Web-based Scalable Visual Exploration of Large Multidimensional Data Using Human-in-the-Loop Edge Bundling in Parallel Coordinates

Wenqiang Cui
Department of ICT and Natural Sciences
Norwegian University of Science and Technology
Norway
wenqiang.cui@ntnu.no

Girts Strazdins
Department of ICT and Natural Sciences
Norwegian University of Science and Technology
Norway
gist@ntnu.no

Hao Wang
Department of Computer Science
Norwegian University of Science and Technology
Norway
hawa@ntnu.no

CEUR-WS.org/Vol-2578/BigVis8.pdf

ABSTRACT

Visual clutter and overplotting are the main challenges for visualizing large multidimensional data in parallel coordinates, which greatly hampers the recognition of patterns in the data. Although many automatic clustering and edge-bundling methods have been used in parallel coordinates to reduce visual clutter and overplotting, a scalable, transparent, and interactive approach that allows analysts to interact with large data and generate interpretable results of visualization in real time is lacking. To solve this problem, we propose an approach, human-in-the-loop edge bundling, to visually explore and interpret large multidimensional data in parallel coordinates. This approach combines data binning-based clustering and density-based confluent drawing, which reduces much data processing time and rendering time. It provides novel interactions, such as splitting, adjusting, and merging clusters, to integrate human judgment into the edge-bundling process. These interactions make the underlying clustering transparent to users, which allow users to generate interpretable visualization without complex data clustering. The scalability of our approach was evaluated through experiments on several large datasets. The results show that our approach is scalable for large multidimensional data, which supports real-time interactions on millions of data items in web browsers without hardware-accelerated rendering and big data infrastructure-based data processing. We used a case study to highlight the effectiveness of our approach. The results show that our approach provides an interpretable way of visually exploring large multidimensional data in parallel coordinates.

KEYWORDS

interactive visualization, human-in-the-loop, visual exploration, multidimensional data, big data, parallel coordinates

1 INTRODUCTION

A multidimensional dataset contains numerical or categorical dimensions (or features), with n ($n > 3$) dimensions and m data items. To avoid confusion, in this paper, a data item is an n-dimensional point, and a data point is the projection of a data item to a particular dimension. Parallel coordinate plots (PCPs) are widely used, and have become a standard tool for visualizing multidimensional data [6]. In PCPs, axes corresponding to the number of dimensions are aligned parallel to each other, and data items are mapped to lines (or edges) intersecting the axes at their respective values. The embedding of an arbitrary number of parallel axes into the plane allows for the simultaneous display of many dimensions to provide a good overview of the data, which reveals intrinsic patterns and trends. However, when datasets are large, PCPs create visual clutter and overplotting in which lines are crossed and plotted on top of one another, overwhelming the display, and obscuring the underlying patterns. This hides information and hampers the recognition of patterns in the data.

Edge bundling [7] and automatic data clustering [10] are two widely used approaches to reduce visual clutter and overplotting in PCPs. Edge bundling bends similar lines to the center of visual clutters in groups to create more informative visualizations. Automatic data clustering aggregates data points in groups that can be visualized in an illustrative fashion using different forms of edge bundling.

However, when datasets become large, these methods face challenges in supporting real-time interactions (limiting the visual response in a few milliseconds) along with mechanisms for information abstraction. Without interactions, these automatic methods provide only groups that may contain interesting combinations of dimensions and data points, but do not give analysts control over the data clustering and visualization processes, and do not offer opportunities for analysts to take advantage of their judgments and expertise.

In this study, we propose a web-based visual analytics system that uses data binning-based clustering and density-based confluent drawing to create a new edge-bundling paradigm in PCPs for large multidimensional data. To the best of our knowledge, this is the first web-based system that supports the HITL (human-in-the-loop) edge-bundling process in PCPs through specific interactions, such as splitting, adjusting, and merging clusters of each dimension, for large multidimensional data. The contribution of this study are as follows:

- **New paradigm for edge bundling in PCP.** Our approach provides a novel edge-bundling paradigm (HITL edge bundling) for the visual exploration of large multidimensional data in PCPs. With the real-time interactions, such as splitting, adjusting, and merging clusters, it enables analysts to integrate their judgments and expertise into the data clustering and edge-bundling processes of large multidimensional data.
- **Fast, scalable, and transparent edge-bundling algorithm.** To support the real-time interactions of large data in PCPs, we propose a fast, scalable, and transparent edge-bundling algorithm that consists of two parts: 1) a data

Copyright © 2020 for this paper by its author(s). Published in the Workshop Proceedings of the EDBT/ICDT 2020 Joint Conference (March 30-April 2, 2020, Copenhagen, Denmark) on CEUR-WS.org. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)

binning-based clustering method, and 2) density-based confluent drawing.
- **A web-based visual analytics system.** We build a web-based visual analytics system to support HITL edge bundling in PCPs for large multidimensional data.
- **Experiments, and a case study.** We conducted experiments and a case study on several datasets to highlight the benefits of HITL edge bundling in PCPs for large multidimensional data.

The remainder of this paper is organized as follows: Section 2 presents the proposed approach. Section 3 reports the experiments, a case study, and discusses the result. Section 4 draws the conclusions of this study and discusses directions for future work.

2 SYSTEM AND METHODS

In this section, we first describe the HITL edge-bundling process with our system. Then, we introduce the methods used in the system and the novel interactions provided by the system.

2.1 System Overview

Figure 1 shows the overview of our system. The system first visualizes multidimensional data in a classic PCP without edge bundling. For example, in Figure 1 (A), the Cars dataset [1] is visualized in a classic PCP without edge bundling. The system then bundles the edges according to the initial clusters for each dimension as shown in Figure 1 (B). The system supports HITL edge bundling by allowing analysts to split, adjust, and merge clusters for each dimension, which is shown in Figure 1 (C). During the HITL edge-bundling process, the system can update the visualization according to the corresponding interactions in real time for large multidimensional data. This makes the underlying clustering process transparent to analysts. With the interactions, analysts can integrate their judgments and expertise into the edge-bundling process to generate visualizations that can be better interpreted. For example, in Figure 1 (C), by creating an empty cluster that ranges from 6 to 8 and a cluster with 0 diameter (ranges from 8 to 8) at 8 on the axis *cylinders*, we found that all cars with eight cylinders in the dataset weighted between 3354 and 5140 kilograms. Moreover, by highlighting the subsets that contains cars with eight cylinders in red, the patterns of other features of these cars are clearly highlighted.

The rudiment of our system is the combination of data binning-based data clustering and density-based confluent drawing, which supports the real-time interactions for large multidimensional data without hardware-accelerated rendering and big data infrastructure-based data processing. Figure 2 shows the workflow of our system, where the HITL process is highlighted in the dashed line rectangle. The system first uses data binning to cluster data points for each dimension with the default settings. Then the density of each pair of clusters on two adjacent axes is computed, and the edges are bundled and rendered through density-based confluent drawing. Finally, users create a more interpretable visualization of edge bundling through the interactions, including splitting, adjusting, and merging clusters.

2.2 Data Binning-Based Clustering

Data binning groups a number of more or less continuous values into a smaller number of given data intervals (also called "bins") to transform numerical variables into their categorical counterparts [12]. Multidimensional binning is used to implement focus + context visualization in PCPs to represent outliers [9]. In this study, we use one-dimensional (1D) binning to cluster data points for each dimension with the following three considerations:

- In PCPs, for a single dimension, the clusters must be ordered because the data points are ordered.
- A data point belongs to only one cluster.
- For large data, to support HITL edge bundling in PCPs, the clustering process must be fast, scalable, and transparent to analysts.

With the first and second considerations, for each axis, the data points are binned into ordered and adjacent clusters, which is shown in Figure 3. Since a data point belongs to only one cluster, there is no overlaps between clusters. This reduces the overplotting of clusters in PCPs created by multidimensional clustering methods, such as DBSCAN [5]. As shown in Figure 3, for each axis, the data points are first grouped into the same number of clusters. For a particular axis, the initial clusters have the same initial diameters. Users then use the control points to split, adjust, and merge clusters (see Section 2.4), which makes the clustering process transparent for analysts. For an axis with k initial clusters (the initial value of k is configured by users), the initial diameter L is computed as:

$$L = (d_{max} - d_{min})/k$$

where d_{max} and d_{min} are the maxima and minima, respectively, of the data points on the corresponding axis. For an axis, the initial control points P_i denotes the boundaries of clusters, which are computed as:

$$P_i = d_{min} + i \times L, i = 1, 2, ..., k-1$$

Then, a data point d is grouped into a cluster C_i as:

$$d \in C_i \text{ if } \begin{cases} P_{i-1} < d < P_i, i = 1, 2, ..., k-1 \\ d > P_{i-1}, i = k \end{cases}$$

To reveal the internal patterns and distribution of data, we compute the density of each pair of clusters and use it for density-based confluent drawing (see Section 2.3). For two adjacent axes $axis_n$ and $axis_{n+1}$, a cluster pair $(C^i_{axis_n}, C^j_{axis_{n+1}})$ consists of a cluster in $axis_n$ and another in $axis_{n+1}$, where $C^i_{axis_n}$ is the i-th cluster in $axis_n$, and $C^j_{axis_{n+1}}$ is the j-th cluster in $axis_{n+1}$. For two adjacent axes, an edge containing two data points (d_n, d_{n+1}) that belongs to a pair of clusters is defined as:

$$(d_n, d_{n+1}) \in (C^i_{axis_n}, C^j_{axis_{n+1}}) \text{ if } d_n \in C^i_{axis_n} \wedge d_{n+1} \in C^j_{axis_{n+1}}$$

The density $D_{i,j}$ of a pair of clusters is computed as:

$$D_{i,j} = \frac{N(C^i_{axis_n}, C^j_{axis_{n+1}})}{\sum_{i=1}^{i} \sum_{j=1}^{j} N(C^i_{axis_n}, C^j_{axis_{n+1}})}, n = 1, 2, ...$$

where $N(C^i_{axis_n}, C^j_{axis_{n+1}})$ is the number of edges that belong to the cluster pair $(C^i_{axis_n}, C^j_{axis_n})$.

The clustering process, including computing the clusters and the density of cluster pairs, is linearly dependent on the number of dimensions, the number of data points, and the number of clusters (see Section 3.1). This fast and scalable clustering process is the basis of real-time interactions (see Section 2.4), which supports HITL edge bundling for large multidimensional data in PCPs.

Categorical variables are not clustered using the above method. Instead, we treat each category as a cluster.

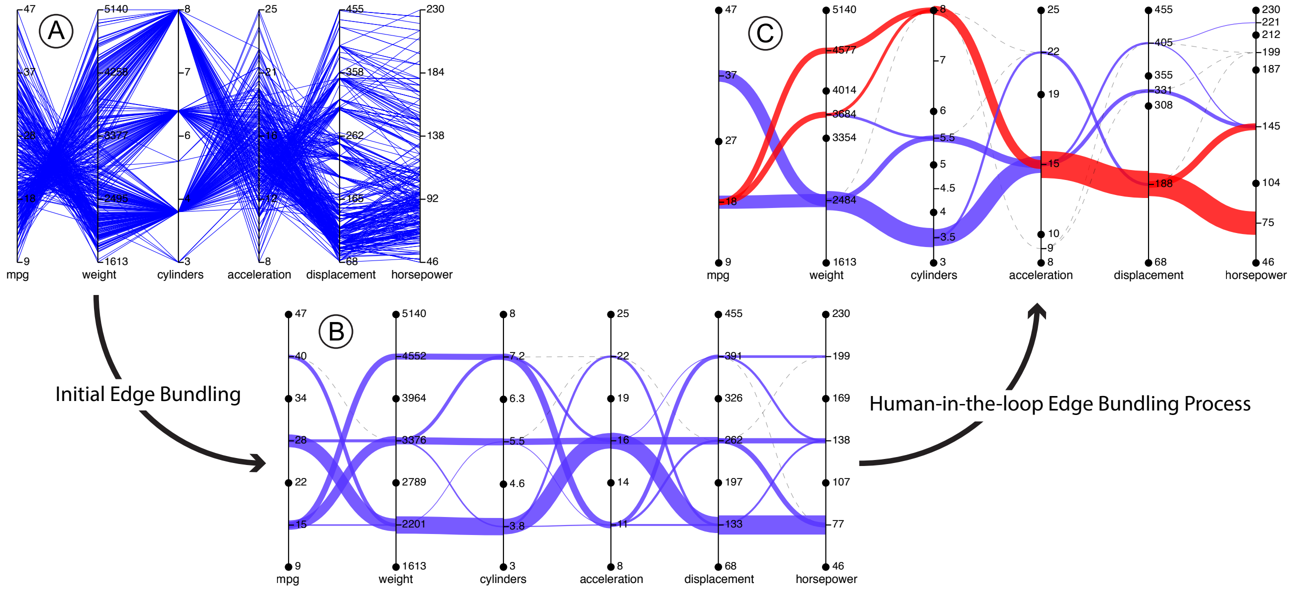

Figure 1: Overview of the system that supports HITL edge bundling in PCPs. A. Visualization of the Cars dataset [1] in a classic PCP. **B.** Edge bundling of the dataset with 3 initial clusters for each dimension. **C.** Interpretable edge bundling of the dataset with a subset highlighted (continuous path over axes) in red, which is generated through user interactions.

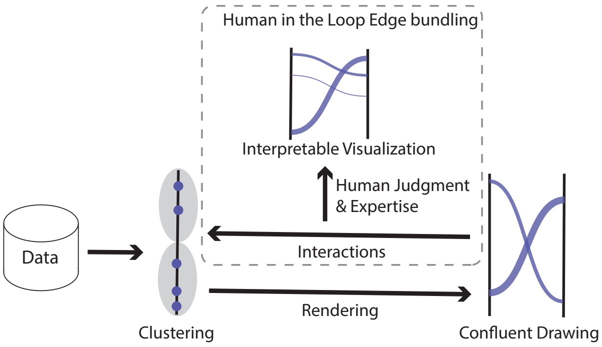

Figure 2: The workflow of the system.

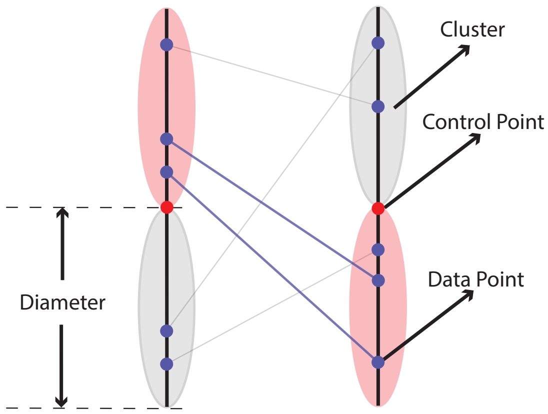

Figure 3: Using 1D binning to cluster data points for each axis in PCPs. The blue points are data points and the red points are control points. An edge between the axes represents two data points that belong to two clusters respectively. Elliptical areas represent clusters in an axis. The initial k is 2. For each axis, the two initial clusters have the same diameter. The two red clusters form a pair of clusters. Its density is 0.4.

2.3 Density-based Confluent Drawing

Confluent drawing is a technique for bundling links in node-link diagrams. It coalesces groups of lines into common paths or bundles based on network connectivity to reduce edge clutter in node-link diagrams [2, 4]. In this study, we use confluent drawing to coalesce edges that belong to a pair of clusters to reduce visual clutter in PCPs, where we use the clusters as nodes and edges between them as links. Each pair of clusters then has only one bundled edge, which is shown in Figure 4. This eliminates the occlusion and ambiguity near the bundle joints created by bundling techniques that bundle edges by spatial proximity. More importantly, it reduces rendering time by coalescing edges, which supports real-time interactions for HITL edge bundling of large multidimensional data in PCPs.

To reveal the information hidden by coalescing of the edges and the distribution of the data points between axes, we use the density $D_{i,j}$ of a pair of clusters $(C^i_{axis_n}, C^j_{axis_{n+1}})$ to define the width $W_{i,j}$ of the coalesced bundle as follow:

$$W_{i,j} = D_{i,j} \times W_{max}$$

where W_{max} is the width of a bundle with the density of one. W_{max} is a constant and is configured by users.

To guarantee C^1-continuity across axes, we draw bundles as Bézier curves. Figure 4 shows the bundled edge of a pair of clusters. Between two adjacent axes, the width of a bundle represents the proportion of the data points (coalesced edges) that belong to the corresponding cluster pair. This reveals the trend and distribution of the data items as well as outliers in large multidimensional data in PCPs (see Section 3.2).

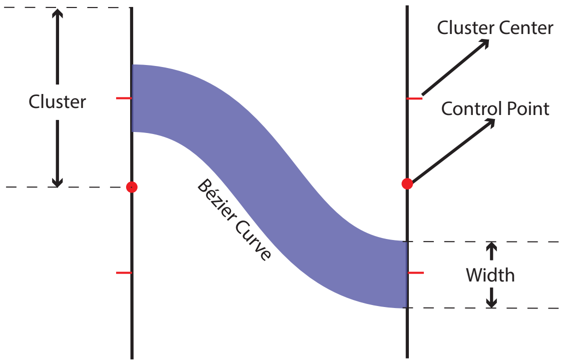

Figure 4: Using the density-based confluent drawing to bundle the edges that belong to a pair of clusters. For a pair of clusters, the bundled edge is rendered as a Bézier curve that starts from the center of a cluster and ends at the center of another. Its width represents the density of the cluster pair.

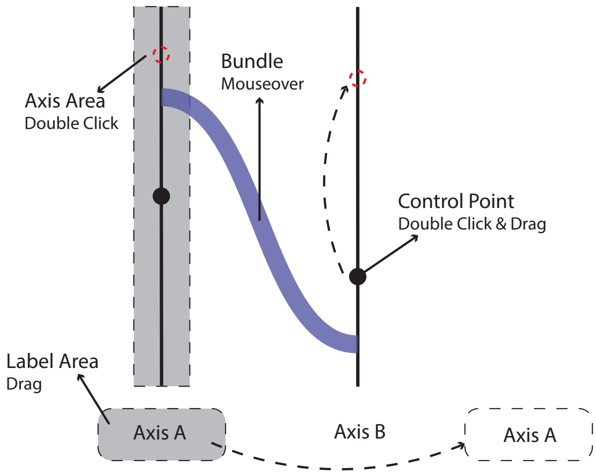

Figure 5: Interactions provided by our system for supporting HITL edge bundling. Double click on the axis area to add a control point to split a cluster. **Double click** on a control point to delete it to merge two clusters. **Drag** a control point along an axis to adjust the adjacent clusters. **Mouseover** on a bundle to highlight a subset with color. **Drag** an axis label to re-order the axes.

2.4 Interactions for HITL Edge Bundling

In our system, in addition to common interactions in PCPs such as re-ordering the axes and brushing (highlighting) [11], we use specifically designed interactions to allow users to split, adjust, and merge clusters. Our system updates the visualization according to user interactions in real time, which is the key to implement the HITL edge bundling process. These interactions are supported by the combination of the data binning-based clustering and the density-based confluent drawing. Figure 5 shows the interactions provided by our system, which are described as follows:

- **Split a cluster.** Each axis has a clickable area (called axis area) around it, which is shown as gray rectangle area around *Axis A* in Figure 5. Double-clicking on this area adds a new control point to the corresponding position on the axis. This control point splits the original cluster into two new clusters. In Figure 5, the red dashed line circle on *Axis A* is a newly added control point by double-clicking.
- **Adjust clusters.** All control points can be dragged along the axes. Dragging a control point to a new position adjusts the boundaries and the diameters of the two adjacent clusters. Figure 5 shows dragging the control point on *Axis B* to a new position (red dashed line circle on *Axis B*).
- **Merge clusters.** All control points can be double-clicked to be deleted. The two adjacent clusters of the deleted control point are merged into a new cluster.
- **Highlight bundles over axes.** Hovering the pointer over a bundle highlights it and its related bundles in red. Only bundles with a density greater than a threshold will be highlighted. The threshold is a constant and is configured by users.
- **Re-order axes.** The labels of axes can be dragged to the front or back of other labels to re-order them to the corresponding positions.

3 EVALUATION

In this section, we evaluate the scalability and the effectiveness of our system through experiments and a case study on the Office Occupancy Detection dataset [3] and the Cars dataset [1].

3.1 Experiments

To examine the scalability of our system, we synthesized several large datasets based on the office dataset. All experiments were conducted on the same laptop without big data infrastructure-based data processing and hardware-accelerated rendering.

In our system, the HITL edge-bundling process contains two time-consuming processes: the data binning-based clustering and the density-based confluent drawing (rendering process). We first performed a run time analysis of the clustering process. Table 1 shows the run times (measured by the second) of the clustering process on large multidimensional datasets (with different number of dimensions, data points, and clusters). According to Table 1, the computation time of data binning-based clustering is linearly dependent on the number of dimensions, the number of data points, and the number of clusters. More importantly, this data binning-based clustering is much faster than other clustering algorithms used for bundling edges in PCPs. For example, Palmas et al. [10] used a density-based clustering method for each dimension independently to bundle edges in PCPs, which takes approximately 60 seconds to cluster 10^5 data points for one dimension. By contrast, our clustering method takes approximately 1 seconds to cluster 10^6 data points for four dimensions.

We then examined the efficiency of the rendering process by comparing the rendering time of our method with both the classic PCP and Lima et al.'s edge-bundling PCP [5] that also uses confluent drawing to coalesce edges. To compare the rendering time, all three PCPs were implemented with the same JavaScript library (D3.js) and rendered in Chrome. The times needed for rendering the axes, labels, and stickers were not included, which are constant regardless of the number of data points. Table 2 shows the rendering time of the three methods (measured by the second) on the datasets that has six dimensions and the different numbers of data items. For our method and [5], each dimension has 3 clusters. According to Table 2, the classic PCP and [5] take 1.7672 and 3.6989 seconds to visualize 10^5 data points. The classic PCP takes 8.7183 seconds to visualize 5×10^5 data points

Table 1: Run-time analysis of the data binning-based clustering

Dimensions	Data Points	Clusters	Run-time
2	10^4	3	0.0169
2	10^4	4	0.0167
3	10^4	3	0.0230
3	10^4	4	0.0277
2	10^5	3	0.0505
2	10^5	4	0.0554
3	10^5	3	0.0937
3	10^5	4	0.0996
4	10^5	3	0.1175
4	10^5	4	0.1404
4	10^5	10	0.2574
4	10^5	20	0.4139
4	10^5	30	0.5495
4	10^5	40	0.6892
4	10^5	50	0.8872
4	10^6	3	0.8211
4	10^6	4	0.9398

Table 2: Comparison of the rendering time

Data Points	Our Method	Classic PCP	[5]
10^3	0.00243	0.0273	0.0503
10^4	0.00231	0.1916	0.3740
10^5	0.00230	1.7672	3.6989
5×10^5	0.00229	8.7183	N/A
10^6	0.00248	N/A	N/A

and crashes the browser when visualizing 10^6 data points. The method [5] crashes the browser when visualizing 5×10^5 data points. By contrast, the rendering process of our method is independent of the number of data points, which takes approximately 0.002 seconds for each dataset.

3.2 Case Study

To assess the effectiveness of our system, we compared our method with the classic PCP and several algorithmic analysis methods with the office dataset. The office dataset uses the data on temperature, humidity, light, and CO_2 to detect the occupancy of an office room. It has five dimensions and 20,560 data points for each dimension.

Figure 6 shows the visualization of the office dataset in the classic PCP and our system. Figure 6c shows the visualization in our system, which is generated by a user who does not have knowledge of the dataset. In Figure 6b and Figure 6c, the red bundles are the subsets highlighted by hovering the pointer on the widest bundle between the axes of *light* and *occupancy*. The extreme narrow bundles (data points with extreme low densities) are visualized as the dashed lines to detect and highlight the outliers (rare data points that raise suspicions by differing significantly from the majority of the data [8]) in the dataset. By comparing Figure 6a and and Figure 6c, it is clear that for large multidimensional datasets, our method reduces the visual clutter and overplotting in the classic PCP and reveals the patterns in the data.

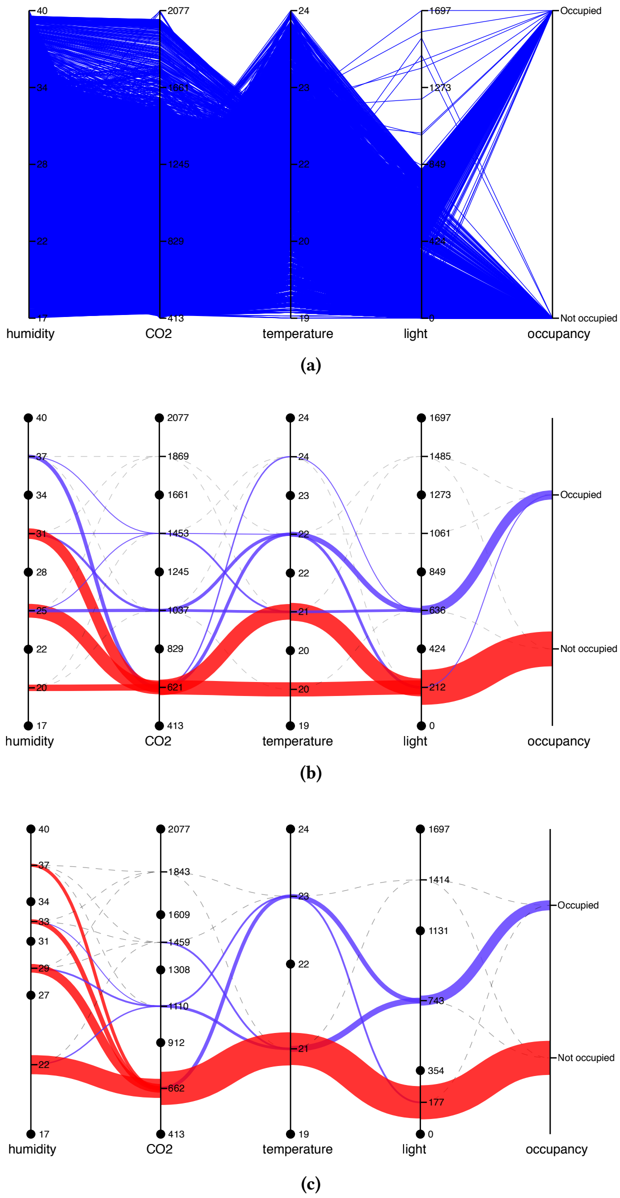

Figure 6: The visualization of the office dataset in the classic PCP and our system. (a) Visualization of the office dataset in the classic PCP. **(b)** Visualization of the office dataset in our system with 4 initial clusters for each dimension. **(c)** Visualization of the office dataset in our system generated by a user who does not have knowledge of the dataset.

Moreover, by integrating human judgments into the edge-bundling process, our method creates a interpretable visualization in PCPs for the office dataset. For example, during the HITL edge-bundling process (from Figure 6b to Figure 6c), the user obtained the following findings:

- **Finding 1.** The dataset contains outliers which are highlighted by the dashed lines in Figure 6c.
- **Finding 2.** When the value of light is smaller than 354 Lux, the room is considered unoccupied. When it is between 354 and 1131 Lux, the room is considered occupied. The accuracy of this estimation is higher than 90% (the

Table 3: The comparison our system with the algorithmic methods in [3].

Criteria	Our Method	[3]
Finding 1	Yes	No
Finding 2	Yes	Yes
Finding 3	Yes	Yes
Finding 4	Yes	Yes
Interpretability	Interpretable visualization with transparent clustering process.	Black-box process of training the models.
Processing time	Real-time.	Time for training and selecting models.

estimated sum of the densities of the two widest bundles between the axes of light and the occupancy).

- **Finding 3.** When the temperature is between 19 and 22 ℃, the room is considered unoccupied. When the temperature is higher than 22 ℃, the room is considered occupied. The accuracy of this estimation is higher than 80% (the estimated sum of the densities of the two widest bundles between the axes of temperature and light).
- **Finding 4.** Using all features may reduce the accuracy of prediction. Humidity has a much weaker correlation with occupancy than other features.

Candanedo and Feldheim tested linear discriminant analysis, classification and regression trees, and random forest on the office dataset to detect the occupancy of rooms [3]. In Table 3, we compared the findings obtained in our system with that obtained in [3] of the office dataset. It shows that our system obtained more findings of the data than the algorithmic methods in [3]. We also compared the interpretability of our system with that of the algorithmic methods in [3]. It shows that without the black-box process of training the models, our system is more interpretable with the visualization by integrating human judgments into the edge-bundling process. Moreover, our system can obtain the result faster by eliminating the time to train the models.

3.3 Discussion

Our approach uses data binning to create initial clusters for each dimension. For a particular dimension, it divides the entire range of values into a series of consecutive, non-overlapping and equal-size intervals (clusters/bins). By computing the density of cluster pairs, our approach counts the number of data points for each cluster, which is represented by the total width of the bundled edges starting from the cluster. Therefore, the initial clustering results in our approach is an adapted histogram for each dimension. With the appropriate initial number of clusters, it can capture the accurate distribution of data points for each dimension. This is the basis for users to use their judgments and expertise in the edge bundling process and generate interpretable visualization. With HITL edge bundling, to obtain the final interpretable visualization, for example, from Figure 6b to Figure 6c, users may need several iterations to adjust the initial clusters for each dimension, such as merging a cluster with small density to an adjacent cluster, or splitting a cluster with large density to obtain more details of data. This process may take 1 or 2 minutes. However, during this process, users can continuously gain insights from data and visualization.

4 CONCLUSION AND FUTURE WORK

In this study, we proposed HITL edge bundling and built a system based on it to support the visual exploration of large multidimensional data in PCPs. The system provides an interpretable visualization, which reduces the visual clutters and overplotting, and eliminates the occlusion and ambiguity of large multidimensional data in PCPs. More importantly, the system provides the specifically designed interactions, including splitting, adjusting, and merging clusters, to integrate human judgments into the edge-bundling process in real time. We evaluated the scalability and effectiveness of the system through experiments and a case study. We compared our system with the classic PCP and the algorithmic analysis methods. The results show that our system provides a scalable and interpretable way of visually exploring large multidimensional data in PCPs.

Anchoring bundled edges in different positions, such as the mean/centroid position of all data points in a cluster, could be investigated in the future to improve the continuity across axes and reveal more information of clusters. This requires more computation and may delay the visual response of the interactions. The interactions and color effects (highlighting subsets in different colors) of the system are not fully evaluated. This can be done in a qualitative user study in future work.

REFERENCES

[1] 2005. Cars DataSet. Retrieved September 20, 2019 from http://davis.wpi.edu/xmdv/datasets/cars.html

[2] B. Bach, N. H. Riche, C. Hurter, K. Marriott, and T. Dwyer. 2017. Towards Unambiguous Edge Bundling: Investigating Confluent Drawings for Network Visualization. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 541–550. https://doi.org/10.1109/TVCG.2016.2598958

[3] Luis M. Candanedo and Véronique Feldheim. 2016. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. *Energy and Buildings* 112 (2016), 28 – 39. https://doi.org/10.1016/j.enbuild.2015.11.071

[4] Matthew Dickerson, David Eppstein, Michael T. Goodrich, and Jeremy Y. Meng. 2005. Confluent Drawings: Visualizing Non-planar Diagrams in a Planar Way. *Journal of Graph Algorithms and Applications* 9, 1 (2005), 31–52. https://doi.org/10.7155/jgaa.00099

[5] Rodrigo Santos do Amor Divino Lima, Carlos Gustavo Resque dos Santos, Sandro de Paula Mendonça, Jefferson Magalhães de Morais, and Bianchi Serique Meiguins. 2018. Understanding Data Dimensions by Cluster Visualization Using Edge Bundling in Parallel Coordinates *(SAC '18)*. ACM, New York, NY, USA, 640–647. https://doi.org/10.1145/3167132.3167203

[6] Julian Heinrich and Daniel Weiskopf. 2013. State of the Art of Parallel Coordinates. In *Eurographics 2013 - State of the Art Reports*, M. Sbert and L. Szirmay-Kalos (Eds.). The Eurographics Association. https://doi.org/10.2312/conf/EG2013/stars/095-116

[7] D. Holten. 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (Sep. 2006), 741–748. https://doi.org/10.1109/TVCG.2006.147

[8] Ling Liu and M. Tamer Zsu. 2009. *Encyclopedia of Database Systems* (1st ed.). Springer Publishing Company, Incorporated.

[9] M. Novotny and H. Hauser. 2006. Outlier-Preserving Focus+Context Visualization in Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (Sep. 2006), 893–900. https://doi.org/10.1109/TVCG.2006.170

[10] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauf. 2014. An Edge-Bundling Layout for Interactive Parallel Coordinates. In *2014 IEEE Pacific Visualization Symposium*. 57–64. https://doi.org/10.1109/PacificVis.2014.40

[11] R. C. Roberts, R. S. Laramee, G. A. Smith, P. Brookes, and T. D'Cruze. 2019. Smart Brushing for Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics* 25, 3 (March 2019), 1575–1590. https://doi.org/10.1109/TVCG.2018.2808969

[12] Bernard W Silverman. 2018. *Density estimation for statistics and data analysis.* Routledge.