Cluster-based Visual Abstraction for Multivariate Scatterplots

Hongsen Liao, Yingcai Wu, Li Chen and Wei Chen

Abstract—The use of scatterplots is an important method for multivariate data visualization. The point distribution on the scatterplot, along with variable values represented by each point, can help analyze underlying patterns in data. However, determining the multivariate data variation on a scatterplot generated using projection methods, such as multidimensional scaling, is difficult. Furthermore, the point distribution becomes unclear when the data scale is large and clutter problems occur. These conditions can significantly decrease the usability of scatterplots on multivariate data analysis. In this study, we present a cluster-based visual abstraction method to enhance the visualization of multivariate scatterplots. Our method leverages an adapted multilabel clustering method to provide abstractions of high quality for scatterplots. An image-based method is used to deal with large scale data problem. Furthermore, a suite of glyphs is designed to visualize the data at different levels of detail and support data exploration. The view coordination between the glyph-based visualization and the table lens can effectively enhance the multivariate data analysis. Through numerical evaluations for data abstraction quality, case studies and a user study, we demonstrate the effectiveness and usability of the proposed techniques for multivariate data analysis on scatterplots.

Index Terms—Data abstraction, scatterplot, glyph visualization, multilabel optimization.

✦

1 INTRODUCTION

The use of scatterplots is one of the most important and popular multivariate data visualization methods. The points on the scatterplot can help reveal underlying patterns in data, such as variable relationships and possible data clusters. Attributes of the points, such as point size and point color, are usually used to encode data values. Users can obtain an overview of the data through these encodings and derive conclusions from the data. Various projection methods, such as principal component analysis (PCA), multidimensional scaling (MDS), and t-SNE [1], can be used to construct a scatterplot from a multivariate dataset. These projections can provide useful illustrations for multivariate datasets and enhance multivariate data analysis on scatterplots.
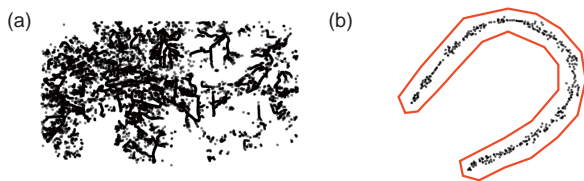

Fig. 1. Example of scatterplots. (a) Scatterplot with clutter problems. (b) Projection of the UCI Auto-MPG dataset using t-SNE.

Although scatterplots have shown great usability in multivariate data analysis, analyzing multivariate data on a scatterplot, especially a cluttered one, is still challenging. First, obtaining a clear view of multivariate data distribution on a scatterplot or intuitively explore the data is difficult, especially when the number of variables increases. A series of color mappings for different variables, which are utilized in many previous applications, can be used to demonstrate the multivariate data distribution. However, users cannot easily and simultaneously obtain a clear view of several color mappings and compare them. Second, the clutter problem occurs when the data scale becomes large or the projected points are extremely close, as shown in Figure 1(a). Then, directly obtaining a clear view of the data distribution is nearly impossible even with proper encodings for variable values, because many points are covered and indirectly rendered. Third, no effective solution is available to analyze a shape of interest (SOI, a set of points) on a scatterplot generated using multivariate projection methods. For example, the scatterplot in Figure 1(b) is generated from the UCI Auto-MPG dataset using t-SNE. To he best of our knowledge, no intuitive solution is currently available to analyze the SOI, as indicated by the red curve. Such solution is necessary to help users intuitively analyze the underlying data relationships conveyed by the SOIs.

In this work, we propose to use cluster-based level of detail (LOD) abstractions in exploring and analyzing multivariate data. An abstraction, which comprises a set of data clusters, is used to enhance the understanding of a multivariate scatterplot, in addition to simple scatter points. The abstraction can briefly illustrate the multivariate data distribution on the scatterplot, and the LOD design can help users intuitively and effectively explore the scatterplot, even a cluttered one. This abstraction is generated using an adapted multilabel optimization based clustering [2] for the data points. This method outperforms previous solutions, such as hierarchical clustering and normalized cuts, in providing abstractions of high quality. To deal with large data problems, an image-based speedup method is proposed to improve the clustering method, which ensures its flexibility and usability in different applications. Furthermore, a pair of glyphs is compared and selected to visualize data clusters. Data distribution information, such as data mean

- *Hongsen Liao and Li Chen are with School of Software, also with Tsinghua National Laboratory for Information Science and Technology, Tsinghua University.*
  E-mail: liaohs082@gmail.com, chenlee@tsinghua.edu.cn.
- *Li Chen is the corresponding author.*
- *Yingcai Wu, Wei Chen are with State Key Lab of CAD & CG, Zhejiang University. Zhejiang University.*
  E-mail: {ycwu, chenwei}@cad.zju.edu.cn

and standard deviation (SD), is encoded in the glyph to provide an overview of a cluster of points or compare among clusters. Users can either clearly achieve an overview of the multivariate data distribution or explore the detailed data under the guidance of the glyph. This procedure can assist users in efficiently exploring multivariate data, even on a cluttered scatterplot. Moreover, we propose to combine the glyph-based abstraction with the table lens. The view coordination between them can effectively help users analyze the possible data relationships represented by SOIs on the scatterplot. These possible data relationships can work as initial guidelines for highly detailed data analysis. Evaluations including a numerical evaluation for the data abstraction quality, a set of case studies and a user study are used to demonstrate the effectiveness and usability of the proposed techniques.

The main contributions of our work include the following.

1) An image-based multilabel clustering method is proposed to generate LOD abstractions for multivariate data points on a scatterplot. This method can provide data abstractions of high quality and support interactive LOD explorations even for large scale data.

2) A pair of glyphs is compared and selected to enhance the LOD data exploration. Users can intuitively explore the multivariate data under the guidance of the glyphs. A user study is carried out to validate the usability of the glyphs in data exploration.

3) View coordination between the table lens and the glyph-based scatterplot is proposed to enhance multivariate data analysis. Users can then efficiently interpret variable relationships represented by SOIs on a scatterplot.

2 RELATED WORK

2.1 Multivariate Data Visualization

Multivariate data visualization has attracted considerable attention in the past. Liu et al. recently provided a detailed survey on the advances in high-dimensional data visualization [3]. Various methods have been implemented in different application scenarios. In these methods, scatterplot and scatterplot matrix are widely used to analyze the relationship between two variables or among multiple variables. Keim et al. enhanced scatterplots by proposing a series of visualization solutions, such as generalized scatterplots [4], scatterplots enhanced by ellipsoid pixel placement and shading [5], and variable binned scatterplots [6]. Mayorga et al. presented the splatterplot to overcome the overdraw problem [7]. These solutions can effectively improve the usability of scatterplots. Other popular methods, such as parallel coordinate [8], star coordinate [9], Radviz [10], and table lens [11], can also be used to show the relationship pattern for high-dimensional data. Many extensions of the aforementioned methods have been proposed to enhance multivariate data analysis. For example, Muller et al. attempted to guide users in exploring scatterplots under different projections of high-dimensional data [12]. They also developed a data context map, which uses an iso-surface along with the scatterplot, to help users achieve a clear view of multivariate data [13]. Yuan et al. combined a parallel coordinate with the scatterplot to help analyze high-dimensional data [14]. However, exploring the multivariate data on a scatterplot still lacks effective interactions apart from color mapping and brushing. In our work, we provide a glyph-based LOD visualization and intuitive interactions to explore scatterplots. Combined with table

lens and parallel coordinate, the scatterplot can effectively help users analyze multivariate data.

In recent years, many studies on multivariate data exploration based on scatterplots have been conducted. Lehmann et al. proposed a dissimilarity maximization-based method to generate a series of scatterplots, which can comprehensively visualize multivariate data [15]. Kim et al. developed an intuitive system for users to interactively generate projections that fulfill their visualization requirements [16]. Liu et al. explored high-dimensional data on the basis of subspace analysis and dynamic projections, and they used a navigation graph to guide users in viewing scatterplots under different projections [17]. In general, these techniques attempt to visualize multivariate data and present the underlying multivariate relationship by generating a series of scatterplots. Unlike these previous works, our work focuses on enhancing visualization on a single scatterplot. Thus our work can be smoothly integrated into the aforementioned works to enhance multivariate data exploration on the basis of a series of projections.

The use of glyphs is a widely used method to visualize multiple variables. Borgo et al. comprehensively surveyed state-of-the-art glyph-based visualization works [18]. The visual channels of a glyph, such as shape, color, and location, are used to encode the variables [19]. For example, radar glyphs are widely used to visualize multi-dimensional data [20]. In our visualization, we use a design similar to the radar glyph and a band design in accordance with our visualization requirements.

2.2 Data Abstraction in Multivariate Data Visualization

Viewing the multivariate data becomes difficult as the dimension number of data or data scale increases. Therefore, data abstraction is necessary to achieve an effective visualization of multivariate data. Different data abstraction methods, as well as the corresponding quality metrics [21], have been proposed in high-dimensional data visualization. A systematized survey on these methods was conducted by Bertini et al. [22]. In these methods, sampling and clustering are extensively used in different application scenarios.

Bertini et al. used a non-uniform sampling method to reduce visual clutter for scatterplots [23]. Ellis et al. proposed several methods for measuring occlusion and supported interactive sampling based on the said methods to reduce visual clutter in the parallel coordinate [24]. Chen et al. [25] proposed to use the multiclass blue noise sampling for the visual abstraction of a multiclass scatterplot.

Clustering is used in various visualization methods and applications. Tree maps and dendrograms can be generated from a hierarchical data structure based on clustering. Kreuseler et al. proposed a scalable framework for information visualization based on hierarchical clustering, and they provided different methods to visualize hierarchies [26]. Clustering has also been widely used in vector field visualization to achieve an abstracted rendering of vector fields. For example, Heckel et al. generated a visualization result by splitting clusters [27]. Telea et al. clustered a vector field by merging data based on an elliptic similarity evaluation [28]. Du et al. provided a visualization solution in accordance with Voronoi regions to enhance the connection between glyphs and data [29].

In our work, we use glyphs to provide a visual abstraction for a multivariate scatterplot. Our method is inspired by the glyph-based visualization for a vector field, which can help users intuitively explore multivariate data.
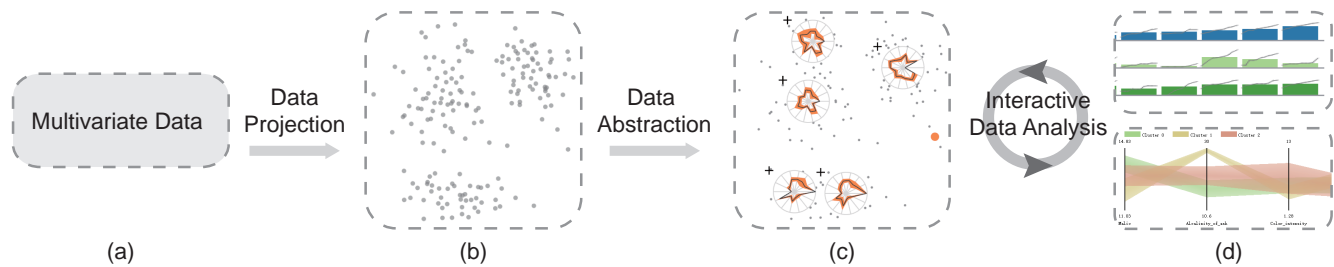
Fig. 2. Overview of the system pipeline. After a multivariate dataset is loaded (a), the data are initially projected to generate the scatterplot (b). Then, the multilabel optimization is utilized to generate the LOD data clusters. Glyphs are used to visualize and analyze the clusters (c) along with a suite of coordinated views (d).

2.3 LOD Visualization

LOD visualization is an important strategy for visualizing data that cannot be fully conveyed with a single view. For example, Fua et al. used a structure-based brushing tool to select the data abstraction level for multivariate data based on hierarchical clustering [30]. Peng et al. used a hierarchical clustering method on a mesh to generate a LOD rendering for a vector field [31].

Much research has been conducted on LOD visualization. In these works, LOD visualization for graphs is one of the most popular topics. Zinsmaier et al. took advantage of edge cumulation to provide an interactive LOD visualization for large graphs [32]. Balzer et al. used implicit surfaces to visualize a clustered graph [33]. Abello et al. provided an architecture to address large scale hierarchical graphs and support interactive exploration on graphs [34]. Different layout methods, such as planar layout, spring layout, and tree layout, were proposed to visualize graphs effectively [35]. The spanning tree of a graph can be extracted to help analyze the relationship among the nodes in the graph [36]. Interaction techniques, such as zoom and pan, focus + context and incremental exploration, can be used to provide navigations for data exploration [37].

In our work, we adopt the LOD design for the abstraction of a scatterplot. We consider scatter points as nodes on a graph in the multilabel optimization. Glyphs are used to provide an overview of the graph, which can guide users explore the graph and multivariate data smoothly.

3 SYSTEM OVERVIEW

The data analysis pipeline of the system is illustrated in Figure 2. It comprises the following three main steps.

Data Projection The goal of our method is to assist multivariate data analysis on a scatterplot. Thus, the first step of the pipeline is to project a multivariate dataset onto a 2D scatterplot. Different methods can be used depending on the data type. For geospatial data, the points are usually distributed on a map according to their longitude and latitude attributes. For other numerical multivariate data, the scatterplot can be generated by conventional low-dimensional embedding approaches, such as PCA, MDS and t-SNE.

Data Abstraction Instead of simple scatter points, we attempt to provide a brief illustration of the multivariate data distribution on the scatterplot for users. This way can help users easily understand and analyze the data. An image-based multilabel clustering method is used to generate LOD abstractions for the multivariate scatterplot from the previous projection. The abstractions can

be generated using a view-dependent clustering or a top-down clustering, which will be detailed in Section 4.

Interactive Data Exploration and Analysis Based on the data abstraction, the following three main kinds of data exploration tasks are supported in the system:

1) **Data overview** helps users obtain an initial understanding of the data. Data abstractions are visualized by the glyph on the scatterplot. Each glyph visualizes the data information of a cluster. Based on the glyph, users are allowed to interactively obtain an overview of the data.

2) **Detecting clusters of interest** enables users to efficiently focus on data subset that may need special attention during data exploration. Two kinds of clusters are considered in the system. The first clusters are those with high SDs and need to be further explored, while the second ones present mean values that differ much from those of others. They may be outliers or the ones that require further analysis. In the system, a pair of glyphs is designed to assist users in detecting clusters of interest.

3) **Variable relationship analysis** is an important task for multivariate data analysis. The view coordination between the glyph-based scatterplot and the table lens is provided to assist users in analyzing possible data relationships conveyed by SOIs on the scatterplot. A parallel coordinate is provided to show detailed data.

Using this pipeline, users can efficiently analyze the multivariate data and obtain intuitive guidance for further data analysis.

4 DATA ABSTRACTION

An image-based multilabel optimization method is used in our system to generate LOD abstractions for the scatterplot.

4.1 Adapted Multilabel Optimization

Multilabel optimization is a graph-based method in which each node in the graph is assigned with a label. The nodes that share the same label form a possible cluster [38]. In this method, the number of independent clusters into which the entire graph should be segmented is determined through graph cuts. Compared with the hierarchical binary structure used in some previous studies, the multilabel optimization method can adaptively determine the appropriate cluster number for each level in the data hierarchy. This scenario is natural for most clustering problems, given that a dataset can be typically divided into more than two classes.

The multilabel optimization method works with a set of graph nodes P and a finite set of labels L. Three types of costs, namely,

data, smooth and label costs, are introduced to determine which label should be assigned to each node. Once the label for each node is assigned, the nodes that share the same label can be alternatively visualized by glyphs (Section 5.1).

The points on the scatterplot are used in our system as graph nodes. Delaunay triangulation is applied to generate the graph structure G for the scatterplot nodes. This method is selected to ensure the continuity of node labels on the scatterplot, which the more commonly used k-nearest neighbor method fails to do. Thereafter, candidate labels should be evaluated for the graph. An intuitional method for this evaluation is to set up an individual candidate label for each node p in the graph. Then the label can be assigned to node p or to nearby nodes. In this case, we expect to determine a possible graph partition L_p that centers at node p for each label. In our implementation, the possible partition is defined as the subgraph within a specified distance to the node. Thus, a label partition is defined as:

$$L_p = \bigcup \{q | d(p,q) < t, q \in G\} \tag{1}$$

where $d(p,q)$ is the Euclidean distance of point nodes p and q; t is the distance parameter for the view dependent control, which can be defined depending on screen resolution and user requirement. In our implementation, t is evaluated using

$$t = (\frac{S}{N_g})^{0.5} \tag{2}$$

where S is the size of the bounding box for the points; and N_g is the expected number of clusters for the abstraction, which can be set according to user requirement. In order to maintain the continuity of a partition, we search for the possible partition from p and extend it to the neighboring nodes that satisfy the distance constraint until no new extension can be found. After all the label partitions are evaluated, the three costs are obtained and defined as follows.

Data Cost Data cost $D_p(f_p)$ penalizes the data difference between point p and other points in label f_p. It ensures that the point is assigned with a label in which the majority of the points' data are similar to its. Thus, the data cost is defined as follows:

$$D_p(f_p) = \frac{1}{N_p} \sum_{q \in L_p} \|V_p - V_q\| \tag{3}$$

where V_p and V_q are the multivariate data of points p and q, respectively; N_p is the number of points in the label partition L_p.

Label Cost Label cost penalizes the appropriateness of a label. This cost assigns a penalty for each label, and large penalties are assigned to labels with large data SD. Therefore, labels with small data SD present high probability to remain in the final result. In our system, the label cost is defined as follows:

$$F_L = \alpha N \sum_{l \subseteq L} h_l \cdot \delta_l(f) \tag{4}$$

$$\delta_l(f) = \begin{cases} 1 & \exists p : f_p = l \\ 0 & otherwise \end{cases} \tag{5}$$

where N is the number of point nodes, α is a ratio value, l is a label, h_l is the weighted summary of the variable SD in each label partition, and $\delta_L(f)$ is the indicator function, We add a small value of 0.01 to this cost in our implementation to avoid a nearly zero cost for a label. In practice, α can be used to control the number of the remaining labels. The larger the value is, the smaller the

number of preferred labels are. Additional detailed discussion on how this cost will work on real dataset is provided in Section 6.

Smooth Cost Smooth cost V_{pq} measures whether two adjacent nodes (p and q) should be assigned with the same label. This cost provides a penalty in assigning different labels to neighboring nodes. In our implementation, we use a simple definition for this cost, which is a constant for all adjacent nodes. The constant is evaluated with the average difference values between adjacent nodes. Additional complex definitions for this cost can be found in [38].

The multilabel optimization aims to minimize the following energy function:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{pq \in N} V_{pq}(f_p, f_q) + \alpha N \sum_{L \subseteq L} h_L \cdot \delta_L(f) \tag{6}$$

The solution for this optimization assigns a label to each point, and the points that share the same label are grouped into a cluster. Then, these clusters are used as the data basis for the glyph visualization in Section 5.

4.2 Image-Based Processing for Large Scale Data

A problem with the adapted multilabel optimization is its performance in dealing with large scale data. Processing a graph with a large number of point nodes is time consuming, which is unacceptable in an interactive visualization system for multivariate data exploration. In our work, an image-based processing procedure is introduced to accelerate the optimization for large scale data. The scatterplot is first transformed into an image with a high resolution; for example, the height of the image can be set to 1000 and the width can be set depending on their relative size ranges. Points on the scatterplot are then mapped into pixels in the image. These pixels, along with the average variable values of their corresponding points, are used for the optimization instead of the original points. However, the pixel number can also be very large when the resolution of the image increases. Inspired by the recent research for image processing in computer vision, superpixel is used in our system to further accelerate the optimization. Particularly, we utilize the simple linear iterative clustering (SLIC) superpixel generation method because of its effectiveness and computing efficiency [39]. Thereafter, the superpixels are used as the graph nodes for the optimization. The computing time for a single run of optimization can then be controlled by assigning the expected number of superpixels for the SLIC superpixel generation. For example, the computing time will be less than a half second if the expected number of superpixels is set to 500. This setting works effectively in all our experiments.

In our system, the image-based processing is applied when the number of points for the optimization is large, such as 2000 in all our cases. Otherwise, the original points and variable values are directly used.

4.3 View-Dependent and Top-Down Clustering

Two types of clustering are supported in our system, namely, view-dependent clustering and top-down clustering. In view-dependent clustering, data points, which are shown in the viewport during user interactions, are utilized for the multilabel optimization. In that case, the abstraction for the scatterplot will be updated each time after user interactions, such as zooming and panning. We also support top-down clustering to construct a data hierarchy for a multivariate scatterplot. The clustering is first applied on the

entire dataset, and then iteratively on the generated clusters. Users can specify an SD threshold for the cluster data. As long as the SD of a cluster is greater than the threshold, the multilabel clustering will be recursively applied. A data hierarchy for the multivariate scatterplot can then be generated. Thereafter, the system queries for the expected level of abstraction in the data hierarchy in accordance with user interaction, and the corresponding cluster data are visualized. In our implementation, we initially map a screen distance (e.g., $150px$, which depends on the glyph size) to the point distance. We then query for the level of clusters whose average distance among cluster centers is closest to the mapped distance.

Ip et al. [40] attempted to segment a 2D image with the normalized cuts (NCuts) algorithm to interactively explore volume data on 2D intensity histograms. In our system, we follow their design of hierarchical exploration. Moreover, we provide the glyph as the visual guidance for the exploration. The mean and SD of the data are encoded in the glyph to provide visual clues for the data in the cluster. Users can decide whether they should explore thoroughly into the cluster depending on the glyph. This scenario is similar to the usual manner in which users conduct a data analysis. We also demonstrate that the results generated using the adapted multilabel optimization exhibit better data abstraction quality than the NCuts for multivariate data in the evaluation (Section 6).

5 VISUALIZATION AND EXPLORATION

In this section, we first introduce the glyph design for a multivariate data cluster and then provide details of the view design in the system.
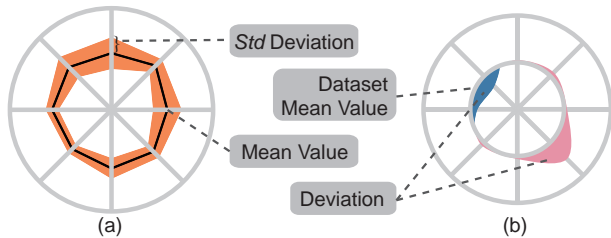
5.1 Glyph

Fig. 3. Glyph design. Glyphs are used to visualize a cluster of multivariate point data.

Glyphs are used to visualize multivariate data in clusters generated via the multilabel optimization (Section 4). They are expected to directly support two main tasks during data exploration and analysis, namely, data overview and detecting clusters of interest, as mentioned in Section 3. However, effectively supporting the two tasks simultaneously with a single glyph is difficult. Thus, a pair of glyphs is used in the system, as illustrated in Figure 3.

The glyph in Figure 3(a) is mainly used to help users obtain an overview of the data and detect clusters with high SDs. Each glyph visualizes the mean and standard (Std) deviation of a data cluster. Each axis of the glyph represents a variable in the data. Mean values of a cluster are visualized by a line that connects all the axes, and corresponding SDs are encoded by the half width of the orange band on the axes. Users can obtain a clear view of the mean values and SDs through the glyph. Moreover, they can easily

detect clusters with high standard deviations through the orange band. However, the band can be an obstacle for users and even lead to misunderstandings when users compare the mean values of different clusters.

The glyph in Figure 3(b) is mainly used to assist users in detecting clusters of interest with mean values that differ much from others. The commonly used radar glyph can also assist users in observing the differences among clusters and detecting clusters whose mean values differ from others. However, when the differences are small, the shape of the radar glyph can no longer effectively support the detection. Thus, we employ an alternative design for this task, which is the Z-Glyph provided by Cao et al. [41]. Color and area, instead of shape, are used to represent the differences. In our system, we first calculate the deviations between the cluster and dataset mean values. Then, the deviations are directly visualized in the glyph. Similar to the glyph in Figure 3(a), each axis represents a variable. However, we use an extra circle to represent the dataset mean values, as shown in Figure 3(b). Then, a curve is used to indicate the deviation values. A radius that is smaller than the extra circle indicates a negative deviation; otherwise, the deviation is positive. Moreover, color is used to enhance the visualization. Blue represents negative deviations and red represents positive ones. Users can intuitively compare among clusters and detect clusters of interest through the differences in curves and filled colors. For example, it still needs some time for users to compare among different glyph shapes and determine their differences in Figure 4(a). However, we can efficiently infer the differences from the curves and colors in Figure 4(b).

A series of glyph alternatives is considered during the development, as shown in Figure 6. We mainly compare two types of glyph designs, namely, the bar chart-based and radar chart-based glyphs. These two types of glyphs are widely used for visualizing multivariate data. In the bar chart based glyphs, as shown in Figures 6(a) and (c), each bar represents a variable. The height of a black line in Figure 6(a) is used to encode the mean value of a variable, and the half height of the bar encodes the corresponding SD. The glyphs in Figures 6(c), (d), (e) and (f) are used to visualize the deviations between cluster and dataset mean values. Independent bars and linked lines are used to visualize the deviations in Figures 6(c) and (d). By contrast, the Z-Glyph uses smoothed curves and colors to visualize the deviations, as shown in Figures 6(e) and (f). A user study is carried out to compare the usability of the glyphs and validate the rationality of our glyph choices (Section 6.6).

5.2 Coordinated Views

We design the glyph-based scatterplot (Figure 5(a)) to help users smoothly explore multivariate data. Other coordinated views, including a table lens (Figure 5(b)) and a parallel coordinate (Figure 5(c)), are provided to promote data exploration and analysis.

5.2.1 Glyph-based Scatterplot

The glyph-based scatterplot is a two-layer view, which includes one for the original points and the other for the glyphs (Figure 5(a)). The glyph layer for clusters is rendered above the point layer. If a cluster contains child clusters in the data hierarchy generated using the top-down clustering, then a plus indicator is shown at the top left of the glyph, as indicated by **a1** in Figure 5(a). This indicator directly provides clues for users whether they can
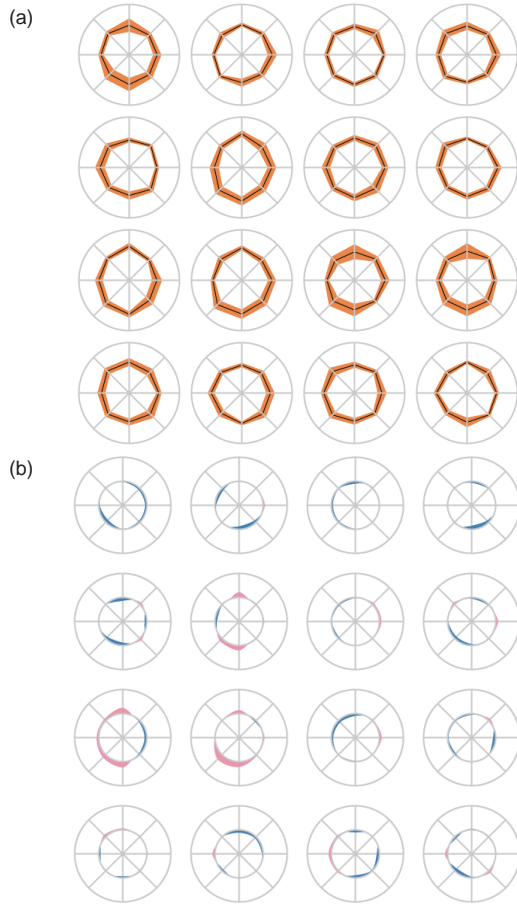
(a)

(b)

Fig. 4. Glyph Comparison. (a) Radar glyph. (b) Z-Glyph.

zoom in to obtain a detailed view of the data. For clusters without child clusters, users can also choose to split the cluster and thus obtain more detailed subclusters. When the point number of a cluster is excessively small (e.g., less than 5), a circle representation is used to identify small clusters, as indicated by the black arrows in Figure 12(b). The tooltip indicated by **a3** provides clues for the number of points represented by a full outer circle. Users can relatively evaluate the number of points in each of the cluster through the angle of the corresponding outer circles. Moreover, the visibility and color encoding for each variable in the glyph can be controlled through an additional panel, as shown in Figure 5(d). Furthermore, set-based visualization methods [42], such as region-based overlay techniques, can be used to enhance the relationship between the glyphs and their corresponding regions. However, they are not utilized in our current solution. We will try to implement them in the future.

For the interaction, users can simply select a cluster for data analysis or a sequence of clusters for comparison. When a sequence of clusters is selected, the selecting path is directly shown on the plot, as indicated by **a2** in Figure 5(a).

5.2.2   Other Views

Three other views are provided to support smooth data exploration for multivariate data.

Table Lens. A table lens (Figure 5(b)) is provided to assist users in comparing selected clusters in a sequence, and analyzing the relationship among variables. In our implementation, the

height and width of each bar represent the mean value of each variable in a cluster and the relative number of points in the cluster, respectively. Each row of the table lens represents the value change of a variable through the selected sequence of clusters. Users can obtain a clear view of the value change of each variable and conclude data relationships through this table lens.

Parallel Coordinate. A parallel coordinate is used to provide a detailed view of multivariate data. After users select clusters on the scatterplot, detailed visualization for the cluster data will be shown, as illustrated in Figure 7(a). The color band is also supported on the parallel coordinate to maintain the consistency with the scatterplot by visualizing the mean and the SD of the cluster. The center of the band on each axis represents the mean value of each variable, whereas the half width of the band indicates the SD. Line-based visualization can assist accurate comparison for clusters, whereas band-based visualization can provide an overall impression for the differences among clusters. The axis order of the variables is determined by maximizing the correlation between all neighboring variables, as described in [43]. The same variable order is applied to the glyphs and the table lens.

5.2.3   Interactions and View Coordination

View coordination is supported in the system among all the aforementioned views. Users can explore the scatterplot to achieve an appropriate LOD abstraction for the data through zooming and panning interactions. At the topmost level, the system provides the data mean and SD of the entire dataset. At the lowest level, users can view each of the data point. Clusters of interest can be selected for analysis. Then, the detailed data of the clusters will be visualized in the parallel coordinate. Users can also select a sequence of clusters and compare them on the table lens.

Two types of explorations can be used. First, view-dependent exploration is supported. Users can zoom and pan to view different regions of data on the scatterplot. Then the clusters and the glyphs will be updated immediately after users stop the mouse interactions. Second, the hierarchical data structure can be generated automatically through preprocessing. Accordingly, users can smoothly explore data through zooming and panning interactions.

6   EVALUATIONS

In this section, we discuss the effect of the label cost and expected number of clusters on data abstraction. Quantitative and qualitative evaluations are used to validate the effectiveness of multilabel clustering in data abstraction. Moreover, we provide a set of glyph-based data abstraction examples using small and large scale data. A user study is also conducted to test the usability of the glyph-based abstraction for multivariate data exploration. The datasets used in our experiments are listed in Table 1. The Wine, Auto-MPG (without the car name), Wdbc (without the ID), and Shuttle datasets are from the UCI database [44], and are all numerical multivariate data. The MetObs and Agent data [45] are real application data. These datasets will be explained in the case studies. As for the implementation, the system is developed with C++/Qt and runs on a standalone machine. B/S framework and distributed implementation will be attempted in the future work.

6.1   Results on Different Parameters

Two important parameters, namely, the label cost ratio α and the expected number of clusters N_g, can directly affect the abstraction result. In this experiment, we project the Wine dataset onto
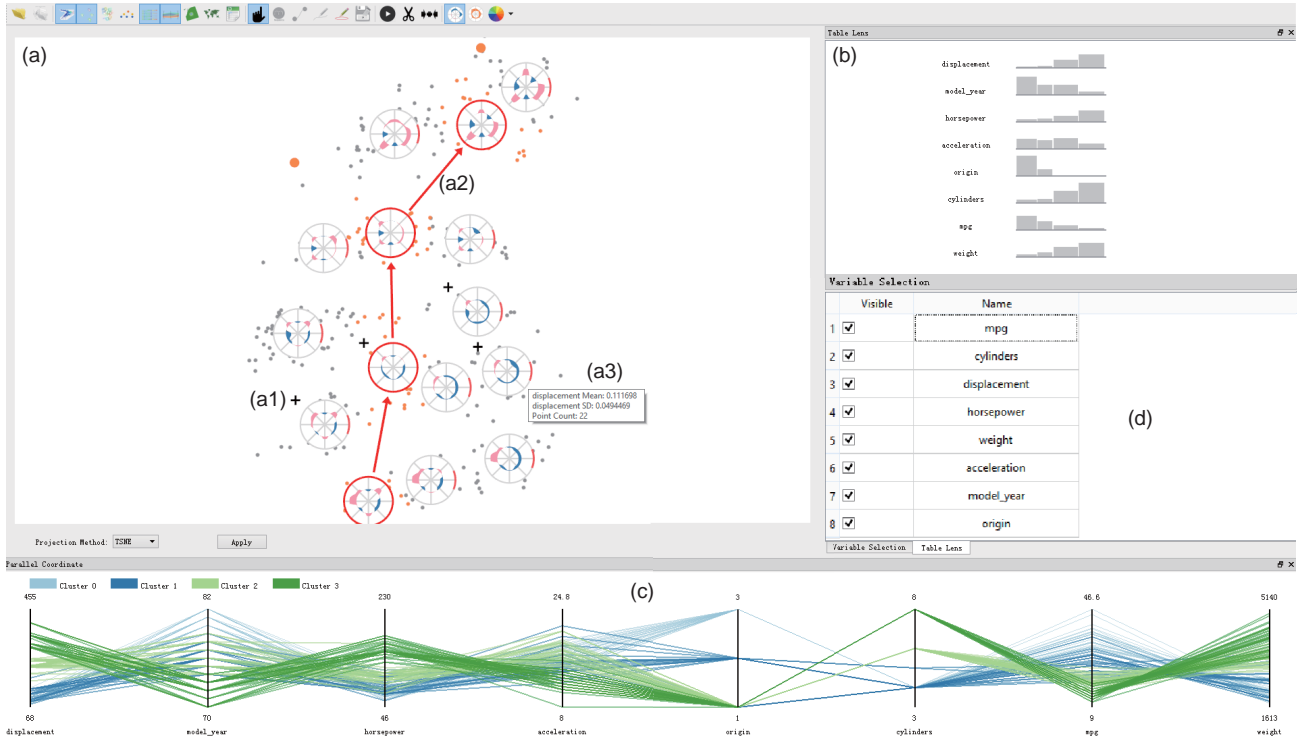
Fig. 5. System interface overview. (a) Glyph-based scatterplot. (b) Table lens for multivariate data analysis and comparison for a sequence of clusters. (c) Parallel coordinate for detailed data visualization and cluster comparison. (d) Property panel for variables.
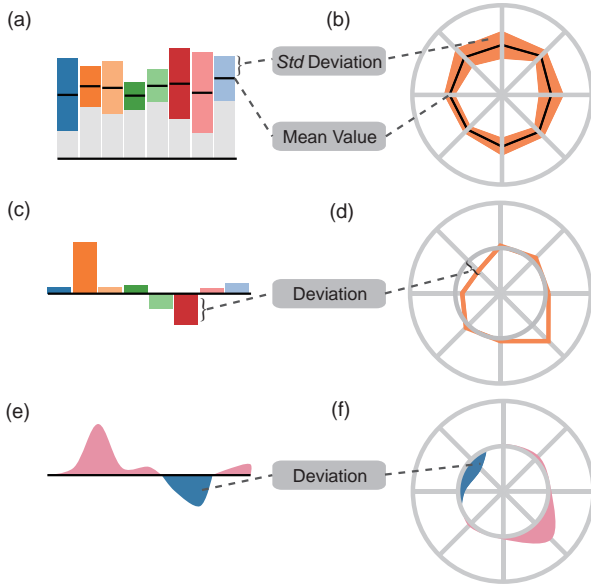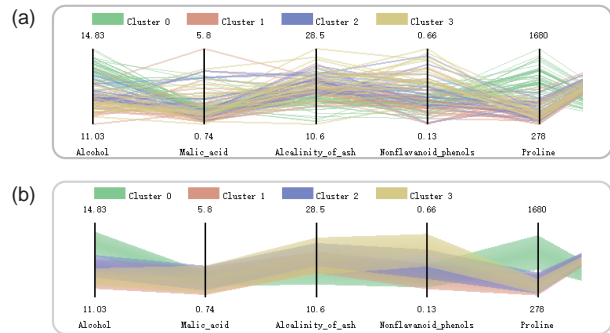

Fig. 6. Glyph alternatives

Fig. 7. Parallel coordinate design. (a) Visualization and comparison of detailed cluster data. (b) Band-based visualization for intuitive comparison among clusters.

TABLE 1
Datasets Used in Experiments

Dataset	Type	Record	Variable	Projection
Wine	Numerical	178	14	MDS
Auto-MPG	Numerical	290	8	MDS/t-SNE
Wdbc	Numerical	569	31	MDS
Shuttle	Numerical	14500	10	t-SNE
MetObs	Geospatial	2776	7	Lon/Lat
Agent	Geospatial	51274	6	Lon/Lat

the scatterplot using MDS. The projected result is illustrated in Figure 9(a).

Figure 9 provides the second-level abstraction results based on different α values when N_g is set to 10. The number of clusters decreases as α increases. When α is set to 0, many small clusters appear as indicated by the arrows in Figure 9(b). After we gradually increase the α, small clusters disappear and large ones are preferred. In real applications, α can be set depending on the application requirements of either obtaining an overview of

the data through large clusters or detecting possible data outliers through small clusters.

Figure 10 provides the second-level abstraction results based on different N_g values when α is set to 1.0. The number of clusters increases as N_g increases. However, the number does not increase linearly with N_g because the cost functions help adaptively control
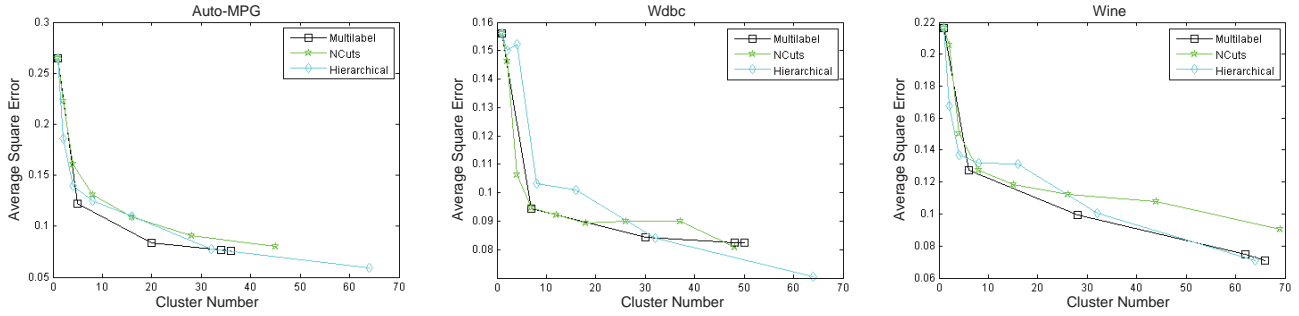
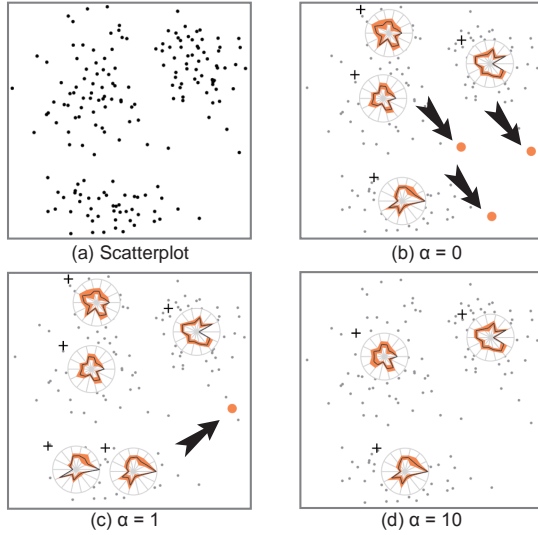Fig. 8. Abstraction quality measure for the Auto-MPG, Wdbc and Wine datasets.

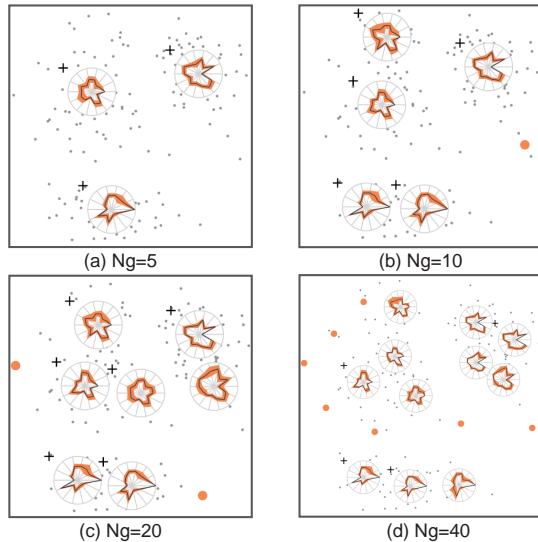Fig. 9. Abstraction results based on different α values.

Fig. 10. Abstraction results based on different N_g values.

the number of clusters for the abstraction. N_g can be set in accordance with the complexity of the data (e.g., number of variables) and the screen resolutions in real applications. If the number of variables is small and the glyph is sufficiently simple, then a high value for N_g can be selected. In all the following

experiments, α is set to 1.0, which works effectively in most of our experiments. N_g is set to 10 except for the agent simulation data wherein N_g is 40.

6.2 Data Abstraction Quality

We adopt the nearest neighbor measure as the data abstraction quality metric to evaluate the usability of the method numerically [21]. The metric uses square error to measure the data abstraction quality in multiresolution visualization. This objective is in line with the general goal of data clustering, that is, to minimize within-cluster variation.

We use the Wine, Auto-MPG and Wdbc datasets for the test. In our experiments, we first linearly normalize each of the variable in the dataset into the range of $[0, 1]$ and then project the data onto 2D scatterplots using MDS. For comparison, we use NCuts and hierarchical clustering with centroid linkage in the system. As for comparison with other classical methods, such as k-means and expectation maximization, Delong et al. [2] provided detailed discussion on the differences among them. For example, they stated that k-means minimizes a special case of the cost as shown in Equation 6, but the multilabel optimization can automatically remove unnecessary models from the initial set of label proposals.

In our experiments, we first generate the hierarchical data structure using the top-down clustering and the NCuts method through preprocessing with a threshold of 0.1 for data SD. Then, the structure is traversed to achieve the different levels of abstraction for the data, and the results are plotted. We also set the expected number of clusters (a number sequence of $[1, 2, 4, ...]$, with an increasing rate of 2) that remains in the hierarchical clustering, and the corresponding average square error is calculated. The results of the Auto-MPG, Wine, and Wdbc datasets are shown in Figure 8. In this figure, the x-axis is the number of the abstracted clusters whereas the y-axis is the average square error. The plots clearly show that the average square error via the multilabel optimization is frequently smaller than or similar to those obtained via NCuts. The multilabel optimization also usually performs better than the hierarchical clustering when the number of clusters is small. However, the hierarchical clustering usually provides better results when the cluster number increases. One main reason for this result is that the high-level clusters in the top-down clustering have defined boundaries for the low-level clusters. As a result, low-level clusters of low abstraction quality appear near the boundaries, which can be solved by assigning high values for both label cost ratio α and expected number of clusters N_g for the clustering. Then, the maximum level of the hierarchy will decrease and few boundaries will be introduced in the top-down
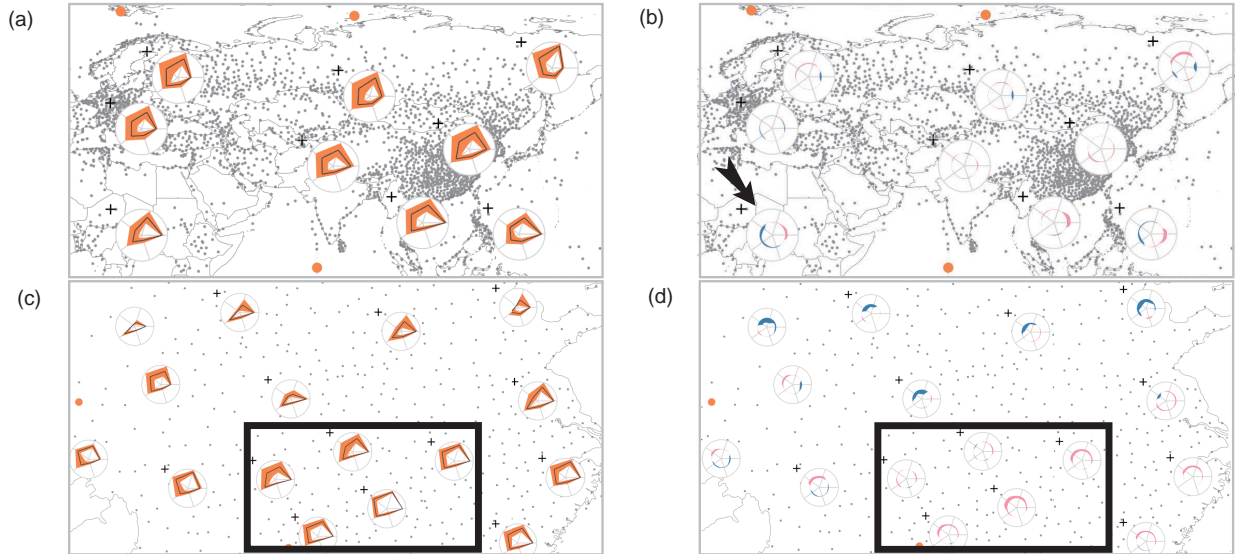
Fig. 11. Meteorological Observation Data Case: (a) Data overview. (b) Z-Glyph-based overview. (c) Detailed view of the data in a small region of China. (d)Z-Glyph-based detailed view.

clustering. However, this condition will result in a large number of clusters in high level abstractions. In real applications, we suggest using the view-dependent clustering in exploring multivariate data to avoid predefined boundaries and ensure the abstraction quality.
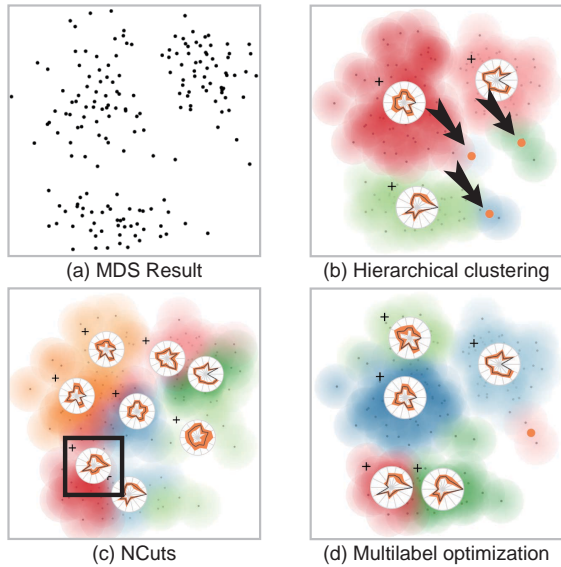

Fig. 12. Cluster comparison for the Wine dataset. (a) Initial projection. (b) Hierarchical clustering result with six clusters. (c) NCuts result with eight clusters. (d) Multilabel optimization result with six clusters.

We also investigate the clustering results by visually comparing them. For example, Figure 12 shows the results for the Wine dataset. We use different colors as the background to indicate which cluster each data point belongs. Figure 12(d) shows the second-level results of six clusters from the multilabel optimization. Correspondingly, we generate the same six-cluster result via hierarchical clustering. For NCuts, we select the fourth-level results of eight clusters instead, because NCuts only generates the binary clustering results. The results indicate that hierarchical clustering generates many small clusters that only contain one

or two points, as indicated by the black arrows in Figure 12(b). This problem was also discussed by Chen et al. [46] when they constructed an overview for a dendrogram, which required additional effort to handle it. This issue is also the reason why hierarchical clustering usually performs worse than the multilabel optimization when the clustering number is small. For NCuts, some clusters that are not visually proper may exist, such as the cluster indicated by the black rectangle in Figure 12(c), because NCuts segments data in a binary manner. The comparison results show that the multilabel optimization method can be effectively applied to generate abstractions for a multivariate scatterplot.

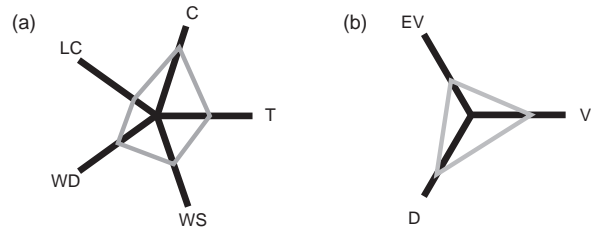## 6.3 Meteorological Observation Data

Fig. 13. Indicators for the glyph. (a) Indicator for the meteorological data case. (b) Indicator for the agent simulation data case.

In this subsection, we provide an example of geospatial multivariate point data. The data include meteorological observations on April 16, 2013 collected from weather stations distributed in Asia, Europe, and Africa. The data also include numerous variables, from which we select the longitude, latitude, cloud amount (C), low-level cloud amount (LC), wind direction (WD), wind speed (WS), and temperature (T) by excluding variables that exhibit considerable data loss.

We first project the data onto a map based on the longitude and latitude, as shown in Figure 11. Each point on the map represents a data record from a weather station, and we obtain a total of 2776 data records in the dataset. Each axis represents a variable in the data, as illustrated in Figure 13(a). We can obtain a clear overview
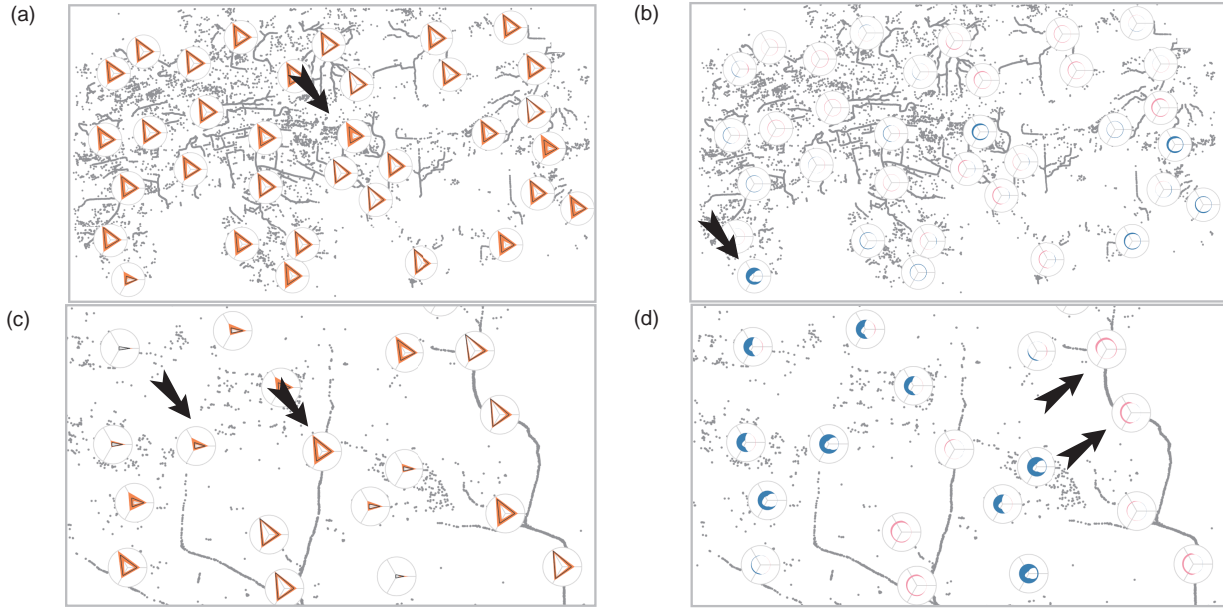
Fig. 14. Agent Simulation Data Case: (a) Data overview. (b) Z-Glyph-based data overview. (c) Closer view for a area with high SDs. (d) Comparison based on Z-Glyph.

of the multivariate data through the radar glyph, as shown in Figure 11(a). For example, T increases from north to south, as indicated by the three arrows in Figure 11(a). All three clusters show high standard deviations in LC, whereas the most northern cluster presents a lower SD in C compared with that of the two others. This comparison can be conducted among different data clusters which are distributed on the entire map. Moreover, we can easily detect a cluster with mean values that differ much from others, as indicated by the black arrow in Figure 11(b). The cluster possesses lower LC and WS than the other clusters. The small differences among clusters can also be easily discovered through the glyphs. Thereafter, we can zoom in to explore much detailed data. For example, Figure 11(c) shows the data in middle China. The data in the southern areas differ from those in the northern areas. The southern areas present higher C and LC values than do the northern areas. Meanwhile, we can easily detect the small differences among clusters through the glyphs, as indicated in the black rectangle in Figure 11(d). However, carefully comparing the glyphs in Figure 11(c) requires much effort.

6.4 Agent Simulation Data

The agent simulation data describe the action of agents during a evacuation in an urban area. They are time series and multivariate data that describe the status of an agent during the evacuation. In this experiment, we only use the data from one of the time steps. The data contain 51274 agents distributed on a map and five attributes, namely x position, y position, velocities (V), effective velocity ratio (EV) and distance (D). The x and y positions indicate where an agent is located on the map. We set N_g to 40 in this example because the number of variables is small.

Similar to the meteorological data case, we also first project the data on the basis of their positions, as shown in Figure 14. Each point on the map represents an agent data. Given that the number of the points is large, the clutter problem occurs. Thus, we can barely obtain a clear view of all data points on the map. However, Figure 14(a) can effectively provide an overview for

the multivariate data distribution. The relationship between the axes and the variables is indicated in Figure 13(b). We can easily locate a cluster of high SDs, as indicated by the black arrow in Figure 14(a). Another cluster with mean values that differ from those of others can also be efficiently detected, as indicated by the black arrow in Figure 14(b). Thereafter, we can zoom into a smaller area. Figure 14(c) provides a detailed view of the area indicated by the arrow in Figure 14(a). Then, we can find that the high SDs in this area are attributed to two types of agents distributed in different environments, as indicated by the arrows in Figure 14(c). One type of agents is distributed on the main street, and these agents have high V and EV; the other type is distributed in the residential area. The evacuation speed is low owing to obstacles, such as the houses. Z-Glyph can be used to detect the small differences among clusters. For example, the differences between the two clusters indicated by the arrows in Figure 14(d) can be compared easily. However, this task is difficult in Figure 14(c). Although these two clusters of agents both lay on the main street, they still have differences in V and D. Through such a view-dependent LOD exploration, users can easily obtain a clear understanding of the data.

6.5 SOI Examples

In this subsection, we provide examples to show the usage of the glyph-based abstraction in analyzing SOIs on scatterplots.

6.5.1 The UCI Auto-MPG Dataset

In this experiment, we use the UCI Auto-MPG dataset. The dataset contains 290 records and 8 numerical variables after we remove the variable of car name. The t-SNE is used to project the dataset onto the scatterplot. An interesting shape appears in the projection, as shown in Figure 15(a). This shape represents some underlying data relationships in the dataset. However, no intuitive methods are available to analyze such an SOI apart from selecting small sets of points iteratively and comparing them. With the glyph-based abstraction and the table lens, we can easily select the sequence
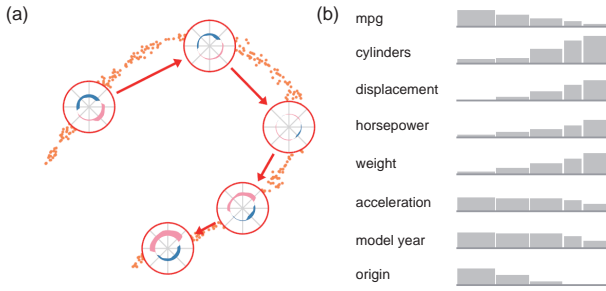
Fig. 15. The UCI Auto-MPG Data Case: (a) The glyph-based abstraction for the whole dataset and selected sequence of clusters. (b) The corresponding table lens for the selected clusters.

of clusters, as highlighted in Figure 15(a). The corresponding variable values of the clusters are visualized in the table lens, as shown in Figure 15(b). From the table lens, we can easily conclude many possible variable relationships. For example, the mpg may present a negative correlation with the weight. Additional detailed analysis can be conducted to confirm this conclusion.

This case shows that the scatterplot and the glyph-based abstraction can provide an intuitive guide for users to explore the multivariate data. The SOIs on the scatterplot and the view coordination between the scatterplot and the table lens can effectively assist users in detecting possible data relationships.

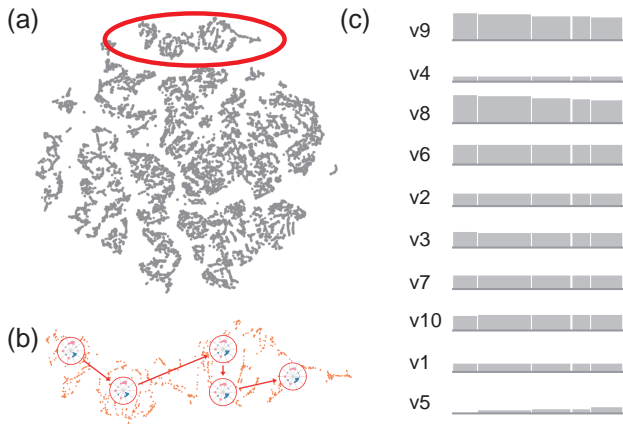### 6.5.2 The UCI Shuttle Dataset

Fig. 16. The UCI Shuttle Dataset Case: (a) An overview of the Shuttle dataset. (b) A glyph-based abstraction for a subset of points in the dataset and a sequence of selected clusters. (c) The corresponding table lens for the selected clusters.

The UCI shuttle dataset contains 14500 records and 10 variables. The scatterplot generated with t-SNE is shown in Figure 16(a). Many small clusters of points can be detected from the projection. Then, we can look into a small cluster, which is indicated by the red circle in Figure 16(a). A glyph-based abstraction for this cluster is provided in Figure 16(b). We can select the sequence of clusters and analyze the data relationship represented by the SOI of the cluster. The corresponding table lens is provided in Figure 16(c). From the table lens, we conclude that v8 may exhibit a positive correlation with v9 and a negative correlation with v5. Similar analysis can be drawn for other small clusters. The scatterplot provides an overview of the dataset; thus,

a subset of the dataset with the glyph-based abstraction can be analyzed easily.

6.6 User Study and Feedback

6.6.1 User Study for Glyphs

A user study is carried out to compare the usability of the glyph alternatives (Figure 6) in supporting multivariate data exploration and analysis. The advantages and disadvantages of the bar chart-based and the radar chart-based glyphs in visualizing multivariate data have been extensively discussed in many previous studies. Cao et al. [41] validated the advantage of Z-Glyph compared with the commonly used radar glyph in detecting data outliers. Therefore, we do not repeatedly compare them in this user study. Moreover, we find that the glyphs in Figure 6(a) and (b) can act similarly in helping users obtain an overview for the data through a pilot user evaluation. Both can provide an intuitive visualization for the mean and SD values of a cluster of data. Thus, we mainly discuss the usability of the glyphs in detecting clusters of interest during the data exploration.

Prior to the user study, we first generated 16 sets of data with eight variables, by adding small perturbations to specified mean and SD values. Thereafter, we randomly selected three sets and added great perturbations to the mean values of three randomly selected variables. Again, we randomly selected three other sets and added great perturbations to the SD values of three randomly selected variables. Then, the data were used for the subsequent user study.

A website was developed for conveying the visualizations and supporting users in completing related tasks in the user study. The 16 sets of data were randomly placed in a web page, as shown in Figure 4. We provided an independent web page for each of the glyphs in Figure 6 and for each task. Users were expected to select three glyphs that best satisfy the task requirements during the user study. They were asked to complete two tasks depending on the two types of clusters which users may be interested in.

T1   Select three glyphs with higher SDs than those of the others.

T2   Select three glyphs with mean values that differ from those of the others.

The glyphs in Figures 6(c)-(f) did not convey the SD information; thus, they were not employed for **T1**. After the users completed the selection, we asked them to rate the usability of the glyphs for the corresponding tasks. The rating is an interval scale between 1 and 5 with an equal interval of 1. 1 stands for the worst usability, which means they can barely judge based on the glyph; 5 stands for the best usability, which means they can complete the selection very intuitively. Selection accuracies and user feedback are also recorded after the user study.

Fifteen users participate in the user study. They were all master or PhD students majoring in computer science and currently performing research on visualization. They were all trained to learn about all the glyphs and asked to conduct a test user study with another test dataset before the formal user study. The detailed statistics of the user study, which are mean and SD, are provided in Table 2. We also perform dependent t-test for the results from **T1** and repeated measures analysis of variance (RM-ANOVA) for those from **T2**. There is a statistically significant effect of glyphs on rating in **T1** ($t(14) = -2.779, p \approx 0.015$). Glyph(b) achieves an improvement of 0.533 ± 0.743 in rating compared with Glyph(a).

TABLE 2
User Study Ratings and Accuracies

	Glyph	T1	Accuracy	T2	Accuracy
(a)		4.26±0.70	95.6±11.7%	3.73±0.70	93.3±18.7%
(b)		**4.80±0.41**	100±0%	**2.93±1.22**	75.6±40.8%
(c)		*	*	**4.87±0.35**	100±0%
(d)		*	*	4.53±0.64	97.8±8.61%
(e)		*	*	4.73±0.59	100±0%
(f)		*	*	**4.87±0.35**	100±0%

TABLE 3
Time Costs of The Cases (ms)

		Super Pixel Number		
Dataset	Raw Data	500	1000	2000
Wine	52	*	*	*
Auto-MPG	152	*	*	*
Wdbc	1042	*	*	*
Shuttle	*	17	56	170
MetObs	*	57	202	579
Agent	*	68	256	897

However, the effect of glyphs on accuracy in **T1** is not statistically significant ($t(14) = -1.468, p \approx 0.164$). Overall, Glyph(b) performs better than Glyph(a) in **T1**. Moreover, there is a statistically significant effect of glyphs on rating ($F(5, 70) = 20.29, p < 0.001$) and accuracy ($F(5, 70) = 4.609, p \approx 0.001$) in **T2**. Post hoc tests using the Bonferroni correction reveals that Glyphs(a) and (b) are statistically significantly different from the other glyphs on rating ($p < 0.05$) in **T2**. Their ratings are lower than those of the other glyphs. Although their accuracies are not statistically significantly different from those of the other glyphs. A few users (2 for Glyph(a) and 5 for Glyph(b)) performed incorrect selections in **T2** with their assistance. Furthermore, Glyphs(c)-(f) all perform well in **T2** with high ratings and accuracies, whereas one user performed an incorrect selection using Glyph(d). The differences in their ratings and accuracies are not statistically significant ($p > 0.05$).

After the formal user study, we also interviewed the participants about their selections and collected their feedback on the glyphs. Most of them pointed out that the color bands in Glyph(a) and (b), which were used to encode the SDs, became obstacles in comparing the mean values. This restriction may be the main reason for the low rating, low accuracy and high SD when users completed **T2** with Glyph(b). Some of them pointed out that Glyph(c) outperformed Glyph(e) in **T2**. The colors used in Glyph(c) can help them clearly distinguish different variables. This task is difficult with Glyph(e). On the contrary, Glyph(d) and (f) can handle this problem appropriately. They used the angles and axes to encode the different variables. Moreover, Glyph(f) was more intuitive than Glyph(d) because the colors were used as additional visual cues for comparison.

The statistics and the user feedback indicate that Glyph(b) can effectively support **T1**, and Glyph(c) and (f) can effectively support **T2**. To maintain the consistency of the glyphs in the system, Glyph(b) and (f) are selected because they are both radar chart-based glyphs.

6.6.2   User Evaluation for System

Eight participants were also asked to use our system and provide feedback on its usability. They were all trained to use our system with the Wine dataset until they were familiar with the system. Thereafter, they were asked to complete three tasks.

T1   Describe the entire dataset in brief.
T2   Find a cluster with mean values that differ from those of others.
T3   Find a cluster with high SDs and further explore that cluster of data.

We collected their comments on the system when they were performing these tasks and discussed about the system with them after they completed the tasks.

All the participants believed that the glyphs were helpful for the LOD exploration of large scale data, as long as the clutter problem existed. However, at the very beginning of the user study, some of them were confused after they zoomed into detailed data areas and new glyphs appeared. This problem disappeared only after they familiarize themselves with the system. Some of them suggested that percentile data should be used instead of the mean and SD. Additional flexible control of the glyphs and their encodings may be implemented to help improve the usability of the system. Some of them also suggested that the system should allow users to manually define some clusters and automatically cluster other data. This issue is out of the scope of the current study but will be tackled in the future.

7   DISCUSSIONS

In this section, we discuss several related problems of the method.

7.1   Performance and Scalability

A problem with the glyph-based abstraction is the performance and scalability of the algorithm and system. The time cost for the entire system contains three main parts, namely the projection, the super pixel construction and the multilabel optimization. The time cost for the projection relies on the projection method. Discussions on the time cost can be found in related corresponding papers. The time cost for the super pixel construction comprises the time for the image-based mapping and the SLIC super pixel generation. The complexity for both processing procedures is $O(N)$, where N is the number of data points and image pixels for the image-based mapping and super pixel generation, respectively. This can be completed in a very short time. Although the solution for the multilabel optimization is heuristic and time consuming when the number of node in the graph is large, the super pixel-based processing significantly decreases the time cost. Detailed time costs of the multilabel optimization for the cases in the study are also recorded, as listed in Table 3. These time costs are the optimization time for the entire dataset displayed in a single view. The experiments are carried out on a desktop computer with an Intel Core i7-3770 CPU and 32G memory. In all the experiments, if the number of points is greater than 2000, then the super pixel-based method is utilized; otherwise, the scatter points are directly used in the optimization. From the table, we can find that the optimization time is generally less than a second. With a proper number of expected super pixels (e.g., 500 in our experiments), the system can support interactive data exploration for even large scale data.

7.2 Cluster-Based Data Analysis

The clusters are used as the basis for the analysis of SOIs in our system. Similar attempts in data mining research area have been conducted previously. For example, Tung et al. [47] attempted to find and visualize non-linear correlation clusters. They first clustered the data and then visualized them to assist users in analyzing data relationships. Alternatively, we first conduct the projection and then cluster the data. The scatterplot works as a guideline for users to explore the data in an LOD manner. The use of this approach is more intuitive for the users, than directly providing the clustering results. There is a difference between the clustering-based visualization and our projection-based clustering. We will try to provide detailed research and discussions on this topic in our future work.

7.3 Curse of Dimensionality

The curse of dimensionality is a common problem for multivariate data analysis based on data projections. Various methods have been proposed to assist users in effectively analyzing high dimensional data in a low-dimensional space, such as dimension reduction, linear or non-linear data projections and user driven data projections [16]. Researchers have also attempted to use several projections together to provide a full illustration for the high dimensional data [15]. Instead of directly solving the curse of dimensionality, our method attempts to enhance the data analysis on the low-dimensional representations of the high-dimensional data. Our method can be smoothly combined with existing scatterplot-based high-dimensional data analysis methods and helps users explore and analyze the data intuitively.

7.4 View-Dependent and Top-Down Clustering

Two types of clustering are supported in our system. They should be selected according to the requirements in real applications. The view-dependent clustering should be used when computing resources are sufficiently powerful to support nearly real-time processing of the data. This type of clustering can usually provide better abstractions of high quality than does the top-down clustering, as discussed in Section 6.2. Meanwhile, the top-down clustering can be used when preprocessing is required to ensure the smooth interactions during the data exploration.

8 CONCLUSION AND FUTURE WORK

In this study, we introduce a cluster-based visual abstraction for multivariate scatterplots. We can assist users in efficiently obtaining an overview of multivariate data distributions on a scatterplot and analyzing SOIs. A pair of glyphs is used in the system to guide users in exploring the scatterplot interactively, and the coordinated views are provided to support multivariate data analysis. The comparison of the multilabel optimization with other popular methods shows that this optimization method can ensure a good data abstraction quality for the scatterplot. The usage of the system is also demonstrated through case studies on a series of datasets, including UCI multivariate datasets, geospatial datasets and a volume dataset. Finally, we discuss related problems with our system and its potential adaptations for other applications.

An existing problem with the LOD glyph-based visualization is the appearance of popping artifacts during the zooming and panning. This problem affects the effectiveness of our visualization design to some degree. We will study this problem systematically and eliminate the artifacts in the future. Given that LOD abstractions for multivariate data are provided, various conclusions can be drawn from different levels of visualizations. In our future work, we will analyze the differences among these conclusions and attempt to provide co-analysis methods using different levels of abstractions.

REFERENCES

[1] L. der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[2] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *International journal of computer vision*, vol. 96, no. 1, pp. 1–27, 2012.

[3] S. Liu, D. Maljovec, B. Wang, P. T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," in *State of The Art Report*. Eurographics Conference on Visualization (EuroVis), 2015.

[4] D. A. Keim, M. C. Hao, U. Dayal, H. Janetzko, and P. Bak, "Generalized scatter plots," *Information Visualization*, vol. 9, no. 4, pp. 301–311, 2010.

[5] H. Janetzko, M. C. Hao, S. Mittelstadt, U. Dayal, and D. Keim, "Enhancing scatter plots using ellipsoid pixel placement and shading," in *System Sciences, International Conference on*. IEEE, 2013, pp. 1522–1531.

[6] M. C. Hao, U. Dayal, R. K. Sharma, D. A. Keim, and H. Janetzko, "Visual analytics of large multidimensional data using variable binned scatter plots," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2010, p. 753006.

[7] A. Mayorga and M. Gleicher, "Splatterplots: Overcoming overdraw in scatter plots," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 9, pp. 1526–1538, 2013.

[8] A. Inselberg and B. Dimsdale, "Parallel coordinates," in *Human-Machine Interactive Systems*. Springer, 1991, pp. 199–233.

[9] E. Kandogan, "Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions," in *Proceedings of the IEEE Information Visualization Symposium*, vol. 650. Citeseer, 2000, p. 22.

[10] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley, "Dna visual and analytic data mining," in *Proceedings of the conference on Visualization*. IEEE Computer Society, 1997, pp. 437–441.

[11] R. Rao and S. K. Card, "The table lens: merging graphical and symbolic representations in an interactive focus+context visualization for tabular information," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1994, pp. 318–322.

[12] J. E. Nam and K. Mueller, "TripAdvisorˆ{N-D}: A Tourism-Inspired High-Dimensional Space Exploration Framework with Overview and Detail," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 2, pp. 291–305, 2013.

[13] S. Cheng and K. Mueller, "The Data Context Map: Fusing Data and Attributes into a Unified Display," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 22, no. 1, pp. 121–130, 2016.

[14] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu, "Scattering points in parallel coordinates," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1001–1008, 2009.

[15] D. Lehmann and H. Theisel, "Optimal Sets of Projections of High-Dimensional Data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 22, no. 1, pp. 609–618, 2016.

[16] H. Kim, J. Choo, H. Park, and A. Endert, "InterAxis: Steering Scatterplot Axes via Observation-Level Interaction," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 22, no. 1, pp. 131–140, 2016.

[17] S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci, "Visual exploration of high-dimensional data through subspace analysis and dynamic projections," *Computer Graphics Forum*, vol. 34, no. 3, pp. 271–280, 2015.

[18] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-based visualization: Foundations, design guidelines, techniques and applications," *Eurographics State of the Art Reports*, pp. 39–63, 2013.

[19] T. Ropinski, S. Oeltze, and B. Preim, "Survey of glyph-based visualization techniques for spatial multivariate medical data," *Computers and Graphics*, vol. 35, no. 2, pp. 392–401, 2011.

[20] Y. Albo, J. Lanir, P. Bak, and S. Rafaeli, "Off the radar: Comparative evaluation of radial visualization solutions for composite indicators," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 22, no. 1, pp. 569–578, 2016.

[21] Q. Cui, M. O. Ward, E. A. Rundensteiner, and J. Yang, "Measuring data abstraction quality in multiresolution visualizations," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 709–716, 2006.

[22] E. Bertini, A. Tatu, and D. Keim, "Quality metrics in high-dimensional data visualization: An overview and systematization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2203–2212, 2011.

[23] E. Bertini and G. Santucci, "By chance is not enough: preserving relative density through nonuniform sampling," in *Proceedings of the eighth international conference on Information Visualisation*, July 2004, pp. 622–629.

[24] G. Ellis and A. Dix, "Enabling automatic clutter reduction in parallel coordinate plots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 717–723, 2006.

[25] H. Chen, W. Chen, H. Mei, Z. Liu, K. Zhou, W. Chen, W. Gu, and K.-L. Ma, "Visual Abstraction and Exploration of Multi-class Scatterplots," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 12, pp. 1683–1692, 2014.

[26] M. Kreuseler, N. Lopez, and H. Schumann, "A scalable framework for information visualization," in *Information Visualization. IEEE Symposium on*. IEEE, 2000, pp. 27–36.

[27] B. Heckel, G. Weber, B. Hamann, and K. I. Joy, "Construction of vector field hierarchies," in *Proceedings of the conference on Visualization*. IEEE Computer Society, 1999, pp. 19–25.

[28] A. Telea and J. J. Van Wijk, "Simplified representation of vector fields," in *Proceedings of the conference on Visualization*. IEEE Computer Society, 1999, pp. 35–42.

[29] Q. Du and X. Wang, "Centroidal Voronoi tessellation based algorithms for vector fields visualization and segmentation," in *Proceedings of the conference on Visualization*. IEEE Computer Society, 2004, pp. 43–50.

[30] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner, "Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 6, no. 2, pp. 150–159, 2000.

[31] Z. Peng and E. Grundy, "Mesh-driven vector field clustering and visualization: An image-based approach," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 2, pp. 283–298, 2012.

[32] M. Zinsmaier, U. Brandes, O. Deussen, and H. Strobelt, "Interactive level-of-detail rendering of large graphs," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 12, pp. 2486–2495, 2012.

[33] M. Balzer and O. Deussen, "Level-of-detail visualization of clustered graph layouts," in *Visualization. International Asia-Pacific Symposium on*. IEEE, 2007, pp. 133–140.

[34] J. Abello, F. Van Ham, and N. Krishnan, "Ask-graphview: A large scale graph visualization system," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 669–676, 2006.

[35] D. Battista, P. Eades, I. G. Tollis, and R. Tamassia, *Graph drawing: algorithms for the visualization of graphs*. Pearson, 1998.

[36] Y. Zhou, O. Grygorash, and T. F. Hain, "Clustering with minimum spanning trees," *International Journal on Artificial Intelligence Tools*, vol. 20, no. 01, pp. 139–177, 2011.

[37] I. Herman, G. Melancon, and M. Marshall, "Graph visualization and navigation in information visualization: A survey," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 6, no. 1, pp. 24–43, 2000.

[38] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.

[39] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, "SLIC Superpixels," EPFL, Tech. Rep., 2010.

[40] C. Y. Ip, A. Varshney, and J. JaJa, "Hierarchical Exploration of Volumes Using Multilevel Segmentation of the Intensity-Gradient Histograms," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 12, pp. 2355–2363, 2012.

[41] N. Cao, Y.-R. Lin, D. Gotz, and F. Du, "Z-glyph: Visualizing outliers in multivariate data," *Information Visualization*, vol. 0, no. 0, pp. 0–0, 0. [Online]. Available: http://dx.doi.org/10.1177/1473871616686635

[42] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, "Visualizing sets and set-typed data: State-of-the-art and future challenges," in *Eurographics conference on Visualization (EuroVis)– State of The Art Reports*, 2014, pp. 1–21.

[43] M. Ankerst, S. Berchtold, and D. A. Keim, "Similarity clustering of dimensions for an enhanced visualization of multidimensional data," in *Information Visualization. IEEE Symposium on*. IEEE, 1998, pp. 52–60.

[44] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[45] M. Wijerathne, L. Melgar, M. Hori, T. Ichimura, and S. Tanaka, "Hpc enhanced large urban area evacuation simulations with vision based autonomously navigating multi agents," *Procedia Computer Science*, vol. 18, pp. 1515–1524, 2013.

[46] J. Chen, A. M. MacEachren, and D. J. Peuquet, "Constructing overview+ detail dendrogram-matrix views," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 889–896, 2009.

[47] A. K. H. Tung, X. Xu, and B. C. Ooi, "Curler: finding and visualizing nonlinear correlation clusters," in *Proceedings of the international conference on Management of data*. ACM, 2005, pp. 467–478.
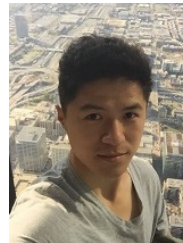
Hongsen Liao received the B.S. degree in Computer Software from Tsinghua University, Beijing, China. He is currently pursuing the Ph.D. degree from the School of Software, Tsinghua University, Beijing, China. His research interests include scientific visualization and visual analytics.

Yingcai Wu is an assistant professor at the State Key Lab of CAD & CG, Zhejiang University, Hangzhou, China. He received his Ph.D. degree in Computer Science from the Hong Kong University of Science and Technology (HKUST). Prior to his current position, Yingcai Wu was a researcher at the Internet Graphics Group in Microsoft Research Asia, Beijing, China. His primary research interests lie in visual behavior analytics, visual analytics of social media, visual text analytics, uncertainty-aware visual analytics, and information visualization. For more information, please visit http://www.ycwu.org.

Li Chen received the Ph.D. degree in visualization from Zhejiang University, China, in 1996. She is currently an Associate Professor with the Institute of Computer Graphics and Computer Aided Design, School of Software, Tsinghua University, China. Her research interests include data visualization, image processing, and parallel algorithm.

Wei Chen is a professor at the State Key Lab of CAD & CG, Zhejiang University. Professor Chen received his PhD from the Zhejiang University in 2002. His research interests include visualization, visual analytics, and biomedical image computing. Professor Chen presently serves on the steering committee of IEEE PacificVis. For more information, please refer to http://www.cad.zju.edu.cn/home/chenwei.