels to be the names of data items (e.g., animal names), the attribute values of data items, or other pieces of information. The lengths of the data labels range from short phrases to long sentences. Basing on the contents of labels, users can easily trace the polylines and determine the labeling information of each polyline as well.

3.3 Color schemes

We use the contents of data labels to give cues for the differentiation of lines. Therefore, the color channel can be used to further encode other interesting pieces of information, such as attribute values and cluster information.

4. EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of our labeling method through experiments on three real datasets.

4.1 Sleep in mammals

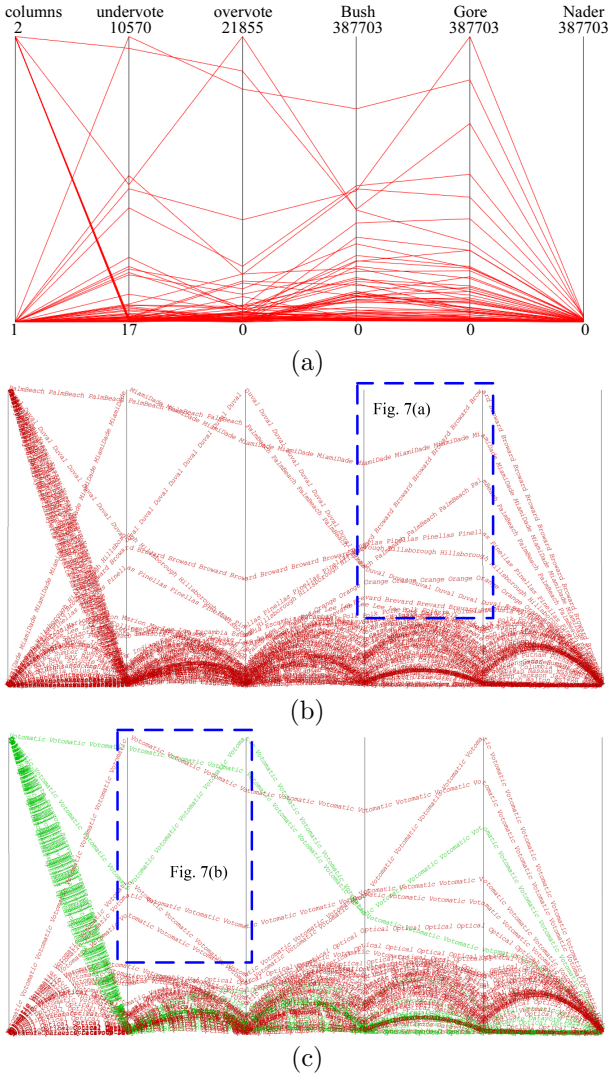

Figure 6. Experiments on the "fl2000" dataset: (a) original plot; (b) with county names as data labels; (c) with the "technology" as data labels and color encoding the "columns".

We tested our method on a "sleep in mammals" dataset with 42 mammals and 10 variables [2]. The 10 variables are body weight in kg (BDW), brain weight in g (BRW), nondreaming sleep in hrs/day (NDS), dreaming sleep in hrs/day (DS), total sleep in hrs/day (TS), maximum life span in years (MLS), gestation time in days (GT), predation index (PI), sleep exposure index (SEI), and overall danger index (ODI). For the variable PI, value 1 means the mammal is least likely to be preyed upon; for the variable SEI, value 1 means the mammal sleeps in a well-protected den; for the variable ODI, value 1 means the mammal has the least danger from other animals. The dataset has 62 mammals with some missing values at first. We removed those mammals with missing values and use the 42 mammals to test our method.

Fig. 5(b) shows our labeling result over the original plot in Fig. 5(a). Our result successfully disperses the over-

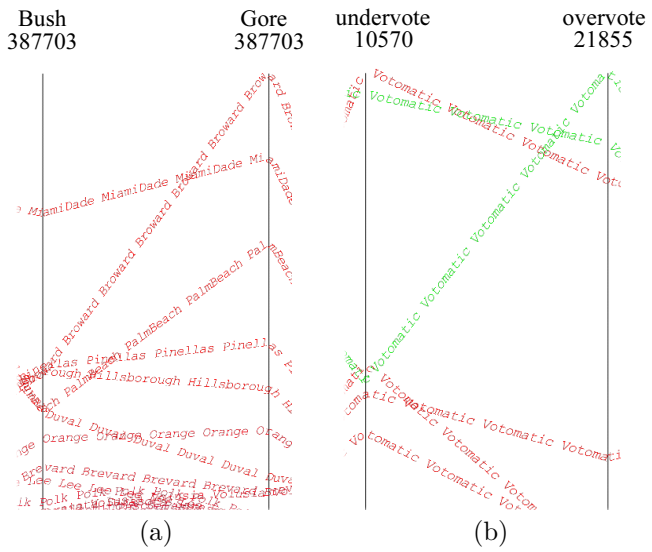

Figure 7. Enlarged blue rectangular areas for Fig. 6: (a) with visible county names of the blue rectangular region in Fig. 6(b); (b) with the visible relationship between "overvote", "technology", and "columns"(i.e., the enlarged blue rectangular region in Fig. 6(c)).

lapping and nearby polylines, clearly shows their labels (mammal names), and gives cues to clarify the crossing ambiguity cases when users are tracing the polylines. From Fig. 5(c), we can notice that the mammals with the most danger from other animals are "Sheep", "Horse", "Rabbit", "Goat", and "Cow", all of which also have the highest values of sleep exposure index and predation index. However, in Fig. 5(a), we cannot determine how many and which mammals have the highest ODI value, not to mention the identification of the lines that have the highest ODI value, the highest SEI value, and the highest PI value, because those lines representing "Brazilian tapir" and "Asian elephant" cause the crossing ambiguity problem. In Fig. 5(d), the names of 11 mammals with the lowest ODI value (i.e., the least danger from other animals) are also clearly shown, and 9 of them also have the lowest SEI value, 6 of them (i.e., "Red fox", "Man", "Little brown bat", "Eastern American mole", "Chimpanzee", and "Big brown bat") have both the lowest SEI value and the lowest PI value. An interesting finding is that the lines labeled "Gray seal" and "Cat" start from the smallest ODI value then go to some bigger SEI values, and finally come back to the smallest PI value. In addition, the labeled polylines for each mammal on the last three axes (i.e., PI, SEI, and ODI) are also traceable. In this example, we can see clearly that our approach can help users identify overlapping or nearby polylines, resolve polyline-crossing-ambiguity problems, and reveal detailed information in the data.

4.2 fl2000

The dataset named "fl2000" [2] is the county data from the 2000 presidential election in Florida, USA. For each of the 67 Florida counties, the data record the type

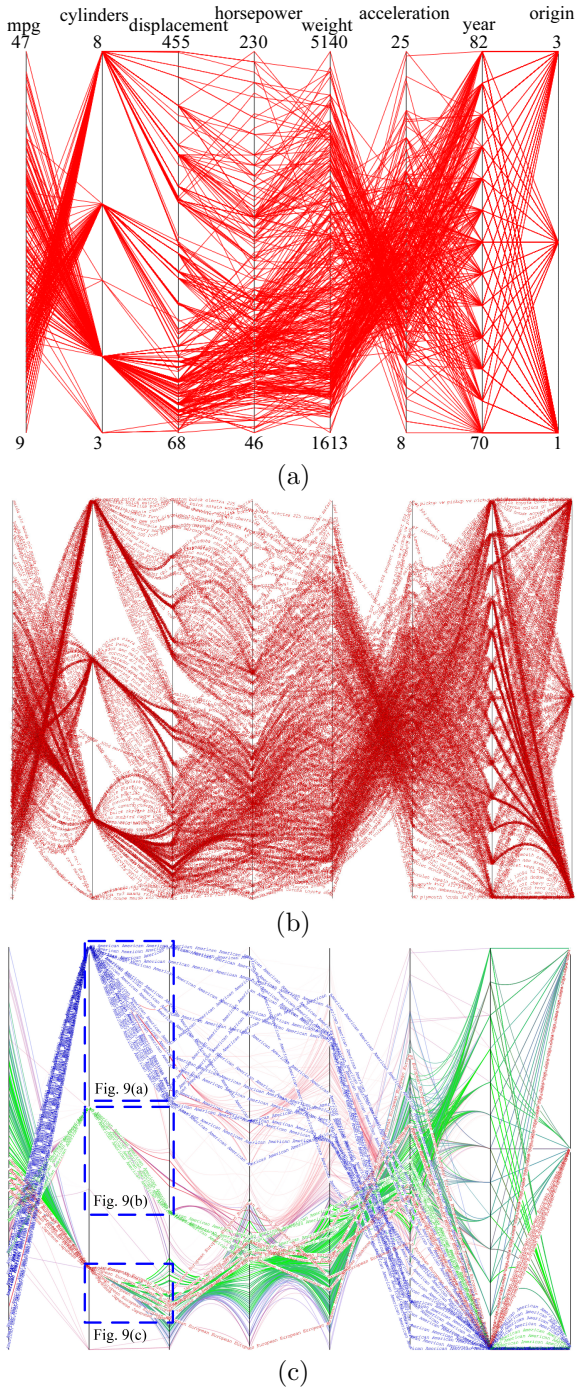

Figure 8. Experiments on the "cars" dataset: (a) original plot; (b) with car names as data labels; (c) with car origins as data labels.

of voting machine used (technology), the number of columns in the presidential ballot (columns), the undervote, the overvote, and the official certified votes for each of the 12 presidential candidates (we only chose the most three popular candidates "Bush", "Gore", and "Nader" for our experiment). Fig. 6(b) is our labeling result over the original plot shown in Fig. 6(a); county names are used as data labels. Our result reveals detailed information in the data. Fig. 7(a) shows an enlarged sub area in Fig. 6(b). In this area, we can notice that in the county of "Broward", "MiamiDade", "PalmBeach", "Pinellas", and "Orange", Gore's votes are higher than Bush's votes.

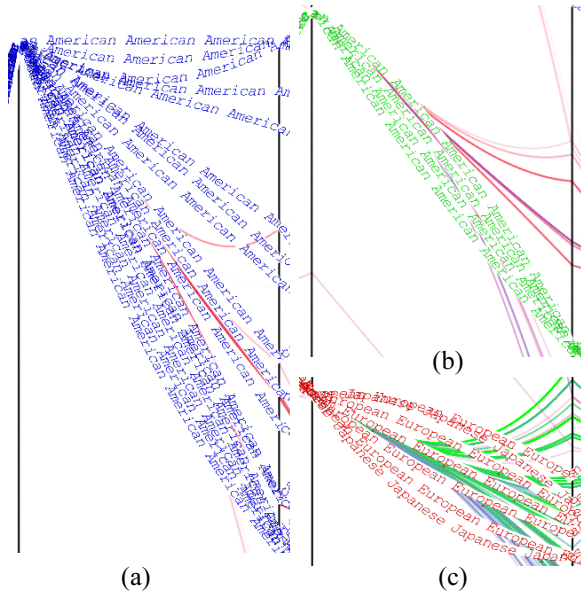

Figure 9. Enlarged blue rectangular areas for Fig. 8(c).

Furthermore, our data labels can aid in the identification of whether any correlation exists among the data attributes. In the "fl2000" dataset, we further verified the relationships among the attributes of technology, columns, undervote, and overvote. In the data, the values of technology are not recorded as numbers, but as words from the text set of "Optical", "Votomatic", "Datavote", "Lever", and "Hand". Therefore, this attribute was not visualized in Fig. 6(a) and 6(b), whereas in Fig. 6(c), we used the type of technology as data label and assigned different colors to the labels with different values of columns. A red label indicates that the ballot listed the presidential candidates in 1 column, whereas the green label indicates that the presidential candidates were spread over 2 columns. The blue rectangular area in Fig. 6(c) is enlarged in Fig. 7(b). We easily note that all the counties with a high number of undervote or overvote must use the "Votomatic" type of voting machine, and the counties with the highest and the second highest number of overvote both have the "2" columns in the ballot and the "Votomatic" type of voting machine. This experiment demonstrates that our labeling method can not only show detailed information in the data, but can also aid in the identification of correlations among data attributes.

4.3 Cars

We also tested our algorithm on a widely used "cars" dataset [2] (See Fig. 8(a)). In Fig. 8(b), the car information dataset with 8 variables and 392 items is visualized

with car names as data labels. Our result successfully shows the labels and line directions of several overlapping lines. However, numerous lines remain overlapping. Our method searches for a global optimized layout, but the high line denseness in local areas and the limited screen space causes our method fail to guarantee the local optimization of visible labels. Therefore, for large datasets, users can apply the proposed approach to a subset of data while using other overview methods simultaneously. We further tested the effectiveness of our method when it is used together with other techniques. In Fig. 8(c), the upper layer is the labeling result highlighting a subset of data items whose value of year attribute is 70, and the background layer is the coloring curves generated by the visual clustering technique [33]. The origins of cars (i.e., American, European, Japanese) are used as data labels, and the cylinder number of cars are represented by the color of labels. In the upper layer, the moving directions of labels and the colors show that the attributes mpg and cylinders have an inverse correlation in this subset. However, the inverse correlation does not exist in the whole dataset, because several red lines do not maintain a consistent slope angle in the background layer. Three interesting areas in Fig. 8(c) are enlarged in Fig. 9. We can notice that in 1970 all the types of cars with 8 or 6 cylinders were made in American (See Fig. 9(a) and (b)). Meanwhile, five types of European cars and two types of Japanese cars have 4 cylinders (See Fig. 9(c)). This experiment demonstrates that our method can be used with other methods, and can successfully reveal detailed information and correlations among attributes for large datasets.

All our results were generated on a T410 Lenovo notebook computer with Intel Core i5 Duo 2.53 GHz CPUs and 4GB memory. The computation times of our labeling results for the datasets used in Fig. 5, Fig. 6, Fig. 8(b), and Fig. 8(c) are 4.7s, 19.5s, 986s, and 3.2s, respectively.

5. DISCUSSIONS

From the experiments, we can see that our approach has some clear advantages. Our method can successfully disperse several overlapping or nearby lines, and it can draw their clear labels to help users trace the polylines. Compared with the NP-hard problem of drawing labels alongside the lines, our method is simple because it uses a linear system to find the optimized line arrangement. The level of line dispersing can be controlled by different energy coefficients. Our default setting of $c_{attaction} = 0.1$, $c_{repulsion} = 6$, and $c_{others} = 0$ works well for most datasets.

Our method is designed to show detailed information for PCPs, so, it works well for small datasets without too many overlapping lines. However, if an area with very dense lines in a PCP exists, because of the limited screen space, showing all the line labels is theoretically impossible. In Fig. 6(b) and Fig. 8(b), we can see that some lines are still overlapping. Therefore, users are suggested to use our technique, together with other overview methods, or just apply our technique to a subset of data.

One major disadvantage of our method is that solving the linear system may be time-consuming. *lp_solve* [1], which we used, has polynomial-time complexity. Therefore, computing the optimized line layouts for very large datasets may take minutes or hours. However, our method can be a one-time preprocess step, and applying to only a small subset from the large datasets is suggested.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a novel technique to visualize parallel coordinates with data labels and to use different labels to provide cues for the differentiation of polylines. We exploit curved lines and adjust their shapes according to an energy model to save space for visible labels. The energy model consists of attractive forces designed to prevent curves with high curvature and repulsive forces designed to disperse overlapping lines or lines that are too close. The energy system is minimized with a linear programming solver, and the data labels are placed along the optimized curves. Therefore, the polylines in parallel coordinates can be easily traced on the basis of labels, and important labeling information is also revealed. Our approach is suitable to be used together with other overview methods or to be applied to a subset of data.

In the future, we plan to investigate the effectiveness of our labeling method through formal user studies. We also plan to design some sophisticated interaction tools for our system, such as brushing.

7. ACKNOWLEDGMENTS

We thank anonymous reviewers for their valuable comments. This work is supported by National Natural Science Foundation of China 61103055, 61170077 and 61373033, NSF GD 10351806001000000, S&T project of GDA 2012B091100198, FDYT LYM11113, and S&T project of SZ JCYJ20120613102030248 and JCYJ20130326110956468.

REFERENCES

1. lp_solve. http://lpsolve.sourceforge.net.
2. Statlib. http://lib.stat.cmu.edu/datasets/.
3. A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *Proc. of IEEE Symp. on Information Visualization*, pages 81–88, 2004.
4. J. Christensen, J. Marks, and S. Shieber. An empirical study of algorithms for point-feature label placement. *ACM Trans. on Graph.*, 14(3):203–232, 1995.

5. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

6. R. T. Editor. *Handbook of Graph Drawing and Visualization*. CRC Press, 2013.

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

8. Z. Geng, Z. Peng, R. Laramee, J. Roberts, and R. Walker. Angular histograms: Frequency-based visualizations for large, high dimensional data. *IEEE Trans. on Vis. and Comp. Graph.*, 17(12):2572–2580, 2011.

9. M. Graham and J. Kennedy. Using curves to enhance parallel coordinate visualisations. In *Proc. of Intl. Conf. on Information Visualization*, pages 10–16, 2003.

10. H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Proc. of IEEE Symp. on Information Visualization*, pages 127–130, 2002.

11. J. Heinrich and D. Weiskopf. State of the art of parallel coordinates. In *Eurographics 2013 - State of the Art Reports*, pages 95–116, 2012.

12. D. Holten and J. J. van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009.

13. D. Holten and J. J. van Wijk. Evaluation of cluster identification performance for different pcp variants. *Computer Graphics Forum*, 29(3):793–802, 2010.

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

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

16. J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *Proc. of IEEE Symp. on Information Visualization*, pages 125–132, 2005.

17. K. G. Kakoulis and I. G. Tollis. On the edge label placement problem. In *Proc. of Graph Drawing*, pages 241–256, 1996.

18. K. G. Kakoulis and I. G. Tollis. A unified approach to automatic label placement. *Int. J. Comput. Geometry Appl*, 13(1):23–60, 2003.

19. S. Liu, W. Cui, Y. Wu, and M. Liu. A survey on information visualization: Recent advances and challenges. *The Visual Computer*, 2014.

20. M. Luboschik, H. Schumann, and H. Cords. Particle-based labeling: Fast point-feature labeling without obscuring other visual features. *IEEE Trans. on Vis. and Comp. Graph.*, 14(6):1237–1244, 2008.

21. K. T. McDonnell and K. Mueller. Illustrative parallel coordinates. *Computer Graphics Forum*, 27(3):10311038, 2008.

22. M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Trans. on Vis. and Comp. Graph.*, 12(5):893–900, 2006.

23. A. J. Pretorius and J. J. van Wijk. Visual inspection of multivariate graphs. *Computer Graphics Forum*, 27(3):967–974, 2008.

24. R. Rosenbaum, J. Zhi, and B. Hamann. Progressive parallel coordinates. In *Proc. of IEEE Pacific Visualization Symp.*, pages 25–32, 2012.

25. R. Santamaria, R. Theron, and L. Quintales. A visual analytics approach for understanding biclustering results from microarray data. *BMC Bioinformatics*, 9(247), 2008.

26. M. Schreyer and G. R. Raidl. Letting ants labeling point features. In *Proc. of IEEE Congress on Evolutionary Computation*, pages 1564–1569, 2002.

27. G. Sun, Y. Wu, R. Liang, and S. Liu. A survey of visual analytics techniques and applications: State-of-the-art research and future challenges. *Journal of Computer Science and Technology*, 28(5):852–867, 2013.

28. C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions - a dual visual analysis model for high-dimensional data. *IEEE Trans. on Vis. and Comp. Graph.*, 17(12):2591–2599, 2011.

29. F. Wagner and A. Wolff. A combinatorial framework for map labelling. In *Proc. of Graph Drawing*, pages 316–331, 1998.

30. C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2004.

31. P. C. Wong, P. Mackey, K. Perrine, J. Eagan, H. Foote, and J. Thomas. Dynamic visualization of graphs with extended labels. In *Proc. of IEEE Symp. on Information Visualization*, pages 73–80, 2005.

32. X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Trans. on Vis. and Comp. Graph.*, 15(6):1001–1008, 2009.

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