Visual Abstraction and Exploration of Multi-class Scatterplots

Haidong Chen, *Student Member, IEEE*, Wei Chen, *Member, IEEE*, Honghui Mei, Zhiqi Liu, Kun Zhou, Weifeng Chen, *Student Member, IEEE*, Wentao Gu, and Kwan-Liu Ma, *Fellow, IEEE*

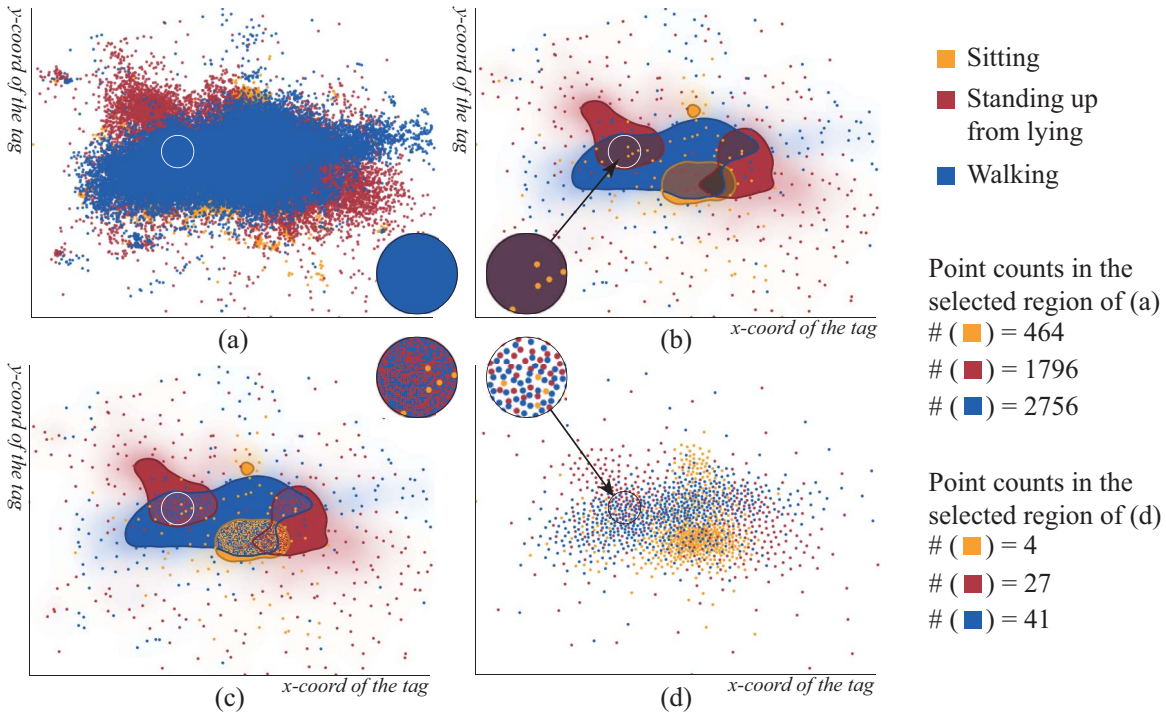


Fig. 1. Visualization for the Person Activity dataset [1]. (a) The conventional scatterplot suffers from severe overdraw. (b) The Splatterplots [33] may synthesize new colors. Data points in sparse regions are explicitly highlighted. (c) The Splatterplots with additional noise [22] can enhance the distinguishability in dense regions. (d) Our result: please notice the differences in overlapped regions indicated by the circles. The number of data points before and after abstraction in this region are listed on the right side. The relative density orders among classes are preserved with our method.

Abstract—Scatterplots are widely used to visualize scatter dataset for exploring outliers, clusters, local trends, and correlations. Depicting multi-class scattered points within a single scatterplot view, however, may suffer from heavy overdraw, making it inefficient for data analysis. This paper presents a new visual abstraction scheme that employs a hierarchical multi-class sampling technique to show a feature-preserving simplification. To enhance the density contrast, the colors of multiple classes are optimized by taking the multi-class point distributions into account. We design a visual exploration system that supports visual inspection and quantitative analysis from different perspectives. We have applied our system to several challenging datasets, and the results demonstrate the efficiency of our approach.

Index Terms—Scatterplot, overdraw reduction, sampling, visual abstraction

- *Haidong Chen, Wei Chen, Kun Zhou, Honghui Mei, and Zhiqi Liu are with State Key Lab of CAD&CG, Zhejiang University. E-mail: {chenhaidong, chenwei, zhoukun, meihonghui, liuzhiqi}@cad.zju.edu.cn.*
- *Wei Chen is the corresponding author. He is also with the Cyber Innovation Joint Research Center, Zhejiang University.*
- *Weifeng Chen is with the Zhejiang University of Finance & Economics. E-mail: cwf818@gmail.com.*
- *Wentao Gu is with the Zhejiang GongShang University. Email: zjgsu-guwentao@hotmail.com.*
- *Kwan-Liu Ma is with the University of California at Davis. Email: ma@cs.ucdavis.edu.*

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014. Date of publication 11 Aug. 2014; date of current version 9 Nov. 2014.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.
Digital Object Identifier 10.1109/TVCG.2014.2346594

1 Introduction

As one of the most fundamental visual representations, scatterplots use 2D Cartesian coordinates to depict a set of bivariate points. The point collection in a 2D plane can be explored and analyzed to study point distributions [15, 42], clusters [11, 47], outliers [33], local trends [8, 9], and axial correlations [25].

One challenging problem for visualizing scatterplots is the overdraw (see Figure 1 (a)) caused by a dense point distribution. In the past decade, many research efforts have concentrated on reduction of overdraw and its effects. In general, one can modulate the limited visual channels of points, e.g., the point size [30, 31] or the opacity [18] to decrease the overplotting degree. Alternatively, the density estimation [2, 3, 18] can be employed to reformulate the point distribution in a simple form. This is effective for clarifying dense regions, but can hardly show outliers which typically lie in low density regions. Spatially moving the overplotted points to unoccupied pixels [25, 26] is another way to address the overdraw problem. However, it is not suit-

able for quantitative analysis because point distribution is changed.

The point overdraw problem is further exaggerated when multi-class points are shown in a single scatterplot [33]. To alleviate this problem, the continuous density fields with respect to each class of points can be reconstructed. Then the commonly used multi-variate data visualization methods such as contouring, color blending [33], and color weaving [20, 22] are utilized. Nevertheless, these methods may yield misleading representations, especially in regions where multiple classes exist. In addition, the relative density orders among classes cannot be preserved well for quantitative analysis (see Figure 1 (b) and Figure 1 (c)). This greatly hinders the abilities of users for identifying outliers, discriminating dense regions, analyzing correlations, and comparing patterns.

Our work is motivated by an artistic map design that illustrates a territorial multi-class statistical dataset. Conventionally, a filled color map is employed to visualize such kinds of datasets, but it will lead to information loss. Take the Chicago ethnic distribution map as an example (see the bottom left in Figure 2), the relative population density and transitions among races are missing in each region (e.g., the one indicated by the black arrow). In contrast, Bill Rankin [35] proposed a dot map technique, with which the racial distributions in a region are directly represented by the density of multiple sets of colored dots. The dot color represents the race and the density encodes the size of the population. Generally, this representation is endowed with several merits: 1) no new colors are synthesized; 2) relative features are preserved locally. Essentially, the dot map technique employs a point sampling scheme to approximate the distributions of multiple variates. Compared with the uniformly filled color map, it greatly improves the readability of the visualization result in terms of showing multi-variate distributions and correlations.

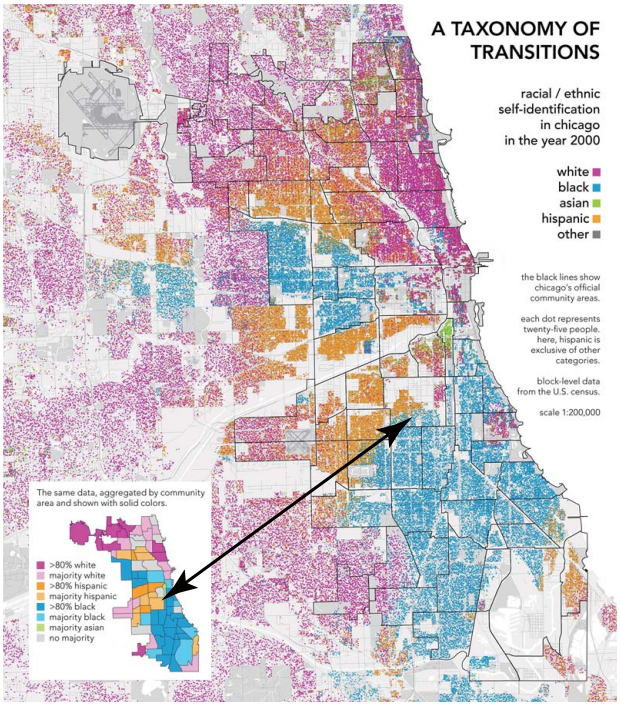


Fig. 2. The ethnicity map [35] of Chicago 2010 with the dot mapping representation motivates our work. For a high resolution version, please refer to the project website (http://www.radicalcartography.net).

Inspired by this dot map design, this paper proposes a feature preserving reformulation to reduce the visual clutter of multi-class scatterplots. The kernel is a reconstruction-and-resampling process that generates a visual abstraction of the input point distribution. To make it amenable for multi-class points, a multi-class blue noise based sampling scheme [41] is employed. We also propose a color optimization technique to enhance the perceptual contrast in multi-class regions. The integrated visual exploration system consists of a suite of distribution study, and focus+context interaction toolkits to support visual inspection and quantitative analysis of multi-class points.

In summary, this paper presents a complementary means to conventional scatterplot visualization techniques with the following contributions:

- A hierarchical multi-class sampling scheme that reduces the overdraw and preserves the point distributions for quantitative analysis.
- A visual exploration system for interactive inspection and analysis of multi-class points.

The remaining parts are organized as follows. Section 2 summarizes related work. Our approach is described in Section 3. A visual exploration system is presented in Section 4. The evaluation of our approach and some directions for future work are discussed in Section 5 and Section 6. Finally, we conclude this paper in Section 7.

2 Related Work

Visualizing a set of points with different attributes in a 2D plane is a fundamental task. Various schemes have been proposed to enhance the visual perception.

2.1 Overdraw Reduction for Scatterplots

One critical issue for displaying a large amount of points in a 2D plane is the overdraw problem. Conventional solutions can be roughly categorized into three classes.

Changing the visual channels are a simple and intuitive way to deal with the overdraw issue. Woodruff et al. [44] chose to visualize scatter dataset with small-size dots in highly dense regions and large-size icons in sparse regions for outliers, as small dots occupy relatively less screen space. A similar scheme was exploited by Chen et al. [11] to explore the resulting projection for DTI fibers. The Utopian [12] system generated an alpha blended scatterplot for interactive topic modeling. Dang et al. [16] proposed a method to handle overplotting issue by stacking points in the third dimension. Luboschik et al. [32] introduced the aligned weaving technique that can improve the representation of overlapping clusters in scatterplots. Generally these methods are effective, but the number of usable visual channels are quite limited. Our method is compatible with these techniques and can further alleviate the overdraw issue.

Density estimation provides an alternative solution to avoid overdraw for visualizing large scatter datasets. The density can be simply measured by dividing the drawing space into bins and counting the number of data points falling into them. Alpha blended scatterplots are an example of this technique. This simple discrete form of density is prone to introduce bias in visualization. Carr et al. [7] used hexagonal cells to accumulate densities. Bachthaler et al. [2, 3] proposed a rigorous, accurate, and generic mathematical model to create the *continuous scatterplots*. Zinsmaier et al. [48] employed the *kernel density estimation* (KDE) to visualize overlapped nodes and edges in large graphs. The density estimation can also be adopted to visualize overlapped trajectories [36, 43]. Color mapping and contouring are two common methods to visualize and highlight dense regions in a reconstructed continuous density field. Unfortunately, these methods neglect the outliers in the resulting visualization, as outliers create small regions of low density in the continuous density field. Feng et al. [18] suggested mean emphasis in kernel density estimation to reveal real outliers. Mayorga and Gleicher [33] proposed to explicitly display outliers coupled with color mapping and contouring. Nevertheless, simultaneously encoding and visualizing multiple density fields are quite difficult for multi-class scatterplots. Multi-variate data visualization methods such as color blending and color weaving can be employed to visualize multiple density fields with limitations as well. The color blending methods may synthesize new colors. The color weaving methods cannot show relative density features among classes. Our method exploits the density information as constraints to resample the input data points. It does not bring new colors and also

enables users to study relative density features through the number of non-overlapped samples.

Spatial distortion can also be used to eliminate the overdraw problem. Keim and Herrmann [26] introduced the *Gridfit* algorithm to avoid overplotting when visualizing large amounts of spatial reference data points. Its key idea is to place the overplotted data points on the nearest unoccupied pixels and shift data points along a screen-filling curve. Later, Keim et al. [25] developed a more flexible technique called the generalized scatterplot, which allows users to strike a balance between overplotting and distortion. Janetzko et al. [24] proposed to enhance the scatterplots using the ellipsoid pixel placement scheme. Further, Wu et al. [45] presented a warping method to avoid important features overplotted during resizing. In essence, distortion based methods change the underlying topology and distribution of the dataset, which may cause misleading perceptions.

2.2 Interactive Exploration and Analysis for Scatterplots

A bunch of navigation tools and scatterplot variants were developed to assist users discovering and exploring the insights from datasets. Buering et al. [5] proposed two interaction techniques to explore a large dataset in a small screen: a geometric-semantic zoom that provides a smooth transition between overview and detail, and a fisheye distortion that displays the focus and context regions of the scatterplot in a single view. Elmqvist et al. [17] exploited animated transitions between scatterplots for visual exploration of a multidimensional dataset. Yuan et al. [47] presented a comprehensive tool equipped with selection, zooming, dragging, and linking for interactive subspace exploration of 2D scatterplots. Collins et al. [13] enhanced the set relation of points on the plot with clustering and contours. Chan et al. [8, 9] introduced the flow-based scatterplots for sensitivity and local trends analysis where a local regression analysis is leveraged. Leland et al. [15, 42] defined a set of measures to characterize the scatterplot point distributions and used these measures to organize scatterplots for high-dimensional data analysis. Radloff et al. [34] used additional visual encodings to represent dots in scatterplots on heterogenous displays. In this work, we provide a complementary means to improve the readability of multi-class scatterplots. We also show how existing analysis methods and interactions such as brushing and focus+context can further enhance users' understanding in meaningful ways.

2.3 Noise for Visualization

Noise has been widely used in various visualization techniques. A well-known application is noise-based flow visualization [23, 37]. Spot noise [40] was introduced to vector field visualization by Jark van Wijk. Inspired by this method, a white noise texture was employed by *line integral convolution* (LIC) based methods [37]. Khlebnikov et al. [28] investigated the possibility of using Gabor noise to visualize 2D multivariate dataset. Later they extended this method to multivariate volume data visualization [27]. A perceptually adapted Perlin Noise was employed by Coninx et al. [14] to visualize uncertain scalar field. Bertini and Santucci [4] presented a random data sampling method to model the underlying data density for scatterplot quality enhancement. Unfortunately it is intractable for multi-class scatter datasets. Our work leverages the multi-class blue noise sampling method [41] to abstract the overlapped multi-class data points while preserving features such as relative density orders among classes. One of its main advantages is that the blue noise feature is guaranteed for each individual class and the union of all classes.

3 Visual Abstraction of Multi-class Scatterplots

Overdraw is caused by the conflict between the limited screen space and the large number of points. In some cases, it is inevitable, and prevents users from insightful data observations. To address this issue, we employ a visual abstraction scheme that estimates the point density of each class and resamples the estimated density fields. The main concept behind our approach is to use a cloud of sampled points to mimic the intrinsic multi-class point distributions. Rather than using a conventional point sampling (e.g., blue noise) for each class, a multi-class blue noise sampling scheme is employed for multiple point classes. Please refer to [41] for the differences between the single-class and multi-class blue noise sampling techniques.

Showing points generated with the multi-class blue noise is straightforward. To clearly differentiate data classes, we optimize the color set by maximizing the distinguishability in multi-class regions. Additionally, different types of point shape such as ellipse and dot-line are employed to show the local trends.

3.1 Point Density Estimation

Density estimation is a common way to handle large datasets. It creates a continuous density scalar field from the given data points. KDE is a well-studied statistical tool for this purpose.

Let $X_i = \{x_i^1, x_i^2, ..., x_i^m\}$ be the data points of the i-th data class. Mathematically, the density at location x is computed by:

$$\hat{f}_i(x) = \sum_{j=1}^{m} K_h(x - x_i^j), \tag{1}$$

where $K_h(\cdot)$ is a kernel function with bandwidth h which determines the smoothing degree of the reconstructed density field. Different from the conventional KDE method, we do not use the number of data points to normalize the density for the purpose of directly comparing densities at location x. A Gaussian kernel is used in our approach with the bandwidth h determined by the Silverman's rule of thumb [38] .

3.2 Feature-preserving Point Resampling

Point sampling is a fundamental tool to reduce the number of data points while best preserving the features for many computational tasks. Due to the uniformity and absence of spectral bias [10], the blue noise sampling technique has been favored in many applications such as image stippling [41, 46] and point cloud resampling [10].

Individually sampling each data class cannot guarantee that the blue noise features are preserved for the union of all data classes. Therefore, we employ the adaptive multi-class blue noise sampling method [41] that uses a dart throwing technique to reproduce the input data distributions. During the sample generation process, an $n \times n$ symmetric matrix $\mathbf{R}^x$ at the trial location x is built for conflict check. n is the number of data classes. $\mathbf{R}^x$ generalizes the distance constraint in the Poisson disk sampling for a single class. The element $\mathbf{R}^x_{i,j}$ specifies the distance constraint between the i-th and the j-th data class at the location x (see Figure 3 as an illustration). We use the estimated density to compute the distance constraint matrix $\mathbf{R}^x$. Specifically, the diagonal elements of $\mathbf{R}^x_{i,i}$ are defined as $\omega/\hat{f}_i(x)$, and all non-diagonal elements are computed according to the suggestion by Wei [41]. ω is a user adjustable parameter. $1/\omega$ can be treated as the frequency for sampling. In our system, users can change this parameter to control the number of points shown in the view.

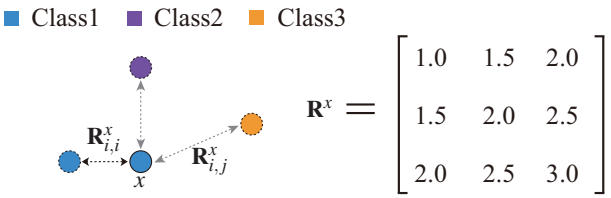


Fig. 3. To position a sample of Class1 (i.e., the solid blue dot) at location x, the distance constraint matrix $\mathbf{R}^x$ is computed first. Its diagonal elements $\mathbf{R}^x_{i,i}$ constrain the intra-class distances, and its non-diagonal elements $\mathbf{R}^x_{i,j}$ restrict the inter-class distances.

By default, ω is computed by means of a heuristic rule in our exploration system. For 2D scatterplots, the orthogonal projection scheme is employed to render data points. Let φ be the zooming scale for rendering points, and r be the point radius. The sampling frequency parameter is approximated by:

$$\omega = \frac{r}{\varphi}\bar{f} \tag{2}$$

where $\bar{f}$ represents the mean density of all data classes over the domain.

A simple way for resampling is to sample the reconstructed continuous density field generated by the KDE technique. However, new data points may be produced (see the green points in the region indicated by the orange ellipse in Figure 4 (b)), which is not preferable.

In our approach, the resampling process is performed in a discrete sampling space constituted by all input multi-class data points. To ensure that each class is well sampled, a new trial sample is always randomly selected from the class that is currently most under-filled (Please refer to [41] for the details on the fill rate computation). If the trial sample passes the conflict check, it will be inserted to the output data point list. Otherwise it will be discarded. Figure 4 compares an input scatterplot, the one by sampling the estimated point density field, and the one sampled in the discrete space.

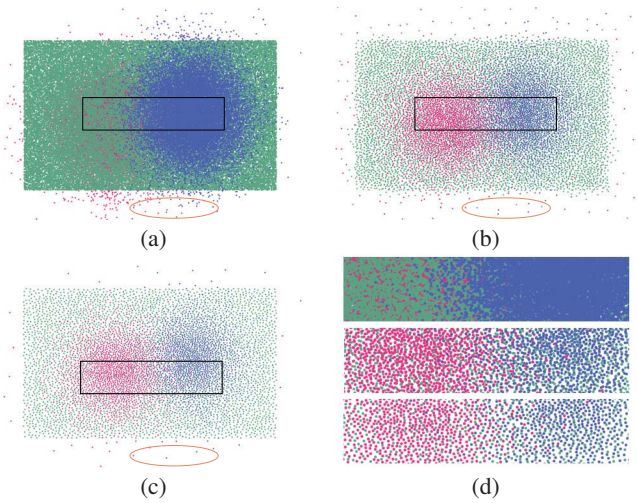


Fig. 4. Applying two sampling schemes to a synthetic multi-class dataset (two data classes follow a Gaussian distribution and the other one follows a uniform distribution). (a) The input scatterplot; (b) Sampling in the continuous density field. (c) Sampling in the discrete space. (d) From top to bottom: the marked regions in (a-c). Notice the differences of points in regions indicated by the orange ellipses.

3.3 Consistency-preserving Hierarchical Sampling

Zooming operations are a commonly used interaction to study a dataset. If using the current zooming scale φ to resample the input data points, it might lose consistency (notice the differences in regions indicated by the orange circles at the top row in Figure 5). Ideally, more points should be added to the view when the view is zoomed in and some points should be removed when the view is zoomed out.

To achieve a smooth zooming, we employ a hierarchical sampling scheme that pre-computes a sequence of coarse-to-fine sampling points. Specifically, we start sampling with an initial small φ which corresponds to the coarsest level. Empirically, φ is set to make all data points are shown in a small region of the entire viewport (in practice, we set it to be 1/8). For each subsequent level, the pre-computed samples are used as partial samples for next runs of sampling. Finally, the output samples record the points that are approximately sorted by zooming levels in an ascending order. Algorithm 1 presents the pseudocode.

The run-time visualization is performed by showing all points with the generated zooming levels that are smaller than the current level. Smooth transitions are supported between different zooming levels.

3.4 Abstractive Visualization

Resampling multi-class points generates a sequence of non-overlapped points. Therefore, visual clutter caused by a high spatial density can be greatly eliminated. Our approach employs two additional techniques to improve the perceptual quality of the abstracted set of points.

Algorithm 1 Hierarchical Sampling

Input: P: the multi-class data points; φ: the initial zooming level; $\hat{f}_i(x)$: density fields; P'': a temporary array
Output: P': the output samples; S: an array that records the number of samples generated at each zooming level

```
while P ≠ ∅ and φ < φ_max do
    m ← 0
    while P ≠ ∅ do
        // Select a trial sample from the most unfilled data class
        x ← SelectTrial(P)
        R^x ← BuildRMatrix(x, φ, f̂_i(x))
        pass ← ConflictCheck(R^x, P') // Do conflict check
        if pass then
            Push x back to P' and remove it from P
            m ← m + 1
        else
            Remove x from P to P''
        end if
    end while
    Push m back to S
    Push P'' back to P
    Clear P''
    φ = 2φ
end while
```

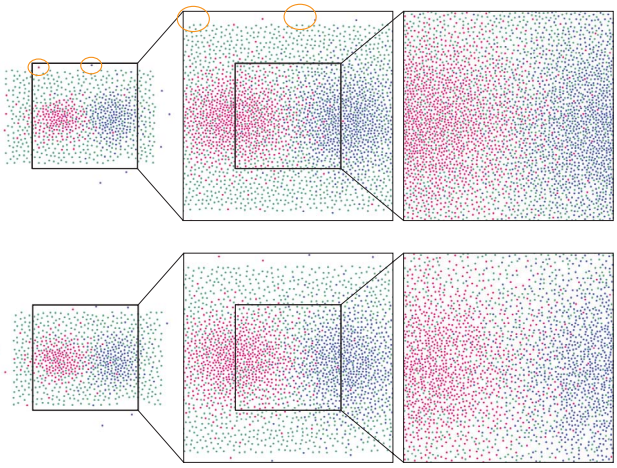


Fig. 5. Top: without our hierarchical sampling scheme, some points maybe discarded when the view is zoomed in; Bottom: our scheme preserves the points of low levels when the view is zoomed in.

3.4.1 Point Color

A recent work [19] suggests that color outperforms shape for multi-class scatterplots in many comparative tasks. Even when the number of points increases, the performance remains good. However, using color to distinguish different classes is not a trivial task. In order to clearly identify different classes, the color distinguishability in overlapped regions needs to be as significant as possible.

We offer two methods for users to select colors, namely, a friendly hue wheel based color picking interface and an automatic color selection approach.

In the hue wheel based color picking interfaces, users are allowed to freely select a set of colors to label individual data classes.

To help users in selecting distinguishable colors for different classes, we also provide an automatic color selection approach, in which an optimal color set is selected automatically. More specifically, we employ a color optimization algorithm, which takes the density information into account.

Suppose that the scatterplot is rendered on a rectangular screen area, which is further divided into M local regions (e.g. 5×5 pixels for a

local region). Our automatic color selection approach seeks to find a color set $\mathbf{C} = \{C_1, C_2, ..., C_n\}$ in the CIELAB color space so that the visualization has a maximal color distinguishability. C_i denotes the color for the i-th data class. We maximize the following objective function to obtain an optimal color set:

$$E_{cost} = \sum_{m=1}^{M} \beta_m \sum_{i,j<n,i<j} \alpha_{m,i,j} |C_i - C_j|, \quad (3)$$

where $\alpha_{m,i,j}$ denotes the *inter-class weight*, which measures the weight of the color distinguishability between the i-th and j-th class in the m-th local region. Meanwhile, β_m denotes the *intra-class weight*, which measures the weight of the m-th local region in the entire screen area. The item $|C_i - C_j|$ denotes the color distinguishability between C_i and C_j. It is computed by the Euclidean distance in the CIELAB color space, which is perceptually uniform. In our implementation, the weights $\alpha_{i,j}$ and β for the m-th local region are defined as

$$\alpha_{i,j} = e^{-|\bar{f}_i - \bar{f}_j|}, \quad \beta = \sum_{i=0}^{n} \bar{f}_i. \quad (4)$$

To prevent the optimization from reaching an unsatisfying result, we add a further constraint that colors are at least apart from each other at distance d in a perceptually uniform color space. Following a soft constraint implementation, it yields a minimization problem:

$$\min -E_{cost} + k \sum_{i,j<n,i<j} E_{penalty}(C_i, C_j), \quad (5)$$

where k is an adjustable weight, and

$$E_{penalty}(C_i, C_j) = \max(0, 1 - \frac{|C_i - C_j|}{d}).$$

We choose the CIELAB color model, in which the L* channel denotes the perceptually perceived lightness.

Instead of optimizing within the entire color space, we intend to seek iso-lightness colors for different classes by leaving L* as a user adjustable parameter. Thereby, the optimization process minimize Equation (5) by search other two channels (i.e. a* and b*) in an iso-lightness plane within the gamut of the CIELAB color space. We apply the Nelder and Mead method [29] to find an optimal solution. As the downhill simplex method works iteratively and is prone to converge to a local minima, we run the optimization method several times with random initializations and keep the best solution finally.

3.4.2 Point Shapes

The sampled data points can be simply rendered as circular dots in the screen space. To aid local correlation exploration and analysis, other visual representations can be employed (see Figure 6):

- The **ellipse** representation is inspired by [46] which uses ellipse to represent the luminance gradient for pointillism painting.
- The **dot-line** scheme is similar to the one used in [8] for sensitivity analysis.

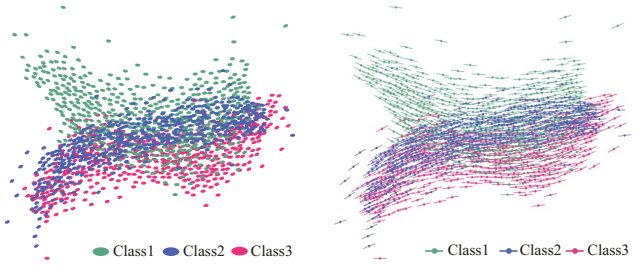
Specifically, the local trend $(u_{x_0}, v_{y_0}) = normalize(1, \frac{\partial y}{\partial x})$ at a 2D location (x_0, y_0) is approximated by a local linear regression analysis:

$$\frac{\partial y}{\partial x} \approx \frac{\sum_{(x_i,y_i)\in N_{(x_0,y_0)}} (y_i - y_0)(x_i - x_0)}{\sum_{(x_i,y_i)\in N_{(x_0,y_0)}} (x_i - x_0)^2}, \quad (6)$$

where $N_{(x_0,y_0)}$ represents the neighborhoods of (x_0, y_0) in a circular region. To obtain a visually pleasing result, we empirically restrict the ratio of the major radius and the minor radius for ellipse to 1.618.

4 Visual Exploration of Multi-Class Scatterplots

Our visual exploration system is equipped with a set of interaction tools. The efficiency of our approach is best shown with high-resolution images. Please refer to the supplementary materials for details.



(a) (b)

Fig. 6. Using (a) Ellipses and (b) Dot-Lines to encode local trend information for a synthetic dataset.

4.1 System Overview

All widgets and views are designed to be collapsible so that more screen space can be preserved for the main view. Figure 8 shows an overview of our system. The data class list view summarizes all data classes (Figure 8 (a)). Users can drag interested data classes into the main view (Figure 8 (b)) for conjoint exploration. A set of visualization methods and interaction tools are provided in Figure 8 (c). Configuration panels (Figure 8 (d)) are put on the left side.

4.2 Visual Exploration of A Single Class

To gain an overview of each class, we employ a data class list view (Figure 8 (a)) that follows a small multiple visual form [39]. Specifically, the visual representation of each data class is designed as a rectangular glyph (Figure 7) which contains two statistical histograms. Other information about the data class is also embedded.

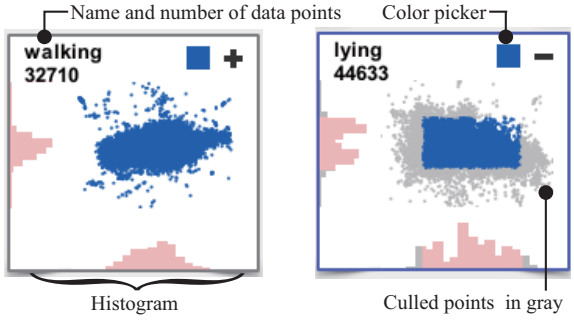


Fig. 7. The glyph representation shows an overview of a data class.

The data class list view provides an overview for exploration. Currently, the following interaction tools are supported in this view:

Clipping on the 1D histogram filters out data points that are not in given ranges. The culled data points are shown in gray as a context.

Dragging the interesting data class glyphs closer to facilitate comparison. Users can also drag a set of glyphs to the main view for conjoint exploration.

Sorting glyphs by the number of data points in each class.

4.3 Conjoint Multi-Class Exploration

The resulting visualization in the main view (see Figure 8 (b)) exhibits a global picture of the selected data classes. It is of equal, if not more, importance to interactively explore the data from different perspectives. Our system provides a set of tools inspired by many editing systems [6, 21].

4.3.1 Data Inspection

Users are allowed to use the following tools to study multi-class scatter dataset in a single view:

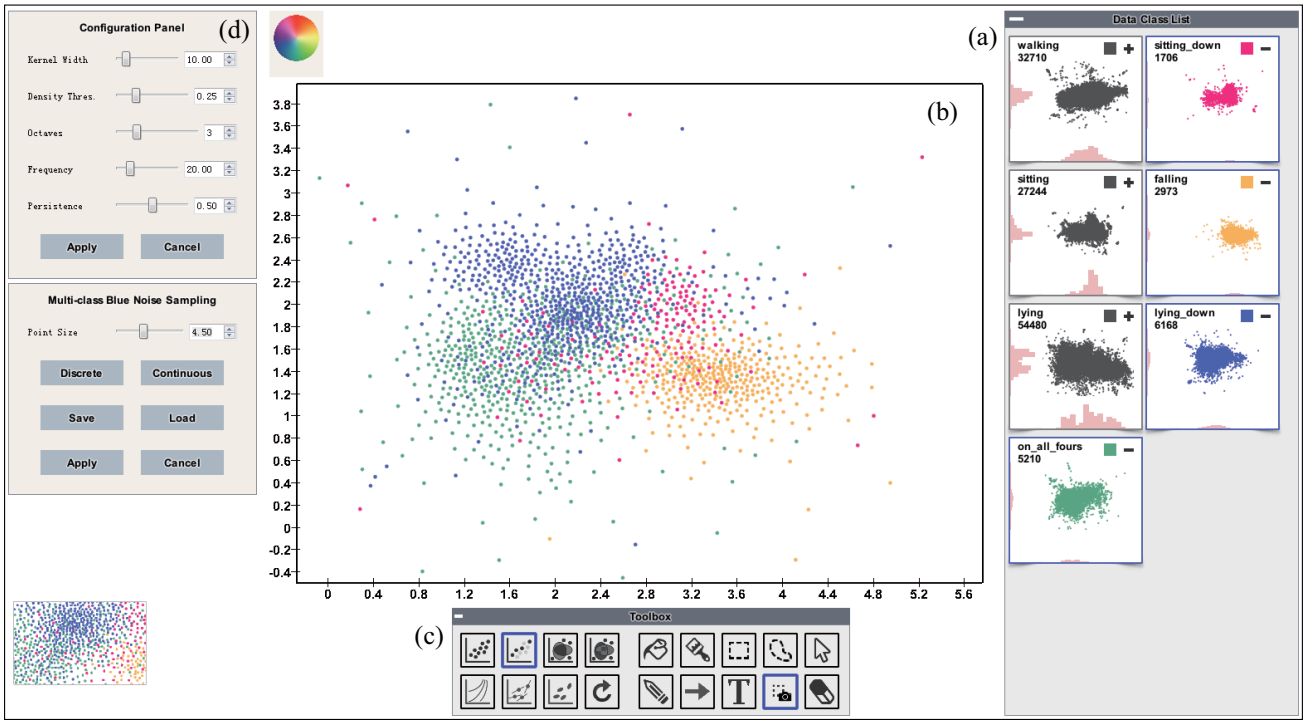


Fig. 8. The main interface of our exploration system.

Highlighting Inspired by the Gestalt common fate principle, the point set in a data class is synchronously moved upwards when the user holds the right mouse button, and restores when users release it. Repeatedly pressing and releasing the right mouse button helps users identify an individual class in the main view.

Selection Two types of selection tools are provided to allow for specifying a region of interest. The *box* selection tool allows for drawing a rectangular region. The *lasso* selection tool supports drawing an arbitrary shaped region.

Painting Users freely specify the visualization mode for a selected region. Currently our system supports: *alpha blending*, *color blending* [33], *color weaving* [22], *color compositing* with Perlin noise [20], and *contouring*.

Brushing Users can gradually brush on the main view with a specific visualization mode for exploration.

Annotations Users can identify and analyze the exploration results by free-style annotations. Iconic and textual annotations are supported.

Snapshot Users use the snapshot tool to record the visualization of a selected region, which is orderly shown in the main view. Double clicking the snapshots will unfold the recorded visualizations in the main view.

4.3.2 Density Exploration

Our approach is compatible with many multi-variate density field visualization methods, like contouring, color blending, and color weaving. Our system incorporates all these features and provides the flexibility of modulating all of them for insightful analysis.

The contouring is an effective method to quickly locate the dense point regions. It can be used to get a high-level overview of the density information. Further, color blending, color compositing with Perlin noise, and color weaving can be employed to identify the data classes in dense overlapped regions. Generally, color blending has limited capabilities for this task. To simultaneously identify data classes and relative density information, our noise sampling based method can be employed.

4.3.3 Local Trend Exploration

When studying at a specific region, users may want to study the local relationship between the two dimensions. With our system, users can use the painting tool with either ellipse shape or dot-line shape to gradually discover local trends. By treating the local trend field as a vector field, a set of streamlines can be generated to further enhance the perception [8]. Once users find an interesting pattern, the annotation tools can be used to record the findings.

5 Evaluation

To evaluate the feasibility and applicability of our approach, we applied our approach to two datasets. The first dataset records the NBA teams' shooting positions on the court. The second one is a real mobile user profile dataset which includes over 380,000 users and their calling records. A preliminary user study was conducted to testify the effectiveness of our approach.

5.1 Case Studies

5.1.1 The NBA Teams' Shooting Positions Dataset

This dataset records the shooting positions of several NBA teams (including the Miami Heat, the Golden State Warriors, the Memphis Grizzlies, and so on) in the 2012-2013 season. To study the strategies employed by different teams, we place them in the same view for comparative analysis.

The *conventional scatterplot* exhibits severe visual clutter (see Figure 9 (a)). Figure 9 (b) shows the result of splatterplot [33] where data points in sparse regions are randomly selected for highlighting. We can see a significant brown colored region near the basket, because it is easier to score when a player is close to the basket. The synthesized color might cause misunderstanding. The *color weaving* method can help users identify different teams in the overlapped regions (see Figure 9 (c)). Neither of these methods can answer the following two types of questions which require quantitative analysis:

- Which team shot more in a specific area, for example the region indicated by the green rectangle?
- Where the players preferred to shooting?

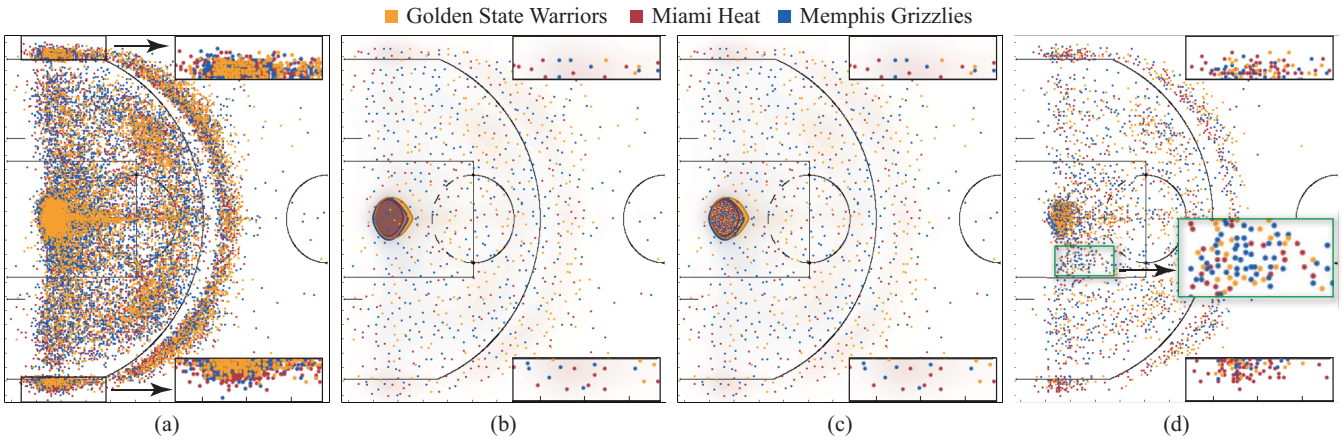


Fig. 9. Studying the shooting differences among the NBA teams with different scatter data visualization schemes. (a) The conventional scatterplot. (b) The Splatterplots [33]. (c) Using color weaving to enhance the perception of the data classes in dense regions. (d) Our result preserves the point distributions and the relative density orders among classes, and can be used for quantitative analysis.

Figure 9 (d) displays our result. It shows that **Memphis Grizzlies** shot more in the green rectangular area which was dominated by blue points. In addition, **Miami Heat** preferred three-points shooting in the left and right corners (see the highlighted regions shown on the right side of each result), because more red points are shown in these areas. However the Splatterplots cannot capture this property due to the random sampling mode employed in these sparse regions. The selection tool allows users to specify a region and inspect the exact number of shooting.

5.1.2 The Mobile User Profile Dataset

This dataset records the detailed usages and bills of 382,779 mobile users in a month. Each user has 17 attributes including the encrypted phone number, the package type, the total-call-charge, the total-talk-time, and so on. Two types of records are removed in our data cleaning process:

- Data records with missing attributes;
- Data records with illegal values, e.g. a record with a negative total-call-charge.

245,309 records are used after cleaning. After a quick look at the total-talk-time and the total-call-charge, we can find a strong linear relationship between them (see Figure 10 (a)). We use the *package type* to label each user. The derived scatterplot is a multi-class scatterplot.

We first employ the *Splatterplots* [33] by *color blending* and *contouring* the reconstructed density fields to study the density distributions. The lightness channel is employed to encode the density. The result in Figure 10 (b) clearly shows the dense regions as well as the points in sparse regions. In the overlapped dense regions, new colors are synthesized. The *color weaving* method can avoid this problem, and can help users identify classes in the overlapped dense regions (see Figure 10 (c)). The limitation of both methods is that the relative density orders among different classes are missing.

Figure 10 (d) presents the result of our method using the circular dot representation. The relative density orders in a local region can be perceived by the number of visible dots. For example, users with **Package Type 3** dominate the lower left part in this view. We can also find that users with **Package Type 2** and **Package Type 4** are the dominant classes in the highlighted rectangle (see Figure 10 (f)), because more cyan and orange points are displayed in this region. In addition, different types of correlations are clearly shown in our result with a dot-line representation (see Figure 10 (e)). The local trend of each point in **Package Type 3** and **5** roughly follows the 45 degree diagonal line (Please zoom in or refer to the supplementary images for details). This may indicate that **Package Type 3** and **5** do not contain a restriction on the minimum charge. In contrast, **Package Type 1**, **Package Type 2**, and **Package Type 4** do not exhibit such property, and the local trends are almost parallel to the dimension of total-talk-time. This may imply that each package type (1,2, and 4) has a varied restriction on the minimum charge.

5.2 User Study

Generally, four schemes that can be used to visualize multi-class scatter dataset were compared: the conventional scatterplots (C), the color blending based method (CB) [33], the color weaving based method (CW) [22], and our method (OURS). Note that a previous work [22] has concluded that color weaving outperforms color blending.

5.2.1 Study Design

Participants

In our user study, 26 participants were recruited from a university. Of these participants, 19 were male and 7 were female, 16 were graduate students and 10 were undergraduate students. Their ages ranged from 21 to 28 years old. All participants reported that they were not colorblind.

Apparatus

The user study was conducted on a normal PC equipped with a dell display (24-inch LCD with resolution of 1920×1080 pixels). The free online survey platform Kuiksurveys (http://kwiksurveys.com/) were employed. In general, 32 datasets were used in our user study (including both synthetic and real application datasets, 16 for the first task and 16 for the second task). Table 1 lists the details of these datasets. All visualization results were created with the same resolution of 800×600 pixels. The colors used to indicate data classes were selected by our optimization technique. All visualizations employed the circular dot to encode a data point.

Tasks

[T1] *Data classes identification*

In this task, all participants had to identify the number of data classes in a marked area, which was separately visualized with four schemes. 16 datasets were tested in this task. In order to avoid learning effects and the possible data bias, the datasets were randomly permuted and then were equally divided for each scheme test. In general, each participant had to answer 16 questions in this task. We provided a *Hard to determine* option in the listed answers. We gave a score of 1 for every correct answer and a score of 0 for every incorrect answer. A participant could achieve a maximum score of 4 for each scheme.

[T2] *Relative densities recognition*

In this task, each participant had to choose an answer that could best describe the relative density orders in the marked region. 16 different datasets were used in this task. Each participant had to finish 16 questions. Similar to **T1**, the datasets were first randomly permuted for a participant. The same scoring strategy was used for this task.

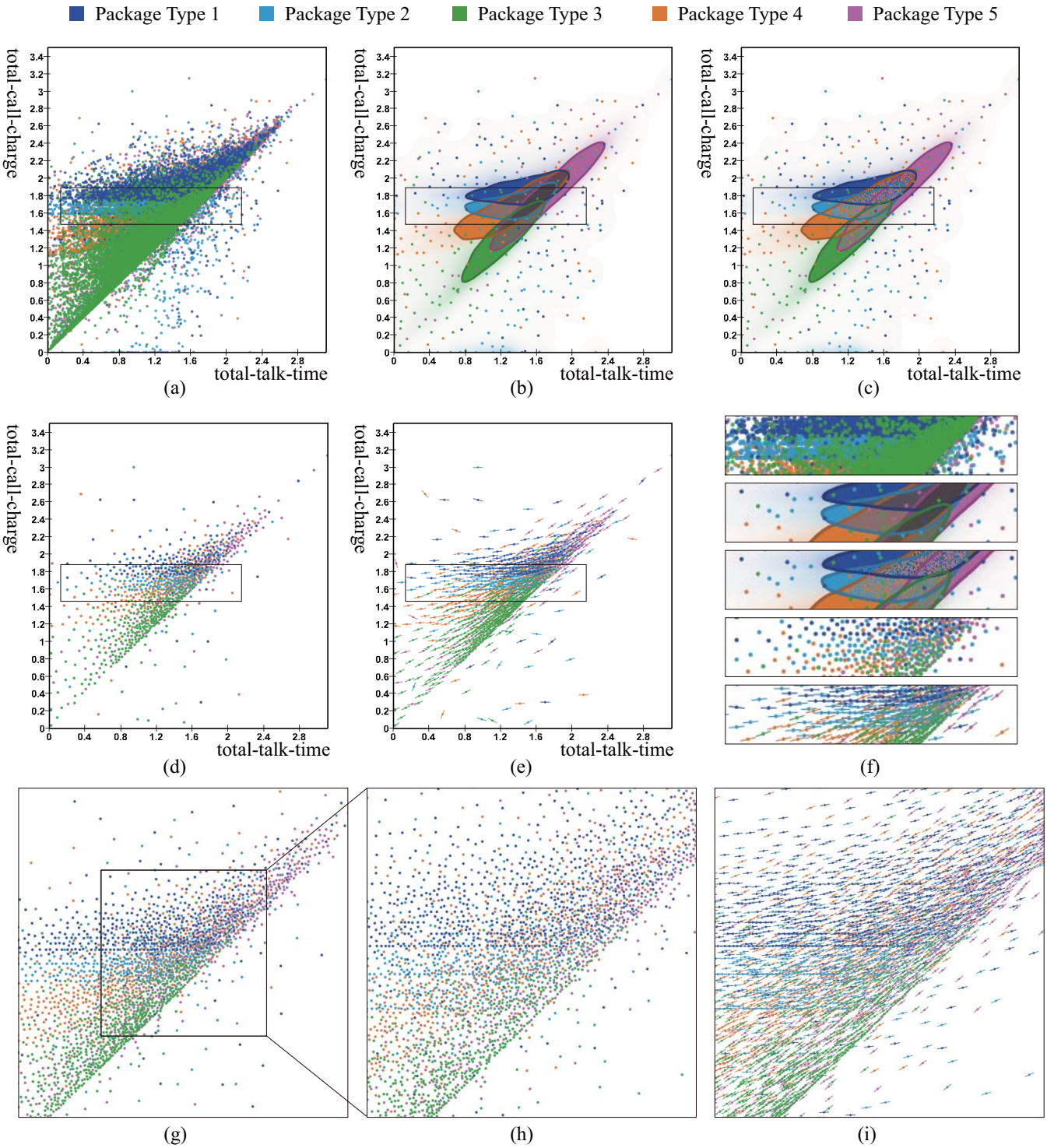


Fig. 10. Applying different scatter data visualization schemes to the mobile user profile dataset consisting of five classes. (a) The conventional scatterplot. (b) The Splatterplots [33]. (c) The splatterplot enhanced with color weaving. (d) Our approach with the circular dot representation. (e) Our approach with the dot-line representation for trends analysis. (f) The marked regions in (a-e). (g) and (h) are two consecutive zooming levels of (d). (i) The dot-line representation for (h).

Procedure

Before the formal study, we spent about 12 minutes to briefly train the participants, including getting their informed consent, completing a general questionnaire, and explaining tasks. After the two tasks were finished, a post study questionnaire were given to record their general comments about the techniques and the entire user study.

5.2.2 Result and Analysis

Figure 11 shows the overall performances on task **T1** and **T2**. It is easy to see that our method significantly outperforms other threes schemes especially in terms of relative density order recognition. By analyzing the incorrect answers for C, CB, and CW in the task **T2**, we find that the answer *Hard to determine* were selected 37 and 31 times for CB

Table 1. Profiles of the 32 datasets tested in our user study. The number of points in each class is separated by the symbol '/'. Bold items denote synthetic datasets generated with the Matlab random number generation toolbox.

Task	ID	No. of points in each class	ID	No. of points in each class
T1	**D1**	20000 / 20000 / 20000	D_2	1348 / 2812 / 5255
	D3	40000 / 40000 / 40000	D_4	11778 / 6135 / 18270
	D5	20000 / 40000 / 60000	D_6	13655 / 36785 / 44947
	D7	20000 / 20000 / 20000 / 20000	D_8	27233 / 54362 / 32611
	D9	40000 / 40000 / 40000 / 40000	D_{10}	1381 / 2845 / 1703 / 2972
	D11	20000 / 40000 / 60000 / 80000	D_{12}	5845 / 5419 / 5291 / 6452
	D_{13}	6046 / 10743 / 8204 / 7093	D_{14}	14614 / 23295 / 21540 / 13044
	D_{15}	5526 / 6458 / 9792 / 13012 / 10454	D_{16}	14000 / 11007 / 15096 / 21336 / 17260
T2	**D17**	25000 / 25000 / 25000	D_{18}	1794 / 6452 / 12086
	D19	50000 / 50000 / 50000	D_{20}	4499 / 4145 / 3976
	D21	20000 / 40000 / 80000	D_{22}	4250 / 3609 / 5624
	D23	25000 / 25000 / 25000 / 25000	D_{24}	7859 / 10123 / 8121
	D25	50000 / 50000 / 50000 / 50000	D_{26}	31801 / 39805 / 99797
	D27	20000 / 30000 / 50000 / 60000	D_{28}	21489 / 20611 / 28818 / 27582
	D_{29}	7050 / 18452 / 16438 / 24671	D_{30}	32549 / 28451 / 38645 / 27314
	D_{31}	6506 / 5482 / 3622 / 6600 / 10402	D_{32}	7141 / 34682 / 12086 / 26432 / 58862

and CW respectively. However, only 8 questions for C were answered with the *Hard to determine* option. This is because the color blending and weaving methods cannot preserve relative density features. They can hardly be employed for quantitative tasks.

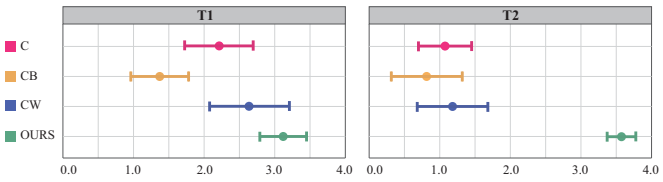


Fig. 11. Overview of performance across tasks and visualization schemes. Points show the average score of each visualization scheme on **T1** and **T2**. Lines represent one-standard errors.

6 Discussion and Future Work

The orders in which the points are drawn greatly influence the visualization results for conventional multi-class scatterplots. Figure 12 shows two visualizations with different orders of a dataset by Tableau (http://www.tableausoftware.com/). To study the patterns in a multi-class dataset with conventional scatterplots, users have to manually exchange the drawn orders. In contrast, our approach is order-independent and yields an occlusion-free result.

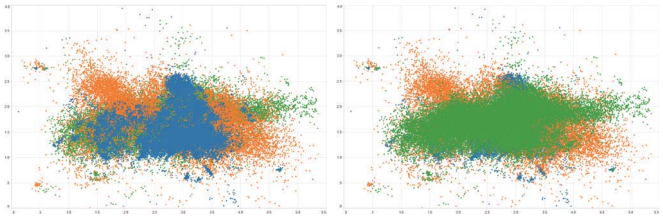


Fig. 12. Different drawn orders for the conventional multi-class scatterplot might cause distinct understandings.

Due to the limitations of the human visual perceptual system, the distance from the eye to the display also has influences on the comprehension of resulting visualizations. Points tend to be blended when users stay far away from the display. A possible solution would be to adaptively modulate the point radii. In practice, we can use depth cameras or eye tracking equipments to measure the eye-screen distance.

At a coarse zooming level, our method with the default sampling frequency parameter might generate relatively less points in medium and low density regions. To study features such as shapes or clusters of points in these regions, gradually increasing the sampling frequency or interactively zooming the view are two feasible solutions.

Although our approach aims at abstracting the multi-class scatterplots in a single view, it can be extended to visualize multi-variate datasets by employing the scatterplot matrix representation.

We see many directions for future work. 1) While the current sampling scheme cannot exactly preserve the ratios among classes, we intend to employ a soft disk sampling technique [41]. With this technique, a fixed number of samples in each class will be produced in a local region such that the relative ratios are preserved. 2) Our method scales well in the number of data points. Using colors to identify different classes limits the the number of data classes shown in a single view. We intend to find the capacity limit of class number in different situations. 3) To assist users identifying classes, our current solution only considers the color distinguishability. We intend to take other perceptual factors such as visual importance and attentions into account. 4) Compared with the ellipse representation, the dot-line representation has the capability to show global trends. However it might obscure the relative density features and introduce visual clutter due to the extra line geometries. We expect to evaluate the effectiveness of different visual representations to encode local trends in scatterplots. 5) We also would like to study the density variations in different regions after applying our approach.

7 Conclusion

This paper presents a preliminary work that employs an alternative method for multi-class scatter data visualization. While previous works leverage opacity, color, and other visual channels to eliminate the overdraw problem, ours utilizes spatial redistribution. The core is a hierarchical multi-class blue noise sampling scheme. As a key benefit, our method generates a visual abstraction of the input point distributions while the overdraw is alleviated. To help users identify different data classes, a color optimization technique is employed. Both case studies and the user study have demonstrated the effectiveness of our method. The visual exploration system equipped with a set of interaction tools enables users to study a multi-class scatter dataset from different perspectives.

Acknowledgments

This work was supported in part by the Major Program of National Natural Science Foundation of China (61232012), the National Natural Science Foundation of China (61202279), the National High Technology Research and Development Program of China (2012AA12090), the Zhejiang Provincial Natural Science Foundation of China (LR13F020001), the Doctoral Fund of Ministry of Education of China (20120101110134), the Fundamental Research Funds for the Central Universities, and the NUS-ZJU SeSama center.

References

[1] A. Asuncion and D. Newman. Uci machine learning repository, 2007.
[2] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1428–1435, 2008.
[3] S. Bachthaler and D. Weiskopf. Efficient and adaptive rendering of 2-d continuous scatterplots. In *Computer Graphics Forum*, volume 28, pages 743–750, 2009.
[4] E. Bertini and G. Santucci. Give chance a chance: modeling density to enhance scatter plot quality through random data sampling. *Information Visualization*, 5(2):95–110, 2006.
[5] T. Buering, J. Gerken, and H. Reiterer. User interaction with scatterplots on small screens-a comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):829–836, 2006.
[6] K. Burger, J. Kruger, and R. Westermann. Direct volume editing. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1388–1395, 2008.

[7] D. B. Carr, R. J. Littlefield, W. Nicholson, and J. Littlefield. Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82(398):424–436, 1987.

[8] Y.-H. Chan, C. Correa, and K.-L. Ma. Flow-based scatterplots for sensitivity analysis. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 43–50. IEEE, 2010.

[9] Y.-H. Chan, C. D. Correa, and K.-L. Ma. The generalized sensitivity scatterplot. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1768–1781, 2013.

[10] J. Chen, X. Ge, L.-Y. Wei, B. Wang, Y. Wang, H. Wang, Y. Fei, K.-L. Qian, J.-H. Yong, W. Wang, et al. Bilateral blue noise sampling. In *SIGGRAPH Asia 2013*, 2013.

[11] W. Chen, Z. Ding, S. Zhang, A. MacKay-Brandt, S. Correia, H. Qu, J. A. Crow, D. F. Tate, Z. Yan, and Q. Peng. A novel interface for interactive exploration of dti fibers. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1433–1440, 2009.

[12] J. Choo, C. Lee, C. K. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013.

[13] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009.

[14] A. Coninx, G.-P. Bonneau, J. Droulez, and G. Thibault. Visualization of uncertain scalar data fields using color scales and perceptually adapted noise. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, pages 59–66. ACM, 2011.

[15] T. N. Dang, A. Anand, and L. Wilkinson. Timeseer: Scagnostics for high-dimensional time series. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):470–483, 2013.

[16] T. N. Dang, L. Wilkinson, and A. Anand. Stacking graphic elements to avoid over-plotting. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1044–1052, 2010.

[17] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008.

[18] D. Feng, L. Kwock, Y. Lee, and R. M. Taylor. Matching visual saliency to confidence in plots of uncertain data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):980–989, 2010.

[19] M. Gleicher, M. Correll, C. Nothelfer, and S. Franconeri. Perception of average value in multiclass scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2316–2325, 2013.

[20] N. Gossett and B. Chen. Paint inspired color mixing and compositing for visualization. In *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004*, pages 113–118. IEEE, 2004.

[21] H. Guo, N. Mao, and X. Yuan. Wysiwyg (what you see is what you get) volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2106–2114, 2011.

[22] H. Hagh-Shenas, S. Kim, V. Interrante, and C. Healey. Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1270–1277, 2007.

[23] J. Huang, Z. Pan, G. Chen, W. Chen, and H. Bao. Image-space texture-based output-coherent surface flow visualization. *IEEE Trans. Vis. Comput. Graph.*, 19(9):1476–1487, 2013.

[24] H. Janetzko, M. C. Hao, S. Mittelstadt, U. Dayal, and D. Keim. Enhancing scatter plots using ellipsoid pixel placement and shading. In *46th Hawaii International Conference on System Sciences (HICSS)*, pages 1522–1531. IEEE, 2013.

[25] D. A. Keim, M. C. Hao, U. Dayal, H. Janetzko, and P. Bak. Generalized scatter plots. *Information Visualization*, 9(4):301–311, 2010.

[26] D. A. Keim and A. Herrmann. The gridfit algorithm: An efficient and effective approach to visualizing large amounts of spatial data. In *Proceedings of the Conference on Visualization '98*, pages 181–188, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[27] R. Khlebnikov, B. Kainz, M. Steinberger, and D. Schmalstieg. Noise-based volume rendering for the visualization of multivariate volumetric data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2926–2935, 2013.

[28] R. Khlebnikov, B. Kainz, M. Steinberger, M. Streit, and D. Schmalstieg. Procedural texture synthesis for zoom-independent visualization of multivariate data. 31(3pt4):1355–1364, 2012.

[29] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM J. on Optimization*, 9(1):112–147, 1998.

[30] J. Li, J.-B. Martens, and J. J. van Wijk. A model of symbol size discrimination in scatterplots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2553–2562. ACM, 2010.

[31] J. Li, J. J. van Wijk, and J.-B. Martens. Evaluation of symbol contrast in scatterplots. In *IEEE Pacific Visualization Symposium, 2009. PacificVis' 09*, pages 97–104. IEEE, 2009.

[32] M. Luboschik, A. Radloff, and H. Schumann. A new weaving technique for handling overlapping regions. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 25–32. ACM, 2010.

[33] A. Mayorga and M. Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1526–1538, 2013.

[34] A. Radloff, M. Luboschik, M. Sips, and H. Schumann. Supporting display scalability by redundant mapping. In *Advances in Visual Computing*, pages 472–483. Springer, 2011.

[35] W. Rankin. Cartography and the reality of boundaries. *Perspecta*, 42:42–45, 2010.

[36] R. Scheepens, N. Willems, H. van de Wetering, G. Andrienko, N. Andrienko, and J. J. van Wijk. Composite density maps for multivariate trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2518–2527, 2011.

[37] H.-W. Shen and D. L. Kao. A new line integral convolution algorithm for visualizing time-varying flow fields. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):98–108, 1998.

[38] B. Silverman. *Density estimation for statistics and data analysis*. Chapman & Hall/CRC, 1986.

[39] E. R. Tufte. *The visual display of Quantitative Information*. Graphics Press, Cheshire, CT, 2nd edition, 2002.

[40] J. J. Van Wijk. Spot noise texture synthesis for data visualization. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 309–318. ACM, 1991.

[41] L.-Y. Wei. Multi-class blue noise sampling. *ACM Trans. Graph*, 29(4), 2010.

[42] L. Wilkinson, A. Anand, and R. Grossman. High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1363–1372, 2006.

[43] N. Willems, H. Van De Wetering, and J. J. Van Wijk. Visualization of vessel movements. *Computer Graphics Forum*, 28(3):959–966, 2009.

[44] A. Woodruff, J. Landay, and M. Stonebraker. Constant density visualizations of non-uniform distributions of data. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 19–28. ACM, 1998.

[45] Y. Wu, X. Liu, S. Liu, and K.-L. Ma. Visizer: a visualization resizing framework. *IEEE Transactions on Visualization and Computer Graphics*, 19(2):278–290, 2013.

[46] Y.-C. Wu, Y.-T. Tsai, W.-C. Lin, and W.-H. Li. Generating pointillism paintings based on seurat's color composition. In *Computer Graphics Forum*, volume 32, pages 153–162. Wiley Online Library, 2013.

[47] X. Yuan, D. Ren, Z. Wang, and C. Guo. Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2625–2633, 2013.

[48] M. Zinsmaier, U. Brandes, O. Deussen, and H. Strobelt. Interactive level-of-detail rendering of large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2486–2495, 2012.