Doctoral theses at NTNU, 2022:307

Wenqiang Cui

Visualization Techniques for Interactive Visual Analysis of Multidimensional Big Data

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of Science and Technology

Wenqiang Cui

Visualization Techniques for Interactive Visual Analysis of Multidimensional Big Data

Thesis for the Degree of Philosophiae Doctor

Ålesund, October 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU

Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Computer Science

ISBN 978-82-326-6889-2 (printed ver.)
ISBN 978-82-326-5176-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2022:307

Printed by NTNU Grafisk senter

Abstract

Visual analysis has been used in many fields of research, such as health, biology, chemistry, social science, astronomy, and physics, to solve data-driven problems. Visualization is an effective tool to communicate, understand, extract information, and interact with data. However, the physical limitations of human visual perception prevent the direct visualization and understanding of multidimensional data. Projecting the data into a lower-dimensional space with a variety of dimensionality reduction techniques or mapping the data to parallel coordinates are two of the most widely used methods for visualizing multidimensional data. Interactive visual analysis plays an essential role in visual analytics to integrate human intelligence into visualization for knowledge discovery. The amount of multidimensional data available in various fields of research has been growing at a tremendous rate. Multidimensional big data brings new challenges and opportunities to visualization techniques for supporting interactive visual analysis.

This thesis presents a systematic review and two practical studies in the field of data visualization, focusing on interactive visual analysis of multidimensional big data. Specifically, it presents two scalable lightweight visualization techniques to solve the scalability challenge of parallel coordinates and dimensionality reduction techniques for supporting interactive visual analysis of multidimensional big data. The research for this thesis is based on several widely used benchmark multidimensional datasets obtained from public data repositories and synthesized datasets with hundreds of dimensions and millions of data points.

In the systematic review, I propose a novel taxonomy of state-of-the-art visual analytics applications based on the dimensionality of data and visualization, and the types of interactions, and summarize the challenges and future directions for interactive visual analysis of multidimensional big data. The results of the systematic review lead to the two practical studies. In the first practical study, I propose a

scalable lightweight bundling method to address the challenge of interactive visual analysis of multidimensional big data using parallel coordinates. It accelerates the clustering process of the data and helps users discover trends and detect outliers in the data by integrating human intelligence into the two-dimensional data binning using novel interactions. It uses the frequency-based representation to render the clusters as histogram-like bundles to reveal the distribution of the data, eliminate visual clutter and overplotting in parallel coordinates, and accelerate the rendering process. In the second practical study, I propose a scalable method, named ColorPCA, to address the challenge of automatically colorizing unlabeled multidimensional big data for discovering classes in the data. It combines principal component analysis and ray casting to compute the composite RGBA color of the data. It provides a fast way to enhance the visualization of the data in lower-dimensional space and help the users find suitable parameters of dimensionality reduction algorithms to balance the running time and the projection results.

Based on the two proposed visualization techniques, I have developed two web-based applications to support interactive visual analysis of multidimensional big data with parallel coordinates and lower-dimensional projections. The usefulness and effectiveness of the two proposed visualization techniques were demonstrated by case studies and user studies using the applications with benchmark datasets. The scalability of the two proposed visualization techniques were evaluated via scalability analysis with synthesized datasets. The experimental results show that the two proposed visualization techniques are well scalable for multidimensional big data. For example, the bundling method can support real-time interactions for clustering millions of multidimensional data records without pre-computation of the data and real-time visualization of the bundling result in web-based parallel coordinates plot without hardware-accelerated rendering. With a one-time pre-processing of the data, ColorPCA can colorize millions of multidimensional data points in real-time without hardware acceleration.

Acknowledgement

First of all, I would like to express my deepest gratitude to my supervisors, Girts Strazdins, Hans Georg Schaathun, Simon McCallum, and Anniken Susanne T. Karlsen.

As my main supervisor, Girts has offered me an open and friendly research environment. I am very grateful for his valuable guidance, suggestions and input to my research, his serious attitude towards research, and his patient and timely responses. Although Simon moved to New Zealand during my PhD study, I appreciate his encouragement and valuable ideas that helped me complete my first survey paper. I am deeply thankful to Hans and Anniken for their valuable suggestions and critical questions about my research and thesis.

I wish to thank Hao Wang for his valuable input on two of the included publications of this thesis.

Last but not least, I give my special gratitude to Roukang Yang, my beautiful and loving wife: no success in the world is worth it unless I can share it with you. I would like to thank my beloved parents, Wei Cui and Jinfang Wang, for their understanding and unconditional support over these years.

Contents

List of Figures

List of Tables

List of Abbreviations

PC	Parallel Coordinates
DR	Dimensionality Reduction
PCA	Principal Component Analysis
MDS	Multidimensional Scaling
LDA	Linear Discriminant Analysis
LLE	Locally-Linear Embedding
LE	Laplacian Eigenmaps
GPLVM	Gaussian Process Latent Variable Models
UMAP	Uniform Manifold Approximation and Projection
t-SNE	t-distributed stochastic neighbor embedding
KDD	Knowledge Discovery in Databases
VA	Visual Analytics
IVA	Interactive Visual Analysis

EB	Edge Bundling
CMV	Coordinated and Multiple Views
SVG	Scalable Vector Graphics

Chapter 1

Introduction

This chapter provides the details of the research background, motivation, and research questions in this thesis. It introduces two widely used methods for visualizing multidimensional data and the related research questions for interactive visual analysis of multidimensional big data. In addition, it lists the included publications and explained their contribution to the research questions. Finally, it outlines the structure of this thesis.

1.1 Background

Visual analysis has been used in many fields of study, such as health [1], biology [2, 3], environment [4, 5, 6], chemistry [7], astronomy [8, 9], and physics [10, 11], to solve data-driven problems. Visualization is an effective tool to communicate, understand, extract information, and interact with multidimensional data. However, the physical limitations of human visual perception prevent the direct visualization and understanding of multidimensional data, thereby, the data must be represented in a lower-dimensional space, usually of two or three dimensions. Mapping the data to parallel coordinates (PC) and projecting the data with a variety of dimensionality reduction (DR) algorithms are two of the most widely used methods for visualizing multidimensional data in lower-dimensional space.

1.1.1 Parallel Coordinates Plots

PC is widely used and has become a standard tool for visualizing multidimensional data [12]. In PC plots, axes corresponding to the number of dimensions are aligned parallel to each other, and multidimensional data points are mapped to polylines

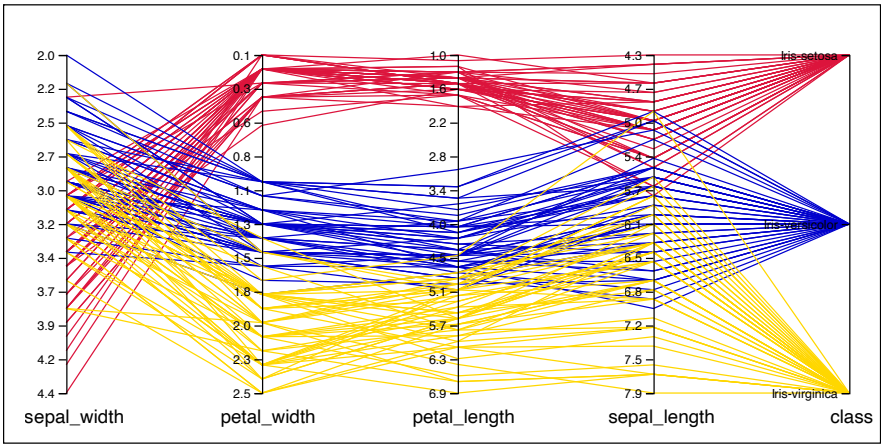


Figure 1.1: The PC plot of the Iris dataset.

(or edges) intersecting the axes at their respective values [13, 14, 15]. The embedding of an arbitrary number of parallel axes in PC plots allows for the simultaneous display of multiple dimensions to provide an overview of the structure of the data and reveal intrinsic patterns and trends of the data.

Figure 1.1 shows an example PC plot of the Iris dataset [16]. The Iris dataset is one of the most popular multidimensional datasets in pattern recognition domain. It contains 150 instances for three different species of Iris, including Iris-setosa, Iris-virginica, and Iris-versicolor. Each specie has 50 instances. Each instance has four measurements (dimensions): sepal length, sepal width, petal length, and petal width. As shown in Figure 1.1, to visualize the Iris dataset with the PC plot, the axes for the four dimensions of the dataset are aligned parallel to each other and the data instances are mapped to the axes as polylines, where the data for the three species of Iris are colorized in red, blue, and yellow, respectively. In Figure 1.1, the polylines in three colors reveal the distribution of the data and the relational patterns between each two adjacent axes.

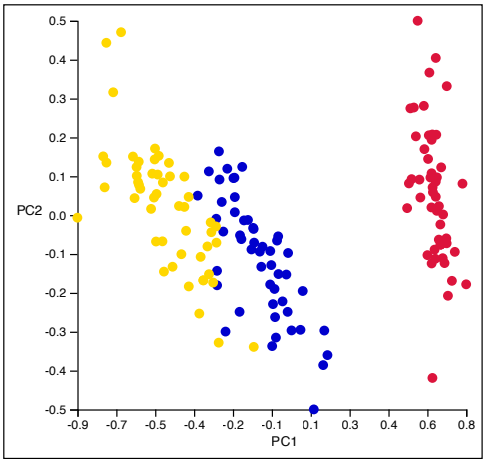


Figure 1.2: The PCA plot of the Iris dataset.

1.1.2 Dimensionality Reduction

DR is one of the fundamental techniques for analyzing and visualizing multidimensional data [17]. It reduces the dimensionality of multidimensional data by projecting the data into a new orthogonal coordinate system with lower dimensionality, and uses the location of the data points in the projected space to retain the structure of the data in their original definition space, such as correlation, neighborhood and distance relationships, and classes/clusters in the data. To reduce the dimensionality of multidimensional data, a variety of linear methods, such as principal component analysis (PCA) [18], multidimensional scaling (MDS) [19], and linear discriminant analysis (LDA) [20], and nonlinear methods, such as kernel principal component analysis (Kernel PCA) [21], Isomap [22], locally-linear embedding (LLE) [23] and its variants (Hessian LLE [24] and MLLE [25]), Laplacian eigenmaps (LE) [26], Gaussian process latent variable models (GPLVM) [27], uniform manifold approximation and projection (UMAP) [28] and t-distributed stochastic neighbor embedding (t-SNE) [29] have been widely investigated.

Figure 1.2 shows an example of visualizing the Iris dataset with PCA. PCA is one of the most widely used DR methods. It uses an orthogonal linear transformation

based on the eigendecomposition of the covariance matrix of the data to transform the data into a new orthogonal coordinate system such that the greatest variance by any projection of the data comes to lie on the first principal component, the second greatest variance on the second principal component, and so on. In Figure 1.2, the Iris data is projected into the two-dimensional orthogonal coordinate system with the first two principal components of PCA. As shown in Figure 1.2, the data points for the three species of Iris are colorized in red, blue, and yellow, respectively, which forms three clusters and reveals the structure of the Iris data in the two-dimensional visualization.

1.1.3 Interactive Visual Analysis

Visual analytics (VA) is an active research field for effectively understanding, reasoning, and decision making on the basis of massive and complex data by combining automated analysis techniques with the intellectual strengths of the human through interactive visual analysis (IVA). IVA combines visualization and interactions as a powerful tool to enable the user to visually discover explainable patterns from data with the computation/visualization-interaction loop. In the loop, the user can discover knowledge in data through visual representations of the data and apply the knowledge to iteratively refine the visual representations or the underlying data analysis models by interactions.

PC is widely used for the IVA of multidimensional data, such as the climate simulation data [30] and the hurricane trends data [31]. Brushing [32, 33] and axes-reordering [34, 35] are the two typical interactions for PC to help the users visually explore the data. Brushing a range on a single axis or an area between two adjacent axes is used to select a subset of data. The selected subset of data is then used as input for subsequent operations, such as highlighting, labeling, bundling, and many more [36]. The order of the axes of PC has a large effect on the patterns emerging from the visualization of data, as they define the shape and position of each individual polyline in PC. Manually changing the order of the axes of PC with a drag-and-drop interaction can help the users visually discover different patterns from multidimensional data.

Many DR-based VA applications [37, 38, 39, 40] have demonstrated the benefits of applying DR algorithms to visually discover understandable patterns from multidimensional data in the projected lower-dimensional space. The parameters and settings of DR algorithms have a large impact on the patterns emerging from the visualization of data in the projections. In addition to common interactions such as navigating, filtering, brushing, etc. for visualization [41], configuring parameters and constraints is a typical type of interactions for DR-based IVA [42]. For example, [43] presents an interactive DR framework to provide flexibility for comparative analysis tasks by enabling the users to interactively tune parameters of an optimization algorithm for refining the DR results. [44] introduces a DR-based VA system that enables a quality-guided investigation of the trade-off between the number of variables and loss of information by allowing interactive control of parameters affecting the DR results.

1.2 Challenges for Multidimensional Big Data

The PC and DR-based IVA plays an important role in VA of multidimensional data. The amount of multidimensional data available in various fields of research has been growing at a tremendous rate. Therefore, scalability has become the most significant challenge of PC and DR algorithms for supporting IVA of multidimensional big data.

Visualizing multidimensional big data with PC can create visual clutter to overwhelm the screen space and prevent the users from discovering patterns in the data. Visual clutter is caused by the polylines that overlap and cross each other. Figure 1.3 shows the PC plot of the office dataset [45]. The office dataset is a 5-dimensional dataset with 20,560 data points. As shown in Figure 1.3, the blue polylines cover the areas between the axes and most of the red polylines, which hides the patterns in the data.

More importantly, due the limited scalability of PC, rendering data as polylines instead of points can significantly increase the rendering time for multidimensional big data, especially when hardware-accelerated rendering is not available, such as in web-based visualization. For example, the experimental results in Appendix Pa-

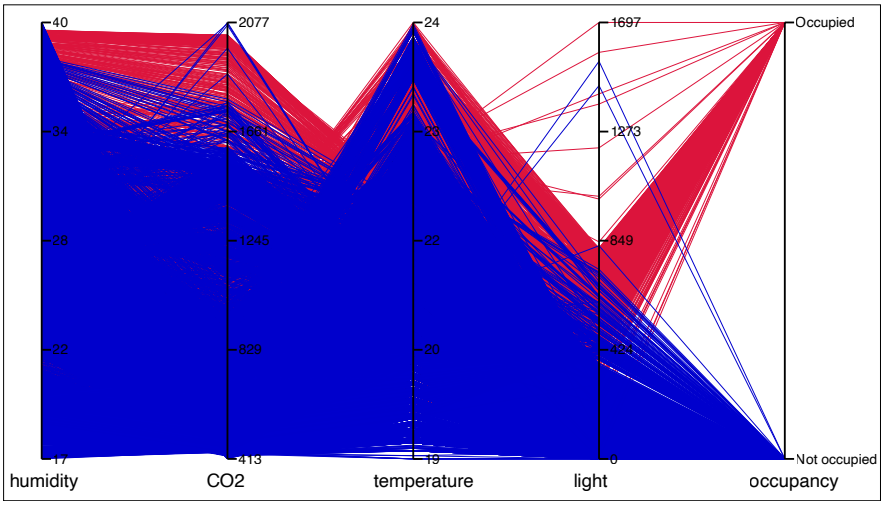
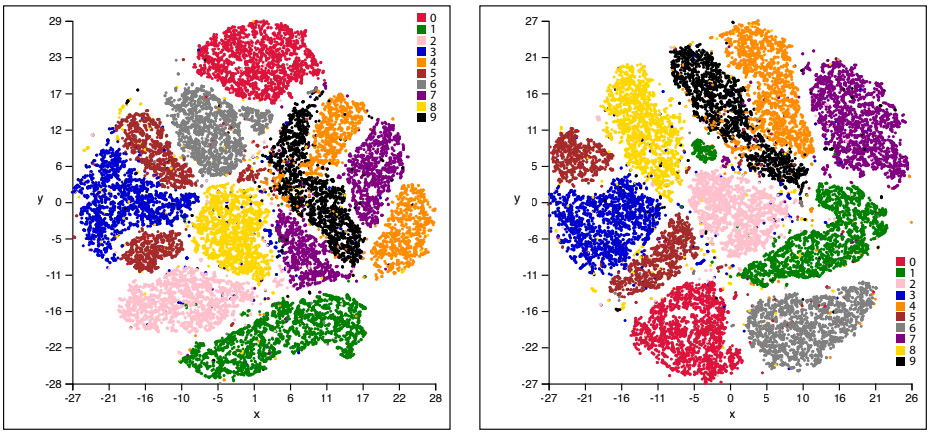


Figure 1.3: The PC plot of the office dataset.

per 2 show that, in web-based visualization, rendering half a million 6-dimensional data points in a PC plot with SVG takes 9 seconds, and rendering one million 6-dimensional data points crashes the web browser. The time-consuming rendering process can significantly delay the visual response of the interactions that require re-rendering of the data, such as brushing and axes-reordering, which hinders the users from gaining insight into data by exploring the patterns emerging from the visualization of data.

Edge bundling (EB) is a common technique used to reduce visual clutter and create more informative visualization in PC plots [46, 47]. It uses varied data clustering algorithms to cluster the data, and render the clustered data as the bundles in different forms. Based on the pre-computation of the clustering at different levels of detail, the users can choose the bundled visualization at different levels of abstraction. However, for multidimensional big data, the limited scalability of the underlying clustering algorithms of existing EB techniques cannot support the interactions that require re-clustering of the data, such as changing the bundling results based on users' perception and axes-reordering, which reduces the usability and usefulness of EB techniques for PC-based IVA of multidimensional big data.



(a) (b)

Figure 1.4: The t-SNE plots of the MNIST dataset.

The scalability of DR algorithms is a significant challenge for DR-based IVA of multidimensional big data. For example, it can take minutes to hours to compute the PCA or t-SNE projection of a multidimensional dataset with millions of data points. The limited scalability of DR algorithms cannot support the interactions that require re-computation of the data, such as configuring parameters and constraints of DR algorithms, which hinders the users from visually exploring the patterns emerging from the lower-dimensional visualization of the data.

For labeled multidimensional data, colorizing the data by label is a common approach to enhance the lower-dimensional projections or mappings of the data. For example, in Figure 1.1, the polylines for the three species of Iris flowers are colorized in three colors by the label to distinguish overlapping polylines and reveal the patterns of polylines for each specie. In Figure 1.2, the clusters for the three species of Iris flowers are colorized in three colors by the label to reveal the cluster for each specie.

In addition, colorizing the data by label can provide an extra hint for configuring parameters and constraints of DR algorithms. For example, Figure 1.4 shows the

two t-SNE plots of the same subset of the MNIST dataset [48]. The MNIST dataset is a 784-dimensional dataset of 70000 images of handwritten digits from 0 to 9. The two plots are generated with different perplexity parameters of t-SNE. In the two plots, the data points for each digit are colorized with a distinct color by the label of the dataset. As shown in Figure 1.4a, the data points for digits 4 (orange), 7 (purple), and 5 (brown) are incorrectly projected to two separated clusters, respectively. In contrast to Figure 1.4a, the t-SNE plot in Figure 1.4b uses a better perplexity parameter, because the data points for digits 4 (orange) and 7 (purple) are correctly projected to a single cluster, respectively.

Several methods have been investigated for mapping unlabeled multidimensional data into colors for different purposes. However, automatically colorizing unlabeled multidimensional data to distinguish classes in the data and help the users tune parameters of DR algorithms is still a challenge, especially for big data.

1.3 Motivation and Objectives

IVA plays an essential role in integrating human cognitive, perceptual and reasoning abilities, and their knowledge into the analysis process to gain insight into data. Addressing the scalability challenges to PC and DR algorithms is helpful to support IVA of multidimensional big data. As the amount of multidimensional data has been growing at a tremendous rate in many fields of study, developing scalable PC or DR-based visualization methods will provide tools to facilitate data-driven research.

The goal of this thesis is to propose novel methods to address the scalability challenges of PC and DR algorithms and build web-based applications for supporting IVA of multidimensional big data. The research for this thesis is based on several widely used benchmark multidimensional datasets obtained from public data repositories and synthesized datasets with hundreds of dimensions and millions of data points.

The following research questions were formulated to accomplish the research objectives:

RQ1 What are the state-of-the-art techniques for interactive visual analysis of multidimensional data?

RQ2 What are the challenges for interactive visual analysis of multidimensional big data?

RQ3 For parallel coordinates, how to make the edge bundling process scalable by simplifying the underlying clustering algorithms and integrating human perception and judgments into the clustering process for supporting interactive visual analysis of multidimensional big data?

RQ4 For parallel coordinates, how to accelerate the rendering process of multidimensional big data without hardware-accelerated rendering using scalable edge bundling methods?

RQ5 How to automatically colorize unlabeled multidimensional big data to discover classes in the data and enhance the visualization of the data in lower-dimensional space?

RQ6 How to support interactive visual analysis of unlabeled multidimensional big data by combining human perception of color and the automatic colorization of the data?

The research of this thesis mainly includes three aspects: 1) a systematic review of VA; 2) a scalable lightweight EB method for PC; 3) a scalable method for automatically colorizing unlabeled multidimensional big data.

To answer the research questions **RQ1** and **RQ2**, I surveyed over 200 publications to construct a systematic review of VA with a focus on the dimensionality of data. In the review, I proposed a novel categorization of VA applications with a focus on the dimensionality of data and discussed the major challenges of VA, especially for big data, and future directions of VA. The results of the review raise the research questions **RQ3 - 6** and lead to the two practical studies in this thesis.

To answer the research questions **RQ3** and **RQ4**, in the first practical study, I proposed a scalable lightweight EB method for visualizing multidimensional big data in PC plots. It integrates human judgments into the two-dimensional data binning by novel interactions to accelerate the clustering process of the data and uses the frequency-based representation to render the clusters as histogram-like bundles to reveal the structure of the data, reduce visual clutter and accelerate the rendering process.

To answer the research questions **RQ5** and **RQ6**, in the second practical study, I proposed a scalable colorization algorithm, named ColorPCA, for unlabeled multidimensional big data. It integrates PCA and ray casting to automatically colorize the data for discovering classes in the data. It provides a fast way to enhance the visualization of unlabeled multidimensional big data in lower-dimensional projections to help the users tune parameters of DR algorithms, and allows the users to interactively explore the data using their perception of color.

In addition, I have developed two web-based VA applications based on the two proposed methods to support IVA of multidimensional big data. The usefulness and effectiveness of the proposed methods are demonstrated by case studies and user studies using the applications with benchmark datasets. The scalability of the proposed methods is evaluated via scalability analysis with synthesized datasets. The experimental results show that the proposed methods are well scalable for multidimensional big data. For example, the EB method can support real-time interactions for clustering millions of multidimensional data records without precomputation of the data and real-time visualization of the bundling result in web-based PC plot without hardware-accelerated rendering. With a one-time preprocessing of the data, ColorPCA can colorize millions of multidimensional data points in real-time without hardware acceleration.

1.4 List of Publication and Contribution

The thesis is based on a collection of four papers, which presents a systematic review and two practical studies in the field of data visualization with a focus on IVA of multidimensional big data. The list of the included papers and their main

contributions is given below.

[**Paper 1**] **Wenqiang Cui**, Visual Analytics: A Comprehensive Overview, in IEEE Access, vol. 7, pp. 81555-81573, 2019, doi: 10.1109/ACCESS.2019.2923736.

Contribution: In this paper, over 200 publications have been surveyed to construct an organized overview of VA, which examines the evolution of VA from visualization and algorithmic data analysis, and investigates how VA is applied in various application domains. In this paper, I propose a novel taxonomy of VA applications based on the dimensionality of data and visualization techniques, and the type of interactions, which presents the state-of-the-art VA techniques and applications in different application domains to bridge the gap between the challenges of discovering knowledge in large and complex datasets and VA solutions. In addition, I discuss major challenges of VA, especially for big data, and future directions of VA.

[**Paper 2**] **Wenqiang Cui**, Girts Strazdins, and Hao Wang, Web-based Scalable Visual Exploration of Large Multidimensional Data Using Human-in-the-Loop Edge Bundling in Parallel Coordinates. CEUR Workshop Proceedings, 2020.

Contribution: In this paper, I propose a scalable EB method for visualizing multidimensional big data in PC plots. Based on the proposed method, a prototype web-based VA application is built to support IVA of multidimensional big data with PC plots. With the data binning-based clustering, the proposed EB method uses novel interactions, including splitting, adjusting, and merging clusters, to integrate human perception and judgment into the bundling process. The case study and scalability analysis of the proposed EB method are conducted to demonstrate its effectiveness and scalability.

[**Paper 3**] **Wenqiang Cui**, Girts Strazdins, and Hao Wang, Visual Analysis of Multidimensional Big Data: A Scalable Lightweight Bundling Method for Parallel Coordinates, in IEEE Transactions on Big Data, doi: 10.1109/TBDATA.2021.3123982. *Early Access.*

Contribution: This paper is an extended work of Paper 2. In this paper, I improve the method described in Paper 2 with the new alignment of the bundles and the new interactions. I integrate human judgments into the two-dimensional data binning by novel interactions to accelerate the clustering process of the data, and use the frequency-based representation to render the clusters as histogram-like bundles to reveal the distribution of the data, eliminate the overplotting of the bundles and accelerate the rendering process. I also compare the design of the proposed method with five state-of-the-art edge bundled PC plots with a focus on visualizing multidimensional big data. Based on the proposed method, I have developed a lightweight web-based VA application for exploring multidimensional big data in PC plots. The experiments are conducted to analyze the scalability of the proposed method. Two case studies and a user study are conducted to compare the proposed method with classic PC plots and two state-of-the-art edge-bundled PC plots. The results show that the proposed method is more efficient and effective for visually analyzing multidimensional big data. It allows the user to interactively optimize the visualization with their judgments to solve data-related problems, such as discovering major trends and specific patterns in the data, estimating the correlation between variables, and detecting outliers in the data.

[**Paper 4**] **Wenqiang Cui**, ColorPCA: A Scalable Method for Colorizing Unlabeled Multidimensional Big Data. *Under Review.*

Contribution: This paper presents a scalable method, named ColorPCA, to address the challenge of automatically colorizing unlabeled multidimensional big data for discovering classes in the data. ColorPCA integrates PCA and ray casting to map unlabeled multidimensional big data into the RGBA color space, where different classes of the data are mapped to different locations in the color space and thus colorized with different colors or color combinations. The scalability analysis of ColorPCA shows that its running time increases linearly with the number of data points or the number of dimensions. After a one-time pre-processing of the data, ColorPCA can map one million 50-dimensional data points into colors in 0.97 seconds on desktop hardware. The case studies and the user study show that ColorPCA can enhance the lower-dimensional projections of unlabeled multidi-

mensional big data by automatically colorizing the data to help the user discover classes in the data and find suitable parameters of DR algorithms to balance the running time and the projection results.

1.5 Thesis Structure and Overview

The thesis is written in the form of a collection of research papers. Figure 1.5 shows the relationship between the research questions described in Section 1.3 and the papers listed in Section 1.4. Paper 1 covers the research questions **RQ1** and **RQ2** with the systematic review of VA. With the focus on visualizing multidimensional big data in PC plots, Paper 2 and Paper 3 address the research questions **RQ3** and **RQ4** by proposing the scalable lightweight EB method for PC. The research questions **RQ5** and **RQ6** are covered by Paper 4 with the focus on automatically colorization of unlabeled multidimensional big data for discovering classes in the data.

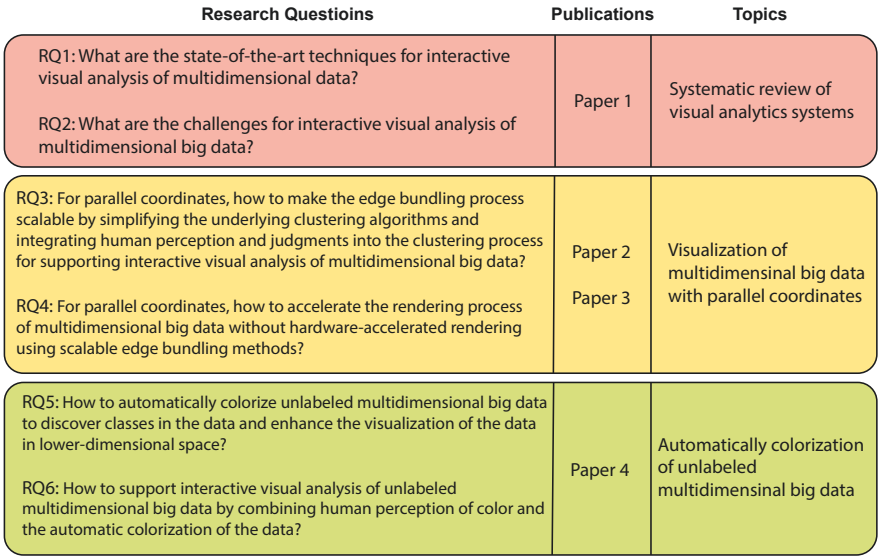


Figure 1.5: Relationship between the research questions and the included papers.

The structure of this thesis is as follows:

Chapter 1 introduces an overview of the thesis, which includes research background, motivation, objectives and questions, and the list of publications.

Chapter 2 presents the literature review of IVA of multidimensional big data, which includes the relevant results selected from the systematic review (Paper 1), the related work of visualizing multidimensional big data with edge-bundled PC plots (Paper 3), and the related work of colorizing unlabeled multidimensional big data (Paper 4). More details of the systematic review can be found in Paper 1.

Chapter 3 describes the proposed methods and the applications for supporting IVA of multidimensional big data. For exploring multidimensional big data with PC, this thesis presents a scalable lightweight EB method and a web-based application (Paper 2 and 3). For automatically colorizing unlabeled multidimensional big data, this thesis presents a scalable colorization method, named ColorPCA, and a web-based application (Paper 4).

Chapter 4 presents the evaluation results and the discussion of the proposed EB method (Paper 2 and 3) and ColorPCA (Paper 4). For each method, the evaluation includes scalability analysis, case studies and user study.

Chapter 5 summarizes the main contributions of this research work, and discusses future research directions.

Chapter 2

Literature review

This chapter presents the literature review of IVA of multidimensional big data (Paper 1), the related work of the edge-bundled PC plot (Paper 2 and 3) and the colorization of unlabeled multidimensional big data (Paper 4).

2.1 Systematic Review of Visual Analytics

Leading by the research questions **RQ1** and **RQ2**, I have conducted a systematic review of VA and its applications. This section presents the relevant results selected from the systematic review with a focus on multidimensional big data, which include 1) a new categorization of VA applications, and 2) the challenges and future directions for IVA of multidimensional big data. More details of the systematic review can be found in Paper 1.

Sun et al. [49] identified five categories of VA applications according to the type of considered data. In their research, VA applications were classified as *space and time*, *multivariate*, *text*, *graph and network*, and *other applications*. However, they did not provide a comprehensive classification. Firstly, it is difficult to categorize a VA application which deals with several different types of data at the same time. For example, Chen et al. [50] proposed a VA system to analyze and explore multiple types of data and correlate them for intelligence analysis. The data analyzed in their system included GPS logs, which contained spatial and temporal data, news and email headers, which are textual data, and transaction logs which contained network data. Secondly, a complex data set may have two or more characteristics so that the corresponding VA application will be classified into two or more

categories simultaneously. For example, Andrienko et al. [51] analyzed streaming-tweets data which consisted of geographical coordinates, time of tweeting, and the tweet text itself in their VA system. According to the classifications of [49], this VA application can be classified into the category *space and time* as well as the category *text*. Furthermore, with the rapid development of VA in different domains, increasing numbers of VA applications will be classified into the category *other applications*.

To address the challenges arising from the limitations of the classification scheme of [49], and direct the future research of VA, in this research, a new comprehensive taxonomy of VA applications is proposed based on the dimensionality of data and visualization techniques, and the type of interactions.

2.1.1 Visualization Based Classification

According to the dimensionality of the data and visualization techniques, four categories of VA applications are identified: **2D-to-2D**, **multidimensional-reduction-2D**, **multidimensional-to-2D**, and **multidimensional-to-3D**.

2D-to-2D

Two-dimensional (2D) visualization is the most common way to visualizing data in information visualization. Within 2D visualization, binary data is naturally visualized in 2D space through the Cartesian coordinate system. A VA application will be classified as **2D-to-2D** if it fulfills the following requirements:

- The data is 2D.
- The data are visualized in 2D visualization.

For VA applications in this category, users gain insight from data by performing analytical reasoning on 2D data through 2D visualization. For example, Bögl et al. [52] developed a **2D-to-2D** VA application (TiMoVA) to guide domain experts in model-selection tasks based on user stories and iterative expert feedback on users' experiences. It closely combined human perception and analytical reasoning and automated computation. Figure 2.1 shows an overview of TiMoVA.

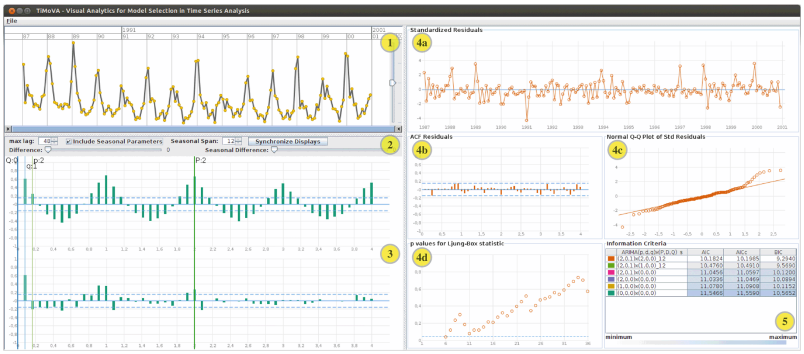


Figure 2.1: TiMoVA Overview [52]. A **2D-to-2D** VA application for finding an adequate model for a given time-oriented dataset.

In addition, **2D-to-2D** VA applications are commonly used for another type of 2D data: movement data. The research of [53, 54] used different 2D visualization techniques, such as mapping and clustering movement data on 2D maps, to analyze and explore various aspects of movement through VA.

Multidimensional-reduction-2D

With the ever-increasing amount of datasets, multidimensional data show up in numerous fields of study, such as economics, biology, chemistry, political science, astronomy, and physics. However, the high dimensionality of a multidimensional data represents a critical obstacle: humans are biologically optimized to see the world and the patterns in it in three dimensions [55]. This challenge and the wide availability of multidimensional data have led to new opportunities for VA.

A VA application will be classified as **multidimensional-reduction-2D** if it fulfills the following requirements:

- The data is multidimensional.
- The dimensionality of the data is reduced by algorithmic approaches to two dimensions.
- The processed data are visualized in 2D visualization.

To break the physical limitations of the human visual system, a variety of DR methods have been investigated for reducing the dimensions of multidimensional data, such as PCA, MDS, LDA and t-SNE. By allowing users explore multidimensional data in 2D space, **multidimensional-reduction-2D** VA applications combine DR methods and the human analytical reasoning ability to help users understand and interpret the result of algorithmic DR methods in an intuitive and meaningful manner. For example, Choo et al. [56] presented a **multidimensional-reduction-2D** VA system (iVisClassifier) for classifications based on a supervised DR approach, which is shown in Figure 2.2.

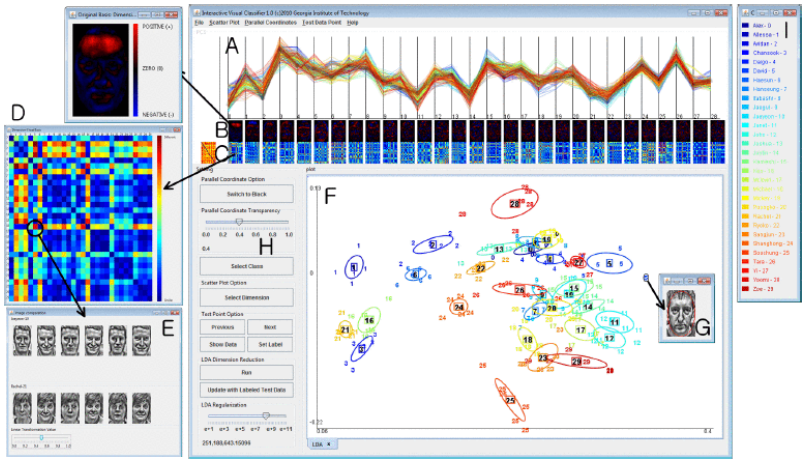


Figure 2.2: An overview of iVisClassifier [56].

Wu et al. [57] introduced a **multidimensional-reduction-2D** VA system (OpinionFlow) to empower analysts to detect opinion-propagation patterns and glean insights, which is shown in Figure 2.3. OpinionFlow uses an information diffusion model to reduce the dimension of the social-media data.

Multidimensional-transformation-2D

Another category of VA applications is **multidimensional-transformation-2D**, which visualizes multidimensional data without algorithmic DR methods. A VA application will be classified as **multidimensional-transformation-2D** if it fulfills the following requirements:

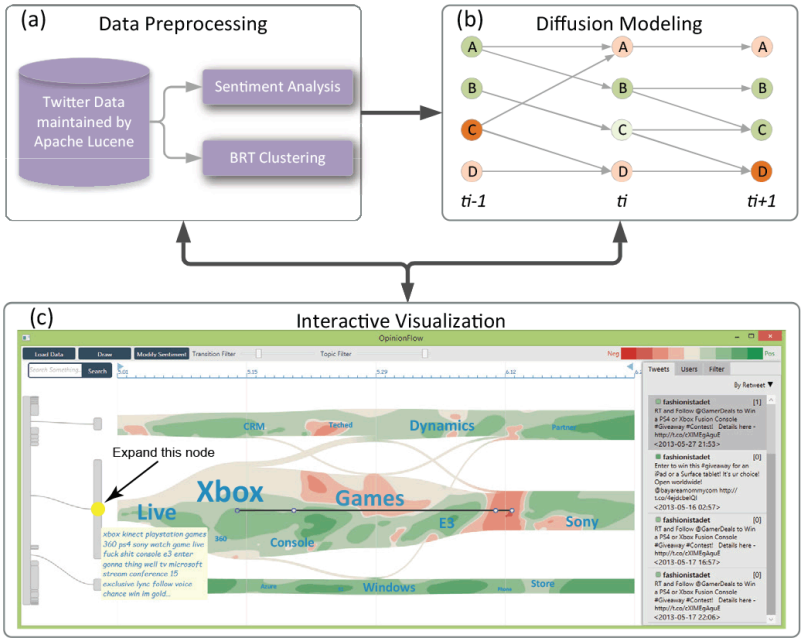


Figure 2.3: Three major parts of OpinionFlow: (a) Data preprocessing, (b) diffusion modeling, and (c) interactive visualization [57].

– The data is multidimensional.

– The multidimensional data is transformed and mapped in 2D visualization.

– The dimension of the data is not reduced by algorithmic DR methods.

Within **multidimensional-transformation-2D** VA applications, multidimensional data is transformed and mapped in 2D space, which encodes data to different representations, such as PC plots and coordinated and multiple views (CMV). CMV encompass a specific exploratory visualization technique that uses two or more distinct views to support the investigation of a single conceptual entity [58]. Guo et al. [59] presented a **multidimensional-transformation-2D** VA system, Triple Perspective Visual Trajectory Analytics (TripVista), for exploring and analyzing complex traffic trajectory data, which was mainly based on a PC plot and

CMV. TripVista is shown in Figure 2.4.

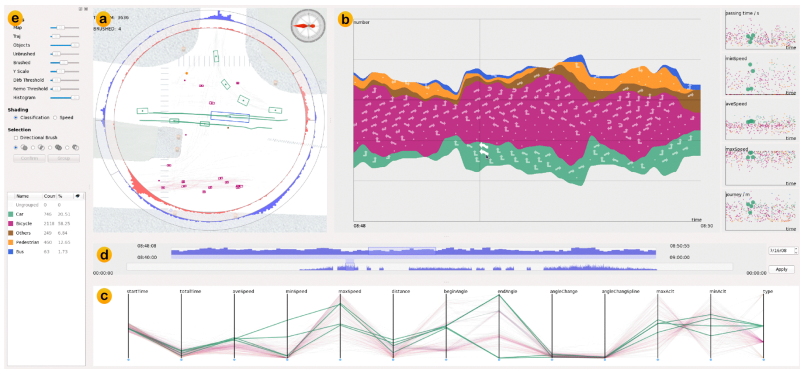


Figure 2.4: TripVista overview [59].

Multidimensional-to-3D

Three-dimensional (3D) visualization was developed for converting 3D objects/phenomena into 2D images through a computer-graphics process. Presently, 3D visualization is widely used in scientific visualization to enable scientists to understand and graphically illustrate the data. Moreover, 3D visualization is often integrated with a variety of approaches to visually analyze multidimensional data. For example Achtert et al. [60] and Johansson et al. [61] visualized parallel coordinates in 3D space to explore the complicated relationships between the axes, which arranged more than two neighboring axes around the central attribute.

Multidimensional-to-3D VA applications are based on the 3D visualization of multidimensional data. A VA application will be classified as **multidimensional-to-3D** if it fulfills the following requirements:

– The data is multidimensional.

– The multidimensional data is transformed and mapped in 3D visualization.

For example, Kurzhals and Weiskopf [62] introduced a **multidimensional-to-3D** VA application to analyze eye-tracking data recorded for dynamic stimuli such as video or animated graphics, which is shown in Figure 2.5.

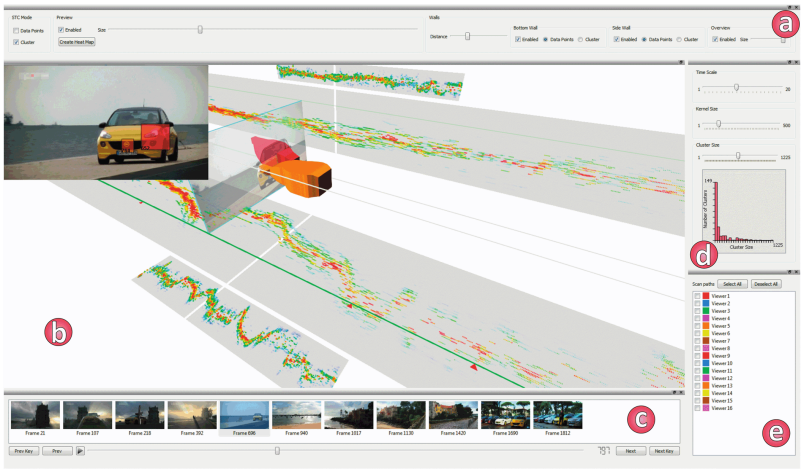


Figure 2.5: A **multidimensional-to-3D** VA application for eye-tracking data [62].

2.1.2 Interaction Based Classification

In VA, analytical reasoning process is facilitated by interactive visual analysis of data through various interaction techniques. According to the VA process, users can directly interact with data, algorithms, and visualization [63]. Heer and Shneiderman [64] give a taxonomy of interactive dynamics for visual analysis, which includes data and view specification (filtering, sorting, deriving values or models from source data, etc.), view manipulation (selecting, navigation, etc.), process and provenance (recording, guiding or sharing, etc.). Endert et al. [65] divide interactions into two categories *exploratory* and *expressive* from observation-level.

In this research, I combine Heer and Shneiderman' taxonomy [64] and Endert et al.'s classification [65] to classify VA applications from the interaction perspective. VA applications are classified into two categories: **exploratory-oriented** and **expressive-oriented** based on their interactions. One application will be classified into the category **exploratory-oriented** if its interactions are designed to explore data and visualization space. For example, the interactions of selecting a different encoding, modifying zoom levels, and filtering data are considered as

exploratory-oriented. With **exploratory-oriented** VA applications, users gain insight into data by observing how data reacts to the interactions in a dynamic visual representation.

One application will be classified into the category **expressive-oriented** if its interactions are designed to change the algorithms for rendering the visualization or the underlying models for data analysis. The interactions of modifying parameters of underlying mathematical models or rendering algorithms, and deriving values or models from source data are considered as **expressive-oriented**. Within **expressive-oriented** VA applications, the interactions are therefore commonly coupled with algorithmic data analysis process. For example, Interactive Principal Component Analysis (iPCA) [66] changes the weight for each dimension in calculating the direction of projection using multiple sliders through user interactions. Also, in a VA application using MDS [67], the dissimilarities in the calculation of the stress function can be weighted through visual controls.

Furthermore, both the two types of interactions may be used by one application at the same time. Accordingly, there is an overlap between the categories of *exploratory-oriented* and *expressive-oriented* in the interaction based classification.

2.1.3 A Complete Taxonomy of Visual Analytics Applications

From a technical perspective, visualization and interactions based classifications form a complete categorization of VA applications. Figure 2.6 illustrates the relationship of two classifications of VA applications. It covers all technical components of state-of-the-art VA applications, including 2D and 3D visualization techniques, algorithmic DR methods and data analysis methods, and exploratory and expressive interactions. This categorization can direct researchers to select appropriate techniques for building VA applications on complex datasets. Table 3 in Paper 1 shows the complete categorization of the approximately 80 VA applications examined in the systematic review.

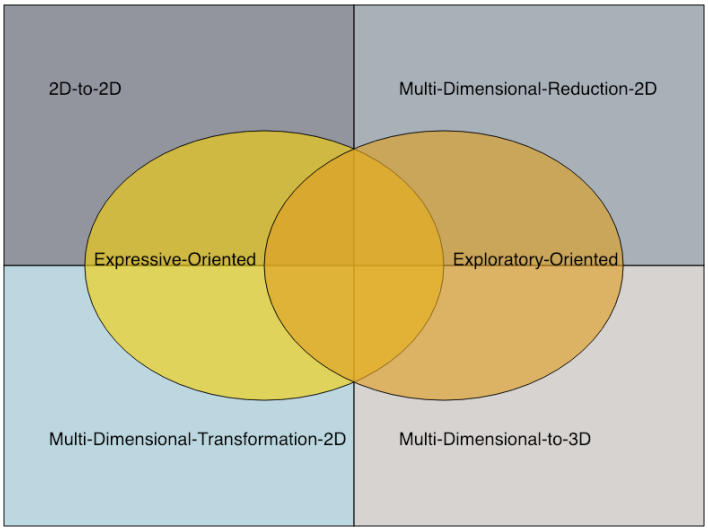


Figure 2.6: The complete categorization of VA applications.

2.1.4 Challenges for Interactive Visual Analysis of Multidimensional Big Data

Scalability

With the explosion of data, IVA techniques need be able to scale with both the size and dimension of data. However, there is a growing mismatch between data size/complexity and the human ability to explore and interact with data [68], which makes scalability is becoming a fundamental challenge for IVA of multidimensional big data.

The scalability of VA is defined as "its capability to effectively display large data sets in terms of either the number or the dimension of individual data elements" [69]. Presently, most research in improving scalability of VA applications is primarily focused on investigating visualization devices [70]. For example, large-scale high-resolution displays [71, 72] have been investigated to display more overview and detail for big data in VA research. However, compared with the amount of data which is continuously growing at a rapid pace, the amount of

pixels on the display still remains rather constant. In this case, the amount of data still commonly exceeds the limited amount of pixels of a display by several orders of magnitude. In addition, although it is possible to build ever-larger and higher-resolution displays, human visual acuity is limited to match the extreme large-screen approach. DR methods haven been widely used for IVA of multidimensional data. However, the use of DR methods has been somewhat limited in VA because they are too slow for interactive use when the number of dimensions or data points is scaled up [68]. This significantly hinders the integration of human perception and judgment into the data analysis process. More importantly, more dimension reduction and a higher rate of compression to data on displays mean more abstract representations and more lost details [73], which requires additional interpretation for performing analytical reasoning.

The scalability challenge for IVA of multidimensional big data involves both human and machine limitations. It is expected to integrate scalable algorithms and visualization techniques for the data to reduce the mismatch between the size and complexity of data and human ability.

Interaction

In the systematic review, interactions for VA are classified into two categories: **exploratory-oriented** and **expressive-oriented**. Both of them are equally important to VA applications. However, the results of the systematic review show that **expressive-oriented** interactions are much less used in recent VA applications than **exploratory-oriented** interactions (see Table 3 in Paper 1). Only a few applications have tried to use these two different kinds of interactions together, such as [56, 74, 62, 75]. One of most possible reasons for this situation is that **expressive-oriented** interactions are associated with the modification of underlying mathematical models or rendering algorithms, which may delay the visual response of the interactions when the size and dimension of the input data are scaled up.

In addition, there have been rapid advances in interaction technologies, however, their advantages have not been fully investigated in VA, as most VA applications

are still based on the traditional desktop, mouse, and keyboard setup of WIMP (Windows, Icons, Menus, and a Pointer) interfaces [76]. A few researchers have focused on new possibilities in interaction technologies for VA, which, however, only tested within simple data sets and scenarios. For example, PaperLens [77] uses a handheld lens and a tracked sheet of paper to navigate the 3D virtual information spaces above a tabletop. Interactive Whiteboards [78] leverages hand-drawn input for exploring data through simple charts. Ball and North [71] discuss embodied interactions, such as physical navigation, physically interacting with large-scale visualizations, for improving performance times on analytics tasks through an empirical study.

Therefore, for IVA of multidimensional big data, the challenge of interaction is to develop novel interactions, especially **expressive-oriented** interactions, by taking advantage of new algorithms and devices.

2.1.5 Future Directions for Interactive Visual Analysis of Multidimensional Big Data

To address the scalability challenge for IVA of multidimensional big data, developing scalable visualization techniques is an important research direction. In the field of information visualization, most methods for multidimensional data are focused on relatively small datasets. For example, various studies [79, 80] on PC plots are both limited when the size of the data is scaled up. Therefore, making visualization techniques for multidimensional data scalable for large datasets would be a potential direction for IVA of multidimensional big data.

To facilitate collaboration and information sharing with VA, building web-based frameworks for VA applications is a potential research direction. A web-based framework could break time and space constraints in communication and collaboration. Moreover, it could also facilitate the integration of VA applications with other big data platforms, since most recent big data platforms provide web services for accessing and processing the data stored within them [81]. This will not only address the scalability challenge of visual analytics but also will accelerate the research and development of VA.

2.2 Edge Bundling for Parallel Coordinates Plots

As a standard tool for visualizing multidimensional data, parallel Coordinates (PC) is one of the most widely used techniques for **multidimensional-transformation-2D** VA applications. Brushing and axes-reordering are two typical **exploratory-oriented** interactions of PC-based VA applications.

PC plots render multidimensional data records as polylines across multiple parallel axes, which can be seen as trail-sets. Therefore, for multidimensional big data, PC plots suffer from the same visual clutter and overplotting as large graphs [46]. More importantly, due the limited scalability of PC, the time-consuming process of rendering multidimensional big data as polylines can significantly delay the visual response of the **exploratory-oriented** interactions of PC-based VA applications, which hinders the users from gaining insight into data by exploring the patterns emerging from the visualization of data.

Edge bundling (EB) technique are used to provide a visual simplification of graph visualization by spatially grouping or concentrating graph edges or trails as bundles [82, 47]. Varied scalable EB methods have been proposed to support visual analysis of large graphs that contains a large number of edges or trails [83]. For example, Lhuillier et al. [84] shifted the bundling process from the image space to the spectral (frequency) space to increase its scalability and computational speed. Huang et al. [85] presented a filtering approach to explore a graph at continuous levels of details in real time. Burch et al. [86] transformed the edges to a pixel-based scalar field in a scalable way to address massive overplotting of edges in huge graphs.

By adapting EB techniques to PC plots, a number of EB methods that bundle the data with different clustering algorithms and render the bundled data in different forms of visualizations have been proposed for reducing visual clutter and overplotting in PC plots. Palmas et al. [87] rendered the bundles using polygonal strips to reduce clutter and decrease rendering time. Lima et al. [88] used confluent drawing to render the bundles as merged curves with their statistical information.

Confluent drawing is an unambiguous method to visualize non-planar graphs in a planar way and reduce ambiguity [89, 90]. It enables groups of edges to be merged together and drawn as "tracks" to facilitate tasks where the user needs to follow lines. Fua et al. [91] used a multi-resolution view to visualize aggregation information of hundreds of thousands of data records. Novotny and Hauser [92] proposed an output-oriented approach on the basis of a binned data representation to visualize the trends and outliers in millions of data records. Artero et al. [79] used frequency or density information to filter the data and visually highlight the corresponding polylines. McDonnell and Mueller[93] used spline-based rendering to reveal the distribution of the data. Johansson et al. [94] used high-precision textures to represent the bundles and transfer functions that operate on the high-precision textures to highlight different aspects of the bundle characteristics.

However, due to the limited scalability of the underlying clustering algorithms, existing EB techniques for PC only support limited **exploratory-oriented** interactions based on the pre-computation of the data. For example, based on the pre-computation of the clustering at different levels of detail, the edge-bundled PC in [87] allows the users to choose the bundled visualization at different levels of abstraction. More importantly, existing EB techniques do not support the **expressive-oriented** interactions in PC that require re-clustering of the data, such as changing the bundling results based on users' perception and axes-reordering, which reduces the usability and usefulness of EB techniques for PC-based VA applications, especially for multidimensional big data.

In addition to lightweight EB algorithms, GPU-based methods and big-data-infrastructure-based methods have been proposed to implement scalable bundling of a large number of edges. Zwan et al. [95] presented a fully GPU-based method to bundle large graphs of up to a million edges in real time. Peysakhovich et al. [96] proposed a GPU-based framework to generate bundled graph layouts according to numerical edge attributes such as directions, timestamps or weights. Perrot and Auber [97] used an established big data ecosystem and WebGL to visualize large graphs in a web browser. Sansen et al. [98] combined big data infrastructures and hardware-accelerated rendering to support web-based visual exploration

of multidimensional big data in PC plots.

Table 2.1: Comparison of Our Method with Other Edge-Bundled PC plots.

Method	Clustering	Scalable	Precomputation	Interactions on Clusters	Data Distribution	Inner-Cluster Distribution	Outlier Detection[1]	Non-overlapping Bundles[2]	Tracing Subset with Proportion[3]
Ours	Binning-based	✔	✗	▲	✔	✗	✔	✔	✗
[98]	Not specified	✔	✔	■	✔	✔	✗	✔	✔
[91]	BIRCH [99]	✔	✔	■	✗	✔	✗	✗	✗
[92]	Binning-based	✔	✔	■	✗	✗	✔	✗	✗
[87]	KDE	✔	✔	●	✔	✗	✔	✗	✗
[88]	DBSCAN	✗	✔	✗	✗	✔	✗	✔	✗

[1] The bundles or edges that do not well align with the major trends are detected with the visualization.
[2] The bundles are aligned without overlapping.
[3] A subset of a cluster and its proportion in the cluster are displayed in the corresponding bundles over the axes.
▲ Change the interval and the number of the clusters on each axis with the control points.
■ Brush on the axes to highlight the selected bundles or subsets of bundles.
● Choose the visualizations at different levels of abstraction based on the pre-computation of the clustering.

In the thesis, I have proposed a scalable lightweight EB method for supporting PC-based IVA of multidimensional big data. Table 2.1 shows the comparison of the proposed method with several state-of-the-art edge-bundled PC plots. Most of the methods listed in Table 2.1 are scalable for visualizing multidimensional big data, for example, millions of data records, by reducing the rendering time with bundled visualizations. However, their scalability of bundling the data is limited by the time-consuming clustering process of the data, which can significantly delay the visual response of the **expressive-oriented** interactions. For example, the method [87] needs multiple minutes to pre-compute the clusters of millions of data records. Accordingly, its interactions are limited to brushing on the axes to highlight the selected bundles or subsets of bundles or choosing the visualizations at different levels of abstraction based on the pre-computation of the clustering. In contrast, the proposed method can cluster and visualize 1 million data records with 6 dimensions in about 1 second in web-based visualization without pre-computation. It provides novel interactions that change the cluster-

ing results to support IVA of multidimensional big data. Moreover, it eliminates the overplotting of the bundles, especially in the area near the axes, reveals data distribution, and detects outliers in the data.

2.3 Colorizing Unlabeled Multidimensional Data

Dimension reduction is one of the fundamental techniques for **multidimensional-reduction-2D** VA applications. It projects multidimensional data into a new orthogonal coordinate system with lower dimensionality in which the locations of the data points can preserve their relationships in the original definition space, such as correlation, neighborhood and distance relationships, and classes/clusters in the data set. For example, a variety of DR methods, such as PCA [18], MDS [19], LDA [20], Isomap [22], LLE [23], LE [26], GPLVM [27], UMAP [28] and t-SNE [29], have been widely used for analyzing and visualizing multidimensional data.

For DR algorithms, different configurations of parameters and constraints can lead to very different projections of the same data. For example, as previously shown in Figure 1.4, with the inappropriate perplexity parameter, the t-SNE algorithm projects the data points of the same class into two misleading clusters. Therefore, to reduce misleading information in the projection of data, the users need to adjust parameters and constraints of DR algorithms with several iterations based on their perception and judgments.

For multidimensional big data, the limited scalability is a major challenge of DR algorithms, which can significantly increase the computation time and the time required for tuning parameters and constrains. For DR-based VA application, this can delay the visual response of **expressive-oriented** interactions that require re-computation of the data.

For labeled multidimensional data, colorizing the data by label (different labels and colors for each class) is a common approach to enhance the lower-dimensional projections of the data for distinguishing the data for each class [100, 101, 102]. Moreover, as the example shown in Figure 1.4, colorizing the data by label can

provide a hint for the users to help them tune parameters of DR algorithms. However, automatically colorizing unlabeled multidimensional data to discover classes in the data and help the users tune parameters of DR algorithms has not been sufficiently investigated, especially for unlabeled multidimensional big data.

A few methods have been proposed for mapping unlabeled multidimensional data into colors with different purposes. Ready and Wintz [103] applied PCA to multispectral images and used the first few component images to preserve as much of the data variance as possible. This method can fuse multispectral images into a single color image by mapping the first three principal components to the coordinates in the CIELAB color space. Lawrence et al. [104] used a nonlinear transformation of multidimensional scaling to fuse features in each component image of multispectral images into a lower-dimensional image whose pixel values are consistent with the pixel values of the original multispectral images. By imposing soft constraints on the color of the output pixels, their algorithm maintains a realistic color scheme and exaggerates fine-scale details in the output image. However, these two methods are limited to light measurements in a small number of spectral bands, and offer litter control about the colors assigned.

Cheng et al. [105] introduced a data-driven method to map multidimensional data with geolocation into colors to appreciate isolated and grouped hot spots as well as uneventful areas. They created a circular multivariate color mapping coordinate system by combining RadViz [106, 107] and the HC slice of the HCL (Hue Chroma Luminance) [108] color space at L = 55. The attributes of the dataset are mapped to the boundary of the circular system to determine the attributes' color schemes, and the data points are mapped inside the circular system to determine the color scale of each attribute. Then, the values of each attribute are mapped into a separate image with a color scale. After that, similar to the fusion of multispectral images, the separate images are fused into a single color image by interpolating the colors at the same geolocation in each image with adaptive kernel density estimation [109]. Their method also showed the possibility of mapping multidimensional data without geolocation into colors with the projection of the data, such as PCA or MDS plot. However, this method does not scale well

to large datasets. Mapping the attribute and data points to the circular system to determine the HCL color of the data points is a time-consuming process. Furthermore, it needs three steps, including non-linear and linear transformations, to convert HCL color to RGB color to display the data points. Moreover, limiting the luminance of the HCL color space to 55 reduces the range of colors, which will cause the data points with different values to be mapped to the same location in the circular system.

In this thesis, I have proposed a scalable method, named ColorPCA, for automatically colorizing unlabeled multidimensional big data to discover classes in the data. It provides a fast way to enhance the lower-dimensional projections of the data and help the user discover classes in the data and find suitable parameters of DR algorithms. In contrast to the method [105], ColorPCA directly assigns predefined color schemes to the dimensions of the data. It uses PCA to derive the opacity of the data and then uses ray casting to compute the composite RGBA color of the data. The linear accumulation process of ray casting makes ColorPCA scalable to multidimensional big data. Applying the idea of ray casting with the emission-absorption model [110] to multidimensional data allows ColorPCA to map the data to the entire RGBA color space.

Separating color into chromatic and intensity information has been widely used for processing color image, such as enhancement [111, 112, 113], filtering [114] and retrieval [115]. In this thesis, a chromaticity-preserving color contrast enhancement method is proposed to enhance the color contrast of the data colorized by ColorPCA. It separates color data into chromatic and luminance information with the CIE xyY color space and re-scales the luminance of the color.

ColorBrewer [116] is an online tool designed to help users select appropriate color schemes for their specific mapping needs. In this study, ColorBrewer is used to derive two types of color schemes (sequential and categorical) to investigate the impact of color schemes on colorizing multidimensional data with ColorPCA.

Hagh-Shenas et al. [117] assessed the information carrying capacities of color

blending and color weaving of conveying multidimensional data with color mixing by visualizing a 6-dimensional dataset on choropleth maps. Their results showed that when the component colors are blended into a single mixed color via a linear combination of the individual values in the CIELAB color space, it is arguably difficult for users to decode the mixed color into the individual values via the color maps. Although ColorPCA uses blending to map multidimensional data points to colors, it can also compute the individual opacity and color for each attribute of the data points. The individual color for each attribute is a component color blended into the composite color of the data point. The individual opacity for each attribute reflects the proportion of the component color blended into the composite color.

Chapter 3

Scalable Visualization Methods and Web-based Applications

PC and DR are two of the most widely used techniques for **multidimensional-transformation-2D** and **multidimensional-reduction-2D** VA applications for multidimensional data, respectively. This chapter presents the proposed scalable EB method and ColorPCA to address the scalability and interaction challenges of PC and DR-based VA for multidimensional big data.

3.1 Scalable Lightweight Edge Bundling for Parallel Coordinates

In this thesis, I have proposed a scalable lightweight EB method for supporting PC-based IVA of multidimensional big data. The proposed EB method integrates human judgments into two-dimensional data binning to cluster the data between each two adjacent dimensions through novel interactions, which allows the users to define specific clusters based on their knowledge and judgments. It uses frequency-based representation to align the clustered data as histogram-like bundles, which reduces visual clutter in PC plots and accelerates the rendering process. This section introduces the clustering and mapping process and interactions of the proposed method and the web-based VA application based on the proposed method.

3.1.1 Two-Dimensional Data Binning-based Clustering

Data binning is the process of grouping a number of data values into a smaller number of given intervals (also called "bins"). Multidimensional binning is used to

implement focus + context visualization in PC plots to represent outliers [92]. In the proposed EB method, I use two-dimensional data binning to cluster the data between each two adjacent axes in PC plots based on the following considerations:

- For a single axis, the clusters can be naturally treated as multiple ordered intervals.
- With given intervals on two adjacent axes, two-dimensional data binning is well scalable for big data.
- Human judgments can be used to determine the intervals on each axis.

To cluster the data between each two adjacent axes, firstly, for a m-dimensional data set D, I first evenly create k intervals on each axis, which are computed as:

$$P_{i,j} = \text{Min}(d_i) + (j-1) \times \frac{\text{Max}(d_i) - \text{Min}(d_i)}{k},\ j = 1, ..., k$$

where $\text{Max}(d_i)$ and $\text{Min}(d_i)$ denote the max and min values of the i-th axis, $I_{i,j} = [P_{i,j}, P_{i,j+1}]$ defines the j-th interval on the i-th axis, in which $P_{i,j}$ and $P_{i,j+1}$ are the boundaries of the interval.

For a given interval $I_{i,j}$, a data point d_i belonging to it is defined as:

$$d_i \in I_{i,j} \text{ if } P_{i,j} < d_i \leq P_{i,j+1}$$

Between two adjacent axes, two data points that belong to a pair of intervals form a data item, which is defined as:

$$(d_i, d_{i+1}) \in [I_{i,p}, I_{i+1,q}] \text{ if } d_i \in I_{i,p} \wedge d_{i+1} \in I_{i+1,q}$$

where (d_i, d_{i+1}) denotes a data item between the i-th and $(i+1)$-th axes, $[I_{i,p}, I_{i+1,q}]$ denotes a pair of intervals which is formed by the p-th interval on the i-th axis and the q-th interval on the $(i+1)$-th axis.

Then, each pair of intervals $[I_{i,p}, I_{i+1,q}]$ determines a two-dimensional cluster $C_{i,p,q}$,

which is defined as:

$$C_{i,p,q} = \{(d_i, d_{i+1})\}_{p,q}$$

where $\{(d_i, d_{i+1})\}_{p,q}$ denotes all the data items that belong to $[I_{i,p}, I_{i+1,q}]$.

Figure 3.1a demonstrates the clustering process. In Figure 3.1a, the black points on the axes are the boundary points $P_{i,j}$. Between each two boundary points are the intervals $I_{i,j}$, which are marked by the dashed lines. Between the two axes, each line is a data item (d_1, d_2). The data items in the red and green areas form two clusters $C_{i,p,q}$ separately. For example, the cluster $C_{1,2,1}$ is determined by the pair of intervals $[I_{1,2}, I_{2,1}]$.

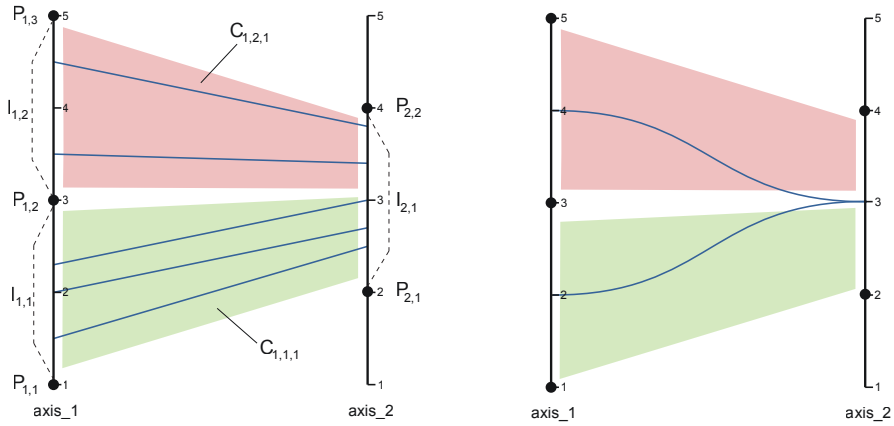
The default value of k is 3, which can be configured by the user (see in Section 3.1.4). The case studies in Section 4.1.2 show that 3 to 5 intervals on each axis are enough for recognizing trends and patterns in the data. Generally, the clusters formed by the evenly distributed intervals on each axis can not fully reveal the patterns in the data. The scalability of the proposed EB method allows the users to adjust the intervals based on their perception and judgments through specifically designed interactions (see in Section 3.1.3).

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

3.1.2 Frequency-based Representation

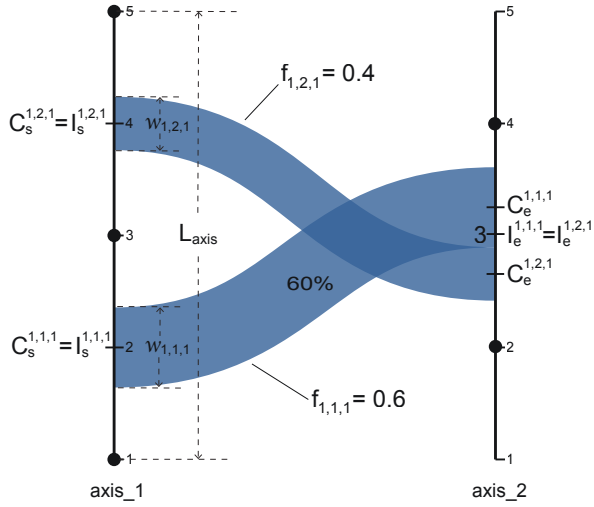
To enable scalable bundled visualization of multidimensional big data and reduce visual clutter, I adopt the EB concept in [88], which merges all the data items belonging to the same cluster into one bundle. To guarantee C^1-continuity across axes, the bundles are represented as Bézier curves. Figure 3.1b shows the bundling result of Figure 3.1a, in which the edges in red and green areas are merged into two bundles and rendered as two Bézier curves separately.

For a data set with m dimensions, the maximum number of bundles N_b is calculated



(a)

(b)



(c)

Figure 3.1: The clustering and mapping process of the proposed bundling method. (a) The process of the two-dimensional data binning-based clustering. (b) The bundling process of the clustered data. (c) The frequency-based representation of the clustering result.

as:

$$N_b = \sum_{i=1}^{m-1} k_i \times k_{(i+1)}, \quad i = 1, ..., m-1$$

where k_i is the number of the intervals on the i-th axis.

For a large multidimensional dataset, N_b is much smaller than the number of data records. Moreover, it is independent of the number of data records. For example, to visualize 1 million data records with 5 dimensions, the proposed EB method draws maximum 100 bundles (5 intervals on each axis) instead of 1 million polylines.

However, comparing Figure 3.1a and Figure 3.1b, merging the data items into the bundles will hide the distribution of the data. To create a more informative visualization, I use frequency-based techniques [84, 79] to render the bundles as histogram-like visual entities between each two adjacent axes.

First, the frequency of each cluster is computed as:

$$f_{i,p,q} = \frac{N(C_{i,p,q})}{\sum_{p=1}^{p} \sum_{q=1}^{q} N(C_{i,p,q})}, \quad i = 1, 2, ...$$

where $f_{i,p,q}$ is the frequency of the corresponding cluster $C_{i,p,q}$, i represents the i-th axis, $N(C_{i,p,q})$ is the number of the data items in the cluster $C_{i,p,q}$. $\sum_{p=1}^{p} \sum_{q=1}^{q} N(C_{i,p,q})$ is the sum of the number of the data items for all the clusters between the i-th and the $(i+1)$-th axes, which is a constant number equals to the size of the data set.

Then, the width $w_{i,p,q}$ of each bundle is determined by:

$$w_{i,p,q} = f_{i,p,q} \times L$$

where L is the width of a bundle whose frequency equals to 1. L can be configured

by the users (see in Section 3.1.4). The default value of L is computed as:

$$L = 0.8 \times \frac{L_{axis}}{max(k_i)}$$

where L_{axis} is the length of the axis and $max(k_i)$ is the maximum number of the intervals on the axes.

In addition, for an interval $I_{i,p}$ that contains two or more bundles $C_{i,p,q}(q = 1, 2, ...)$, all the bundles are aligned around the center of the interval one by one. This makes the bundles occupy the middle area of the intervals and eliminates the overplotting of the bundles in the area near the axes. To align the bundles, the start/end position $C_{s/e}^{i,p,q}$ of each bundle $C_{i,p,q}$ on the corresponding axes are computed as:

$$C_{s/e}^{i,p,q} = I_{s/e}^{i,p,q} - \frac{1}{2}\sum_{q=1}^{q} w_{i,p,q} + w_{i,p,q-1} + \frac{1}{2}w_{i,p,q}$$

where $w_{i,p,0} = 0$, $C_{s/e}^{i,p,q}$ is the start/end position of the corresponding bundle $C_{i,p,q}$, and $I_{s/e}^{i,p,q}$ is the center position of the interval where the bundle starts or ends.

Figure 3.1c shows the frequency-based representation based on Figure 3.1b. In Figure 3.1c, the start positions of the two bundles are the centers of the two intervals on axis_1. The end positions of the two bundles are aligned around the center of the interval on axis_2 without overplotting. The widths of the two bundles are computed by their frequencies. The maximum number of the intervals on the axes is 2. The frequency of the wider bundle is displayed in its center with a mouseover.

Between two adjacent axes, an extremely low frequency of a bundle implies its proportion differs significantly from major trends in the data. In addition, the extremely low frequency can make the bundle invisible or easily overlooked because the width of the bundle is scaled according to its frequency. To address this problem, our method renders the bundles whose frequencies are lower than a user-defined threshold $t\%$ as dashed curves. The threshold indicates that each dashed

curve accounts for up to $t\%$ of the whole data set. The default value of the threshold is 1% and can be configured by the user. The case study in Section 4.1.2 shows the example of detecting outliers in the data with the proposed method.

3.1.3 Interactions

According to Section 3.1.1, the clusters between each two adjacent axes are determined by the intervals on the two axes. To integrate human judgments into the clustering process, I designed three interactions to divide, adjust, and merge the intervals. These interactions allow the users to visually analyze the data by adjusting the clustering and the visualization in real time.

Figure 3.2 shows the interaction of adding a new interval by dividing the origin interval. There is an invisible clickable area around each axis, which is marked by the dashed rectangle in Figure 3.2. Double-clicking in this area will add a new boundary point at the corresponding position on the axis, which will divide the original interval into two smaller intervals. Then, the clusters formed by the two new intervals will be computed and re-rendered. As shown in Figure 3.2, left, double-clicking in the area marked by the dashed red circle adds a new boundary point on the axis. The image on the right shows the result, where the two new intervals are created, and the new clusters are computed and re-rendered.

Dragging the common boundary point of two adjacent intervals along the axis to a new position will adjust their ranges. Then, the clusters formed by with the two new intervals will be computed and re-rendered. Figure 3.3 shows the interaction of adjusting the ranges of the intervals. In Figure 3.3, left, the boundary point marked in red is dragged along axis_1 from the position at 3 to the position at 4. Figure 3.3, right, shows the result which expands the origin interval between 2 and 3, narrows the origin interval between 3 and 5 and re-renders the new bundles.

Double-clicking on the common boundary point of two adjacent intervals will delete the boundary point and merge the two intervals into a bigger interval. Then, the clusters formed by the new interval will be computed and re-rendered. Figure 3.4 shows the interaction of merging two adjacent intervals. In Figure 3.4, left, the

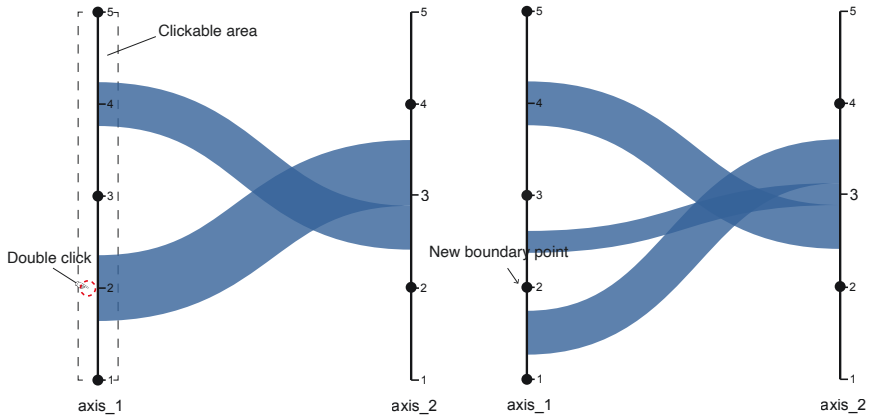


Figure 3.2: The interaction of dividing an existing interval on the axis. Left: before the interaction. Right: after the interaction.

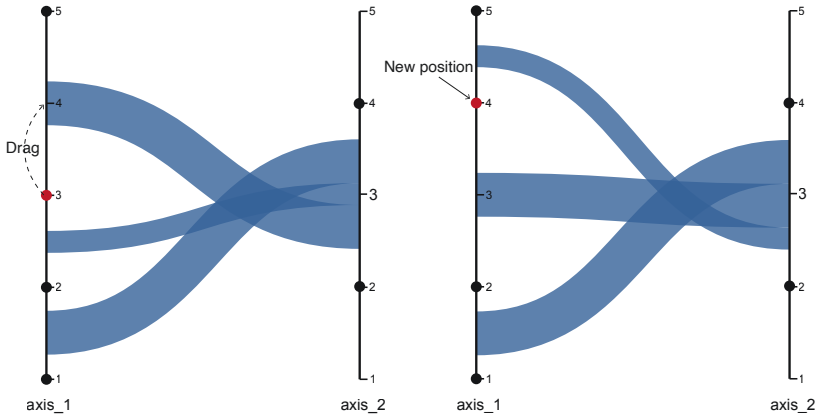


Figure 3.3: The interaction of adjusting two adjacent intervals on the axis. Left: before the interaction. Right: after the interaction.

boundary point at 2 on axis_1 is double-clicked. Figure 3.4, right, shows the result which deletes the boundary point, merges the two adjacent intervals, and re-renders the new bundle.

Besides the interactions that change the clustering results, our method also pro-

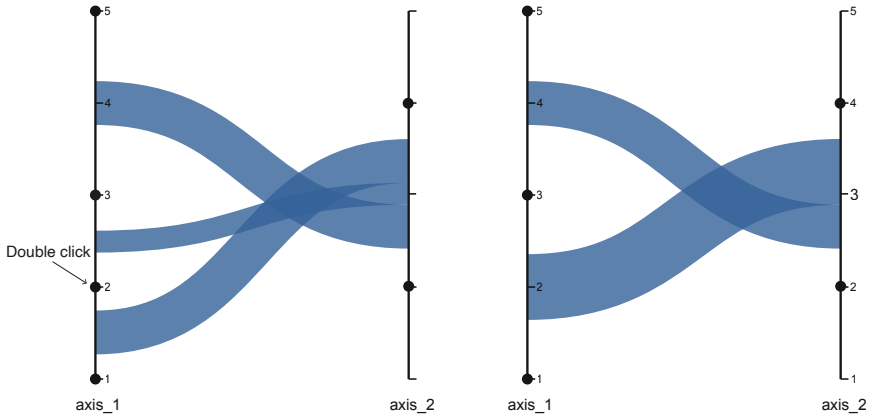


Figure 3.4: The interaction of merging two adjacent intervals on the axis. Left: before the interaction. Right: after the interaction.

vides the following interactions:

- Dragging the axis labels will re-order the axes. The clusters between each two adjacent axes will be computed and re-rendered. Since the number of data points between each two axes is constant, different dimension orders will not affect the scalability of the method. However, different dimension orders will lead to different bundling results. The users can interactively re-order the dimensions to explore different bundling results and find the best one based on their judgments.

- Double-clicking on a bundle between two adjacent axes will select all the data points in this bundle and highlight the bundles that contain the selected data points in red over the axes to distinguish different paths throughout the plot (see in Section 4.1.2).

- A mouseover on a bundle will display its frequency in its center (see the wider bundle in Figure3.1c).

Brushing is a common interaction for PC plots to select and highlight the data

records, such as brushing a subset of the data and re-rendering all the selected data records with highlighting. The proposed EB method does not support brushing because the re-rendering of the brushed data records can delay the visual response of the interaction, especially for millions of data records in lightweight web-based visualization. Instead, the proposed EB method allows the users to create a bundle with the desired intervals on the axes and select all the data points in the bundle by double-clicking on the bundle. Then, it re-renders the bundles that contain the selected data records to highlight the selected data records, which makes the running time of the re-rendering independent of the number of the selected data records. Please see the video in the supplemental material of Paper 3 for an impression of the interactions provided by the proposed EB method.

3.1.4 Parallel Coordinates-based VA Application

Based on the proposed EB method, I have built a lightweight web-based VA application for interactively visualizing and exploring multidimensional big data in PC plots. I implemented the clustering process of the proposed method as a web service in Java to enable the scalable EB process of the data. I used D3.js [118] (a JavaScript library) and SVG (Scalable Vector Graphics) to render the clustered data in web browsers. Without using WebGL and GPU-accelerated rendering, this lightweight web-based application can run across devices and platforms.

Figure 3.5 shows the screenshot of the application. The user can configure the parameters for the clustering and mapping process of the proposed EB method, such as the initial number of the intervals in each dimension (k in Section 3.1.1), the width of the bundle whose frequency is 1 (L in Section 3.1.2), and the threshold for detecting outliers (t% in Section 3.1.2). In addition, the user can choose whether to display the outliers (dashed curves).

With the interactions described in Section 3.1.3, the application allows the user to interactively optimize the clustering and the visualization to visually analyze multidimensional big data in PC plots with their perception and judgments, such as discovering major trends and specific patterns in the data, estimating the correlation between variables, and detecting outliers in the data. The case study in

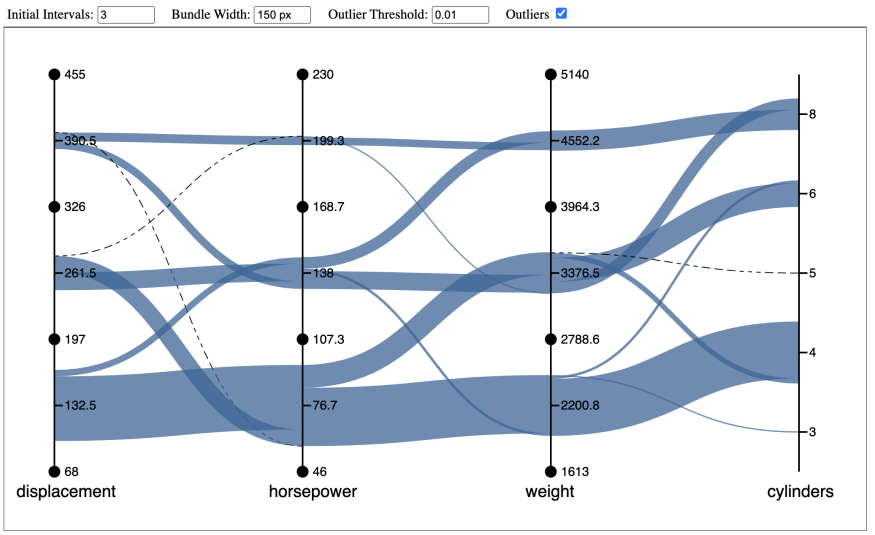


Figure 3.5: The screenshot of the web-based VA application based on the proposed EB method.

Section 4.1.2 shows the example of using the application to visually analyze a large multidimensional dataset.

3.2 Scalable Colorization of Multidimensional Big Data

For DR-based VA applications, the limited scalability of DR algorithms is a major challenge for multidimensional big data. The time-consuming processes of data computation and parameters configuration can significantly delay the visual response of **expressive-oriented** interactions that require re-computation of the data. For labeled data, colorizing lower-dimensional projections of the data by label can provide a hint for exploring the data and configuring parameters of corresponding DR algorithms.

In this thesis, I have proposed a scalable method, named ColorPCA, to address the challenge of automatically colorizing unlabeled multidimensional big data for

discovering classes in the data. It provides a fast way to enhance the lower-dimensional projections of the data and help the user discover classes in the data and find suitable parameters of DR algorithms. Moreover, I have proposed a method to enhance the color contrast without changing the chromatic information of color, which can help the users to distinguish the data points by color. This section introduces ColorPCA, the color contrast enhancement method and the web-based VA application based on ColorPCA.

3.2.1 ColorPCA

Given a set of unlabeled data points S in $\mathbb{R}^m$, $m > 3$, ColorPCA aims at mapping the data points in S into the RGBA color space in $\mathbb{R}^4$ to discover classes in S by colorizing the data points. In this thesis, the 'attribute' refers to a certain attribute value of a data point in S. The 'dimension' refers to the set of all the attribute values at a certain dimension of S.

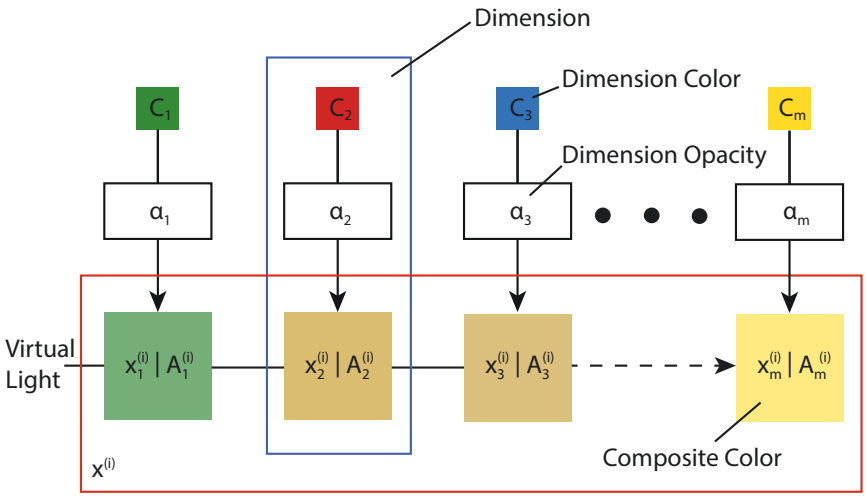


Figure 3.6: The process of mapping the data point $x^{(i)}$ in S into RGBA color space with ColorPCA.

With ColorPCA, the dimensions of S are arranged sequentially for color mapping. Figure3.6 shows the process of mapping the data points $x^{(i)}$ in S into the RGBA

color space. In Figure3.6, the squares highlighted in the red rectangle represent the data point $x^{(i)}$, of which each square represents an attribute $x_j^{(i)}, j = \{1, ..., m\}$ of $x^{(i)}$. The RGBA color of each attribute square is the composite color of the corresponding attribute of $x^{(i)}$, which is accumulated by ray casting from the first attribute $x_1^{(i)}$ to the current attribute $x_j^{(i)}$. The composite color of the last attribute $x_m^{(i)}$ is the final color that the data point $x^{(i)}$ mapped into the RGBA color space.

The composite color and opacity of $x_j^{(i)}$ are computed by the dimension color C_j and the attribute opacity $A_j^{(i)}$, where the set $\{C_j\}, j = \{1, ..., m\}$ is the color scheme assigned to the dimensions of S. In Figure3.6, the smaller squares show the RGB colors of the color scheme, which are assigned to the dimensions of S by the user. The attribute opacity $A_j^{(i)}$ is computed by the value of $x_j^{(i)}$ and the dimension opacity α_j. α_j is derived from PCA. As highlighted in the blue rectangle in Figure3.6, the dimension color C_j and opacity α_j are two constants for the j-th dimension of S. By accumulating the attributes of $x_j^{(i)}, j = \{1, ..., m\}$, $x^{(i)}$ with same or similar attributes will be mapped to same or close locations in the RGBA color space.

Three steps of ColorPCA, including computing the dimension opacity α_j, mapping $x^{(i)}$ into RGBA color space, and color contrast enhancement are described as follows.

Derivation of the Dimension Opacity

Consider S that contains n data points as a $n \times m$ matrix:

$$S = (x^{(1)}, x^{(2)}, ..., x^{(n)})^T$$

where the row vector $x^{(i)} = [x_1^{(i)}, x_2^{(i)}, ..., x_m^{(i)}], i = \{1, ..., n\}$ denotes the i-th data point in S, and the column vector $d^{(j)} = [d_1^{(j)}, d_2^{(j)}, ..., d_n^{(j)}]^T, j = \{1, ..., m\}$ denotes the j-th dimension of S.

PCA is an orthogonal linear transformation that computes the most meaningful basis, a set of linearly uncorrelated principal components, to form a new coordinate system (space T) to express the data. It transforms the data to the new coordinate

system such that the greatest variance by any projection of the data comes to lie on the first principal component, the second greatest variance on the second principal component, and so on.

Let $W = (w^{(1)}, w^{(2)}, ..., w^{(m)})$ be the PCA transformation to S, where each column vector $w^{(i)} = [w_1^{(i)}, w_2^{(i)}, ..., w_m^{(i)}]^T, i = \{1, ..., m\}$ is an eigenvector of the covariance matrix of S.

With W, the data points in S can be transformed into a new m-dimensional space T by:

$$T = S \cdot W = \begin{bmatrix} x^{(1)} \cdot w^{(1)} & \cdots & x^{(1)} \cdot w^{(m)} \\ \vdots & \ddots & \vdots \\ x^{(n)} \cdot w^{(1)} & \cdots & x^{(n)} \cdot w^{(m)} \end{bmatrix}$$

where the row vector $t^{(i)}$ of T is the projection of $x^{(i)}$ in the new m-dimensional space T. The entry $x^{(i)} \cdot w^{(i)}$ of T is the projection of $x^{(i)}$ onto the i-th component of T. Despite the direction of the i-th component, $|w_m^{(i)}|$ reflects the variation of the m-th attribute of $x^{(i)}$ due to the i-th component. The greater value of $|w_m^{(i)}|$ indicates that the m-th attribute of $x^{(i)}$ has greater contribution on the projection of $x^{(i)}$.

According to Figure 3.6, the dimension opacity α_j of $d^{(i)}$ is a constant that reflects the contribution of $d^{(j)}$ on the projection of S into RGBA color space.

To derive α_j from the PCA transformation, let $v^{(j)} = [w_j^{(1)}, w_j^{(2)}, ..., w_j^{(m)}], j = \{1, ..., m\}$ be the row vectors of W, then T can be derived as a linear combination of $d^{(j)}$ as:

$$T = \sum_{j=1}^{m} d^{(j)} v^{(j)}$$

where $d^{(j)}v^{(j)}$ is the outer product of the two vectors. $v^{(j)}$ is the row vector composed of the j-th element of all the eigenvectors $w^{(i)}, i = \{1, ..., m\}$, which means that the value of $|v^{(j)}|$ reflects the contribution of $d^{(j)}$ on the projection of S to

the j-th component of T.

Therefore, analogy with the contribution of $d^{(j)}$ on the projection of S to the space T, the dimension opacity α_j of $d^{(i)}$ is computed as:

$$n_j = \left\| v^{(j)} \right\|_1, \quad i = \{1, ..., m\}$$
$$\alpha_j = 0.8 \times \frac{n_j - n_{min}}{n_{max} - n_{min}} + 0.1$$

where n_j is the L^1 norm of $v^{(j)}$, n_{max} and n_{min} are the maximum and minimum values of all n_j.

According to the emission-absorption model [110, 119] with ray casting, if α_j is equal to 1, $d^{(j)}$ is completely opaque, which causes all the colors of the dimensions behind $d^{(j)}$ cannot be accumulated into the composite color. If α_j is equal to 0, $d^{(j)}$ is completely transparent, which causes the color of the j-th dimension cannot be accumulated into the composite color. Therefore, to accumulate the colors of all the attributes of $x^{(i)}$ into its final composite color, the dimension opacity α_j is normalized between 0.1 and 0.9.

PCA normally orders $w^{(i)}$ in W in descending order of the corresponding eigenvalues and uses the first two or three $w^{(i)}$ to reduce the dimensionality of the data and plot the data in a 2 or 3-dimensional space. The different order of $w^{(i)}$ will result in various PCA plots. In contrary to PCA, ColorPCA uses all the information captured in W to compute α_j for each $d^{(j)}$.

Color Mapping

As shown in Figure3.6, to map $x^{(i)}$ into the RGBA color space, the color scheme $\{C_j\}, j = \{1, ..., m\}$ is assigned to the dimensions of S. The idea of ray casting is adapted to accumulate the final composite color and opacity of $x^{(i)}$.

For a dimension of S, the attribute values also contribute differently to the final composite colors and opacity, where the attributes with higher values can absorb more light and contribute more to the final composite colors. Therefore, the attribute values $x_j^{(i)}, i = \{1, ..., n\}$ of the j-th dimension are mapped to the fraction

$F_j^{(i)}$ to adjust the dimension opacity α_j for each attribute value as:

$$F_j^{(i)} = 0.8 \times \frac{x_j^{(i)} - d_{min}^{(j)}}{d_{max}^{(j)} - d_{min}^{(j)}} + 0.1$$

where $F_j^{(i)}$ is between 0.1 and 0.9, and $d_{max}^{(j)}$ and $d_{min}^{(j)}$ are the maximum and minimum values of the j-th dimension.

Then, the attribute opacity $A_j^{(i)}$ for each attribute $x_j^{(i)}$ is computed as:

$$A_j^{(i)} = \alpha_j \times F_j^{(i)}$$

According to the emission-absorption model with ray casting, later dimensions absorb less light, which means they contribute less to the composite color of the data. Therefore, to maximize the data features that can be mapped to the color, the dimensions of S are arranged sequentially in descending order of their standard deviations. Furthermore, this allows ColorPCA to colorize the data with only the first few dimensions to discover classes in the data.

Finally, the front-to-end composite color and opacity for each single attribute $x_j^{(i)}$ of $x^{(i)}$ are accumulated as:

$$C_j^{(i)\Delta} = C_{j-1}^{(i)\Delta} + (1 - A_{j-1}^{(i)\Delta}) \times A_j^{(i)} \times C_j,\ j = \{2, ..., m\}$$
$$A_j^{(i)\Delta} = A_{j-1}^{(i)\Delta} + (1 - A_{j-1}^{(i)\Delta}) \times A_j^{(i)},\ j = \{2, ..., m\}$$

where $C_j^{(i)\Delta}$ and $A_j^{(i)\Delta}$ are the composite color and opacity of j-th attribute of $x^{(i)}$, C_j is dimension color of $d^{(j)}$, and $C_j^{(i)\Delta}$ and C_j are the value of R, G or B of the RGB color. For the first attribute $x_1^{(i)}$ of $x^{(i)}$, $C_1^{(i)\Delta}$ is equal to $A_1^{(i)} \times C_1$, and $A_1^{(i)\Delta}$ is equal to $A_1^{(i)}$.

With the idea of ray casting, the features (attributes) of $x_j^{(i)}, j = \{1, ..., m\}$ are accumulated into $(C_m^{(i)\Delta}, A_m^{(i)\Delta})$ to map $x^{(i)}$ into the RGBA color space, where $C_m^{(i)\Delta}$ and $A_m^{(i)\Delta}$ are the composite color and opacity of the last attribute $x_m^{(i)}$ of $x^{(i)}$.

3.2.2 Chromaticity-Preserving Color Contrast Enhancement

To further help users to distinguish the data by color, the color $C_m^{(i)\Delta}$ of each data point $x^{(i)}$ is separated to chromatic and luminance information with CIE xyY color space to enhance the color contrast of the data without changing the chromatic information.

The concept of CIE xyY color space can be divided to two parts: (x,y) and Y. x and y are the two derived independent parameters, which construct an orthogonal chromaticity coordinate system [120]. Y is the measure of the luminance of the color. Figure3.7 shows the gamut of the standard RGB color space on the xy chromaticity diagram. In Figure3.7, the area in the dashed triangle is the standard RGB color space. The outer curved boundary is the spectral locus. $x^{(i)}$ can be projected to the triangle area in Figure3.7 by the xy chromaticity information of its composite color $C_m^{(i)\Delta}$.

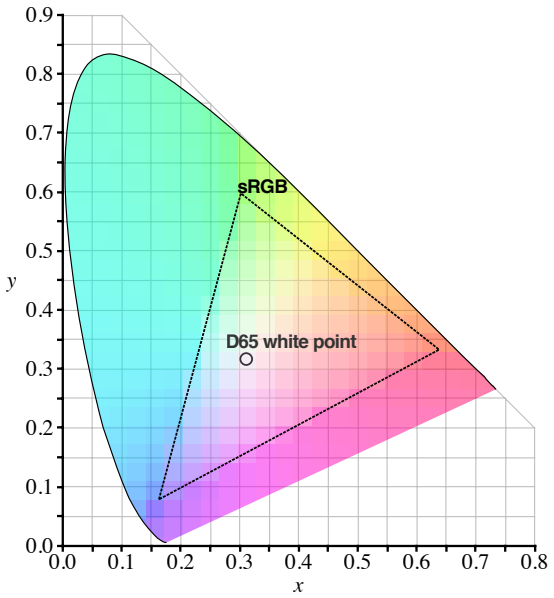


Figure 3.7: The gamut of the standard RGB color space (the triangle) on the xy chromaticity diagram.

In addition to the chromaticity and luminance, the opacity is an independent visual channel to encode a color. As shown in Figure3.8, changing the luminance and opacity of the colors with the same chromatic information can enhance their contrast without changing their chromatic information. Therefore, re-scaling the luminance of $C_m^{(i)\Delta}$ and $A_m^{(i)\Delta}$ are used to enhance the color contrast of $x^{(i)}$.

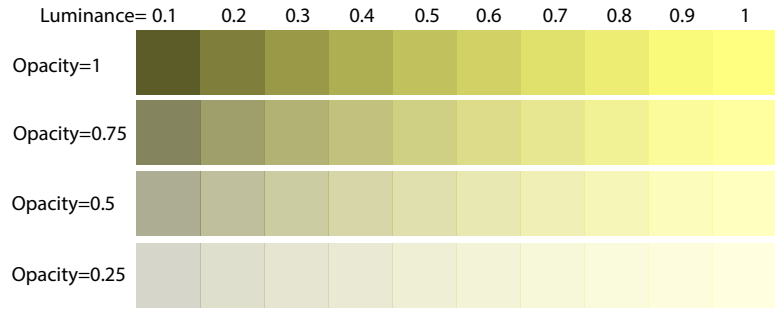


Figure 3.8: The colors with the same xy chromaticity $(0.39, 0.45)$, whose luminance increase from 0.1 to 1 and opacity decrease from 1 to 0.25. The colors in each column have the same luminance. The colors in each row have the same opacity.

To scale the luminance of $C_m^{(i)\Delta}$, first, $C_m^{(i)\Delta}$ is converted to the CIE xyY color by [121]:

$$R_l, G_l, B_l = \gamma^{-1}(u) = \begin{cases} \frac{u}{12.92} & u \leq 0.04045 \\ \left(\frac{u+0.055}{1.055}\right)^{2.4} & otherwise \end{cases}$$

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \cdot \begin{bmatrix} R_l \\ G_l \\ B_l \end{bmatrix}$$

$$x_i = \frac{X_i}{X_i + Y_i + Z_i}$$

$$y_i = \frac{Y_i}{X_i + Y_i + Z_i}$$

where u is R, G or B value of $C_m^{(i)\Delta}$, $\gamma^{-1}(u)$ is the gamma expansion of a standard

RGB color. (R_l, G_l, B_l) are the gamma-expanded RGB values of $C_m^{(i)\Delta}$, which is also called linear RGB. (x_i, y_i, Y_i) is the CIE xyY color of $x^{(i)}$.

Then, the luminance of $C_m^{(i)\Delta}$ is scaled to $[a, b]$ as:

$$
\begin{aligned}
Y_i' &= (b-a) \times \frac{Y_i - Y_{min}}{Y_{max} - Y_{min}} + a \\
X_i' &= x_i \times Y_i' / y_i \\
Z_i' &= (1 - x_i - y_i) \times Y_i' / y_i \\
\begin{bmatrix} R_l' \\ G_l' \\ B_l' \end{bmatrix} &= \begin{bmatrix} 3.2406 & -1.5372 & 0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \cdot \begin{bmatrix} X_i' \\ Y_i' \\ Z_i' \end{bmatrix} \\
R_i', G_i', B_i' = \gamma(u) &= \begin{cases} 12.92u & u \leq 0.0031308 \\ 1.055u^{\frac{1}{2.4}} - 0.055 & otherwise \end{cases}
\end{aligned}
$$

where $0 \leq a \leq b \leq 1$, Y_{max} and Y_{min} are the maximum and minimum luminance of all $C_m^{(i)\Delta}, i = \{1, ..., n\}$, Y_i' is the new luminance of $C_m^{(i)\Delta}$, (R_l', G_l', B_l') is the linear RGB color converted from (x_i, y_i, Y_i'), u is R_l', G_l' or B_l', $\gamma(u)$ is the gamma compression of a linear RGB color, and $C_m^{(i)\Delta\prime} = (R_i', G_i', B_i')$ is the contrast-enhanced RGB color of $x^{(i)}$. The default value of $[a, b]$ is $[0.01, 0.99]$.

The opacity $A_m^{(i)\Delta}$ is scaled to $[c, d]$ as:

$$
A_m^{(i)\Delta\prime} = (d-c) \times \frac{A_m^{(i)\Delta} - A_{m_{min}}^{\Delta}}{A_{m_{max}}^{\Delta} - A_{m_{min}}^{\Delta}} + c
$$

where $0 \leq c \leq d \leq 1$, $A_{m_{max}}^{\Delta}$ and $A_{m_{min}}^{\Delta}$ are the maximum and minimum opacity of all $x^{(i)}, i = \{1, ..., n\}$, $A_m^{(i)\Delta\prime}$ is the contrast-enhanced composite opacity of $x^{(i)}$. The default value of $[c, d]$ is $[0.5, 0.99]$.

Because $(C_m^{(i)\Delta\prime}, A_m^{(i)\Delta\prime})$ and $(C_m^{(i)\Delta}, A_m^{(i)\Delta})$ have the same chromaticity, this color contrast enhancement method does not change the position of $x^{(i)}$ in the xy chro-

maticity diagram. It allows the users to enhance the color contrast of $x^{(i)}$ by scaling up or narrowing down the range of luminance or opacity based on their perception of color. Although it preserves the chromaticity of $C_m^{(i)\Delta}$, $C_m^{(i)\Delta\prime}$ converted from (x_i, y_i, Y_i') may fall outside the standard RGB space. To guarantee $C_m^{(i)\Delta\prime}$ is in the standard RGB space, $C_m^{(i)\Delta\prime}$ is limited to the tristimulus values between 0 and 255 by simple clipping [121]. The effect of this color contrast enhancement method is shown and discussed with the case study in Section 4.2.2.

3.2.3 Dimensionality Reduction with ColorPCA

In addition to colorizing unlabeled multidimensional big data, this section describe the possibility of using ColorPCA to reduce the dimensionality of the data. According to the color converting process described in Section 3.2.2, the color $(x_i, y_i, Y_i, A_m^{(i)\Delta})$ and $(C_m^{(i)\Delta}, A_m^{(i)\Delta})$ represent the same color of $x^{(i)}$. By converting the RGBA color of $x^{(i)}$ to the xyY + Opacity color, ColorPCA reduces the dimensionality of S to four by mapping $x^{(i)}$ into the 4-dimensional xyY+A color space (x, y, Y, A), where A is the accumulation result of the dimension opacity of $x^{(i)}$. According to the concept of CIE xyY color space, x and y construct an orthogonal chromaticity coordinate system. According to the light emission-absorption model with ray casting, the luminance Y of the composite color of each $x^{(i)}$ is determined by the accumulation of the dimension opacity of $x^{(i)}$. Therefore, the 4-dimensional xyY+A color space can be divided to two 2-dimensional coordinate systems: the chromaticity coordinate system (x,y) and the luminance-opacity coordinate system (Y, A). By colorizing $x^{(i)}$ with a xyY + Opacity color $(x_i, y_i, Y_i, A_m^{(i)\Delta})$, ColorPCA can project $x^{(i)}$ to the position (x_i, y_i) in the (x,y) coordinate system and the position $(Y_i, A_m^{(i)\Delta})$ in the (Y, A) coordinate system. Because ColorPCA colorize $x^{(i)}$ based on its attribute values, it can use the locations of $x^{(i)}$ in the two coordinate system to reveal the structure of S.

The example of reducing dimensionality of multidimensional data and visualizing the data in the two coordinate system are demonstrated in Figure 3.10.

3.2.4 Color Schemes of the Dimensions

By adapting ray casting, the color $C_m^{(i)\Delta}$ of the data point $x^{(i)}$ is determined by the color scheme assigned to the dimensions of S and the virtual light that passes through the attributes of $x^{(i)}$. For a given data point $x^{(i)}$ in S, the virtual light that passes through its attributes is determined and immutable because the opacity of each attribute of $x^{(i)}$ is a constant. However, different color schemes will impact the colorization of the data. The impact of two types of color schemes on colorizing multidimensional data with ColorPCA is discussed in this section. These two types of color schemes are:

- **Sequential color schemes** logically progress the colors from light to dark with sequential lightness steps, with light colors for low data values to dark colors for high data values [122, 123]. As shown in Figure 3.9, **CS1** is a discrete sequential color scheme.

- **Categorical color schemes** use the colors with different hues and similar lightness or saturation to show differences between categories and are popular in a wide range of mapping and data visualization contexts, such as categorical maps [124, 125]. In Figure 3.9, **CS2** is a categorical color scheme derived from ColorBrewer.

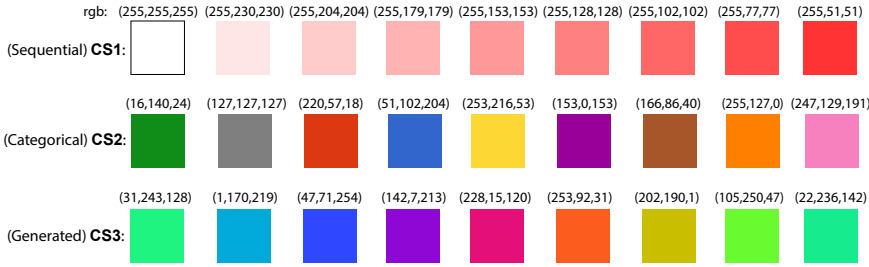


Figure 3.9: Three color schemes for ColorPCA.

Because the color mapping process of ColorPCA is a linear accumulation process, larger differences between the colors in a color scheme can encode more information in the data into the composite color of the data. The colors in sequential color

schemes are controlled by lightness steps, which generally have smaller differences between the colors than categorical color schemes. Therefore, categorical color schemes can achieve a better result than sequential color schemes on colorizing the data with ColorPCA to discover classes in the data.

In addition to deriving color schemes by tools, such as ColorBrewer, color schemes can be automatically generated by functions whose codomain falls into $[0, 255]$. For example, in Figure 3.9, **CS3** is a categorical color scheme automatically generated by:

$$
\begin{aligned}
r(x) &= \lfloor \sin(0.8 * (x-1) + 4) \times 127 + 128 \rfloor \\
g(x) &= \lfloor \sin(0.8 * (x-1) + 2) \times 127 + 128 \rfloor \\
b(x) &= \lfloor \sin(0.8 * (x-1)) \times 127 + 128 \rfloor
\end{aligned}
$$

where $x = [1, 2, \dots]$, $r(x), g(x)$ and $b(x)$ are the R, G, and B values of the x-th color of the color scheme.

For an automatically generated color scheme, the differences between the colors can be maximized by maximizing the variance of the colors:

$$\sigma_R^2 + \sigma_G^2 + \sigma_B^2$$

where, σ_R^2, σ_G^2 and σ_B^2 are the variance of the R, G, B values of the colors, respectively.

Furthermore, colors with small RGB values should be avoided in the color schemes for ColorPCA because smaller RGB values can reduce the differences between the attributes that can be accumulated into the composite colors. For example, with the zero RGB values (black), all the attributes of the corresponding dimension can only accumulate black into the composite color.

The impact of the three color schemes in Figure 3.9 on colorizing unlabeled multidimensional data with ColorPCA is compared with the case study in Section 4.2.2.

3.2.5 ColorPCA-based VA application

Based on ColorPCA, I have built a web-based VA application for IVA of unlabeled multidimensional big data. The colorization and projection processes of ColorPCA are implemented as a web service with Java. The results are visualized as SVG (Scalable Vector Graphics) figures in web browsers, which are implemented with D3.js JavaScript library [118].

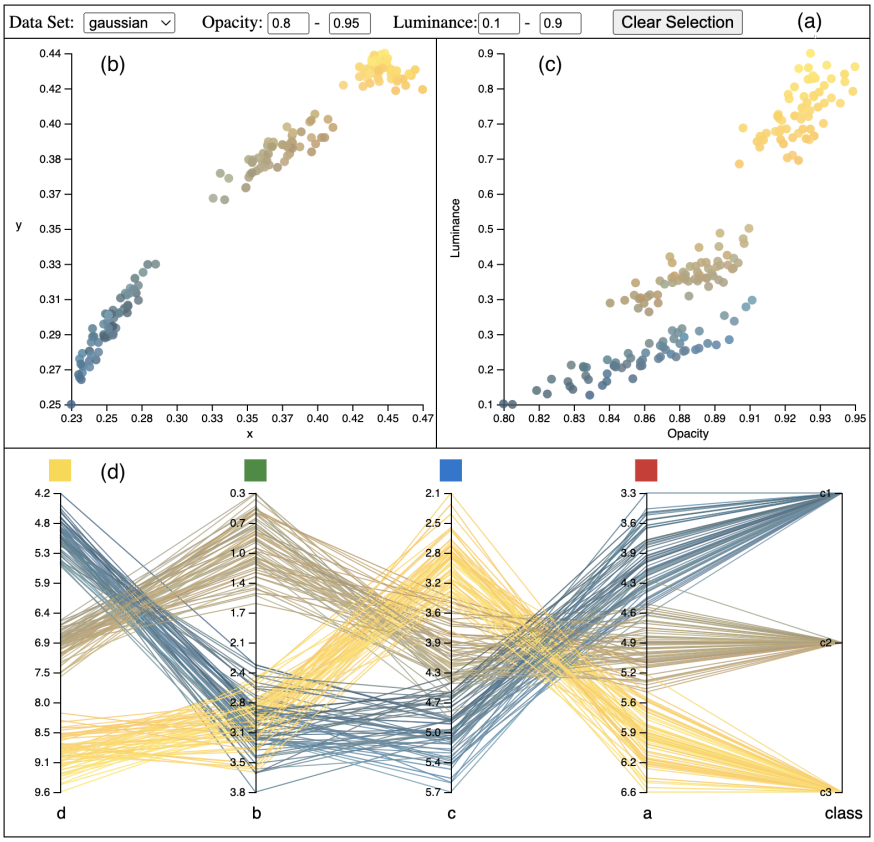


Figure 3.10: The screenshot of the system. (a) control panel. (b) xy plot. (c) luminance-opacity plot. (d) color-enhanced PC plot.

Interface of the Application

Figure 3.10 shows the interface of the system. Along with (a) control panel, the system contains three projection plots: (b) xy plot, (c) luminance-opacity plot, and (d) color-enhanced PC plot. The three plots are coordinated by color, in which the data points (lines) are colorized by ColorPCA. The control panel (a) includes the four controls for the maximum and minimum values of the luminance and opacity of the colors. The color contrast in the three plots can be enhanced by changing the ranges of the luminance and opacity.

The data shown in Figure 3.10 is an artificial dataset with four dimensions. The data set contains three Gaussian clusters (classes) which are randomly generated with different means and the same deviation for each dimension. To reveal the structure of the dataset using the locations of the data points in the color space (x, y, Y, A), ColorPCA colorizes the data points of the same cluster/class with nearly the same color and colorizes the data points of different clusters/classes with distinguishable colors. In Figure 3.10, (b) and (c) show the xy and luminance-opacity plots of the data, in which the three clusters in the dataset are projected to three separate areas and colorized with yellow, tan, and slate gray.

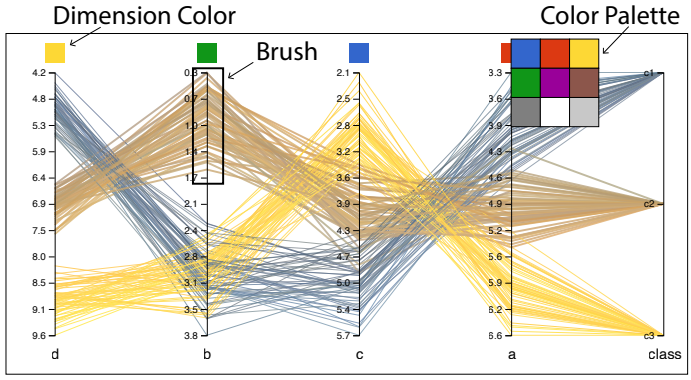
Figure 3.10 (d) shows the color-enhanced PC plot of the data, in which the square on the top of each axis shows the color assigned to the corresponding dimension. In Figure 3.10 (d), the lines are colorized with ColorPCA into the three colors, which forms three bundles by color to reveal the trend of each cluster in the data.

User Interaction

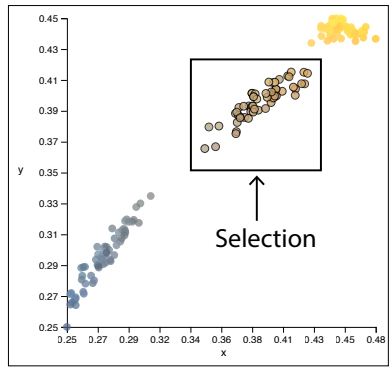
With ColorPCA, different orders and color schemes of the dimensions of a multidimensional dataset will lead to different colorization and projection of the data. To explore the data using different color schemes, the system provides the interaction for changing the color schemes. As shown in Figure 3.11a, clicking on the square on the top of each axis will display the color palette of nine colors. The user can select a new color and assign it to the corresponding dimension by clicking on the color in the palette. The results of different color schemes are discussed in Section 3.2.4. Although the dimensions of the data are pre-ordered with ColorPCA by

default, the system allows the user to re-order the dimensions to explore the data by dragging the axis labels.

In addition, the system provides two interactions for selecting and highlighting the data points in the plots: brushing the data points in a range at an axis in the PC plot and selecting the data points in an area in the xy or luminance-opacity plots. In Figure 3.11a, the rectangle on axis b shows the interaction of brushing the data points at the axis, where the rectangle indicates the range of the brushing. In Figure 3.11b, the rectangle shows the interaction of selecting the data points in an area of the xy plot, where the rectangle indicates the selected area. The rectangles in Figure 3.11 are created by clicking and dragging the mouse cursor. Because the three projection plots are coordinated by color, brushing/selecting in one plot will highlight the selected data points in the three plots. As shown in Figure 3.11, the selected data points (colorized in tan) are highlighted with strokes in the xy plot and highlighted as the thicker lines in the PC plot.



(a)



(b)

Figure 3.11: The interactions of the system. (a) The interactions of changing the color schemes, and brushing the data points in the PC plot. (b) The interaction of selecting the data points in the xy plot.

Chapter 4

Evaluation

This chapter presents the evaluation results of the proposed EB method (Paper 2 and 3) and ColorPCA (Paper 4). Section 4.1 presents the scalability analysis, case studies, and user study of the proposed EB method. Section 4.2 presents the scalability analysis, case studies, user study of ColorPCA.

4.1 Scalable Lightweight Edge Bundling for Parallel Coordinates

4.1.1 Scalability Analysis

The proposed EB method contains the clustering and the rendering process. To examine its scalability, I performed run-time analysis of the two processes with several large datasets. The datasets are synthesized based on the office data set [126]. The machine used in the tests has an Intel Core i5 Processor of 3.1 GHz and 8 Gigabytes of memory. Datasets were fully stored in the memory to avoid hard disk access.

According to Section 3.1.1, the clustering algorithm has time complexity $\mathcal{O}(mnc)$, where m is the number of data records in the data set, n is the number of dimensions, and c is the number of the intervals on each axis. Figure 4.1 shows the approximate lines for the running times of the clustering process, in seconds, for the data sets with different values of m, n, and c. According to Figure 4.1, with the different values of n and c, the running time of the clustering process increases linearly with the amount of data records, which makes the EB method

well scalable for multidimensional big data. For example, it can cluster 1 million data records with 6 dimensions in 1.29 seconds.

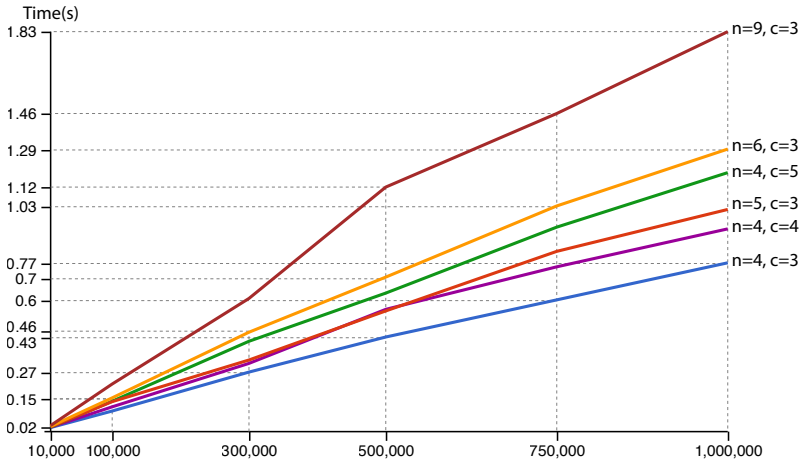


Figure 4.1: Running times (in seconds) of the clustering process of the proposed EB method for different large multidimensional datasets.

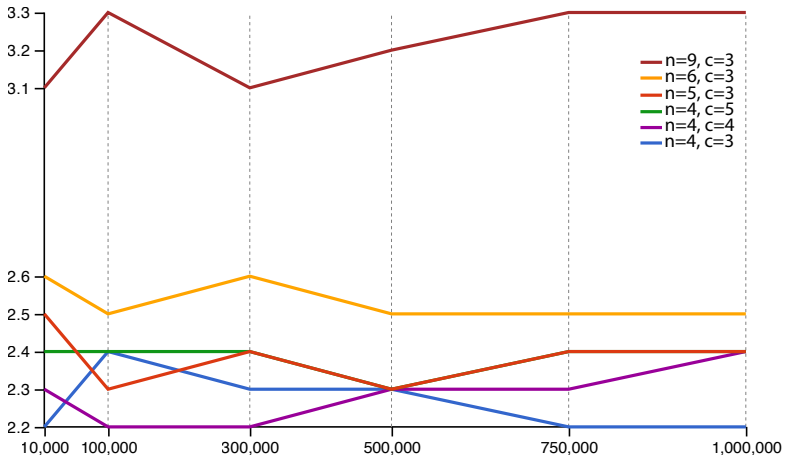


Figure 4.2: Running times (in milliseconds) of the rendering process of the proposed EB method for different large multidimensional datasets.

The rendering process is implemented in JavaScript to visualize the data with SVG

in a web browser without hardware-accelerated rendering. By only rendering the bundled edges, the rendering time of the EB method is independent of the number of data records. Figure 4.2 shows the running times of the rendering process, in milliseconds, for the data sets with different values of m, n, and c. According to Figure 4.2, with the different values of n and c, as the amount of data records increases, the running times of the rendering process always fluctuates between 2.2 to 2.6 milliseconds. As a comparison, with the same implementation, classic PCP takes 31 seconds to render 5×10^5 data records with 6 dimensions, and more than 1 minute to render 1 million data records with 6 dimensions. Instead of rendering all the edges, the proposed EB method is $24,000$ times faster than classic PCP by rendering much fewer bundled edges.

4.1.2 Case Studies

To assess the effectiveness of the proposed EB method, I conducted two case studies to compare it with classic PC plots and two state-of-the-art edge-bundled PC plots [87, 88].

Comparison with Classic PC plots

Figure 4.3 shows the visualizations of the office dataset [126] with classic PC plot. The data set contains sensing data from an office room with five dimensions, including temperature, humidity, light, CO_2 and occupancy, and 20,560 data records. In Figure 4.3, although the edges are rendered with the opacity of 0.1 to reduce visual clutter and overplotting, the areas between the axes are still almost entirely covered by the lines, which hinders the user to gain insight into the data. For example, it is difficult for the user to estimate the distribution of the data between the axes of *temperature* and *light*.

Figure 4.4 shows the initial visualization of the office dataset with the proposed EB method. With the initial settings, each axis is evenly divided into three intervals. For detecting outliers, the threshold is set to 0.9‰, and outliers are rendered as dashed curves. Comparing to Figure 4.3, the proposed EB method creates a clutter-reduced and more informative visualization in the edge-bundled PC plot, which shows the distribution of the data, reveals the correlation of the variables,

and detects outliers in the data. For example, the two wider bundles between the axes of *temperature* and *light* reveals a positive correlation between them.

The proposed EB method allows the user to continuously adjust the initial visualization using their knowledge and judgments through the interactions. Figure 4.5

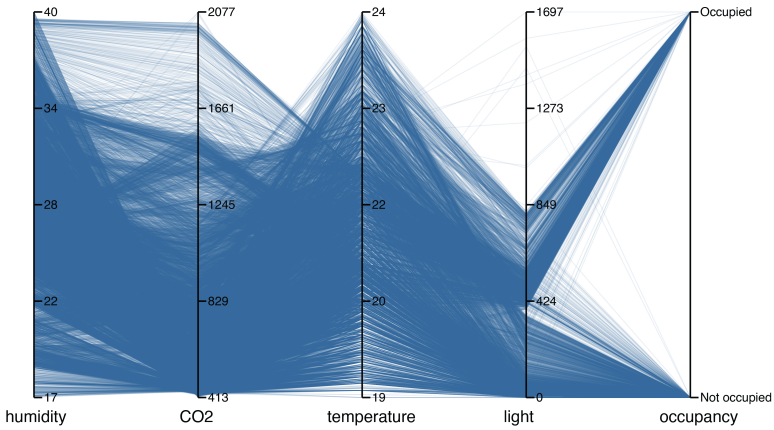


Figure 4.3: The visualizations of the office dataset with classic PC plot.

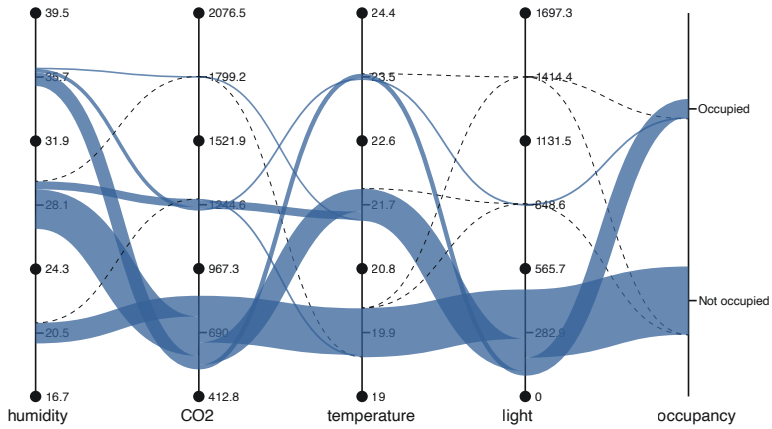


Figure 4.4: The initial visualization of the office dataset with the proposed EB method.

shows the user-adjusted visualization based on Figure 4.4, which is adjusted with the aim to discover the occupancy patterns of the room in the data. To discover patterns from the data, the user may adjust the visualization with the following strategies:

- Defining the clusters with specific intervals to reveal the distribution and patterns of the data with the desired intervals on the axes.

- Merging the data points in low-frequency bundles into adjacent high-frequency bundles to reduce the number of bundles by adjusting or merging adjacent intervals. For example, most data points in the bundles within the interval of 22.6 - 24.4 on the axis of *temperature* in Figure4.4 are merged into the bundles within the interval of 20.7 - 23.2 on the axis of *temperature* in Figure4.5. In Figure4.5, only one bundle (except for the outliers) with a low frequency remain within the interval of 23.2 - 24 on the axis of *temperature*.

- Separating the data points in high-frequency bundles into several thinner bundles to reveal the distribution and patterns of the data points by adjusting or splitting intervals. For example, the data points in the bundles within the interval of 0 - 565.7 on the axis of *light* in Figure4.4 are separated into the bundles within the intervals of 0 - 406.3 and 406.3 - 745.8 on the axis of *light* in Figure4.5. This reveals the relationship between the value of *light* and the occupancy patterns of the room.

Figure 4.6 shows the same clustering result in Figure 4.5, which highlights the major trends in the data by hiding the outliers (dashed curves).

Candanedo and Feldheim [126] tested linear discriminant analysis, classification and regression trees, and random forest on the office dataset to detect the occupancy of the room. Several patterns in the data that are discovered by the algorithmic methods in [126] are also discovered based on Figure 4.5 and Figure 4.6, which are listed as follows:

- **Pattern 1.** Using only one predictor (*light*) is able to estimate the occupancy

with an accuracy over 95%. The threshold for detecting outliers is set to 0.9‰. Therefore, in Figure 4.5, the outliers, four dashed curves, between the axes of *light* and *occupancy* accounts for up to 3.6‰ of the data. Without considering the outliers, 406.3 Lux is the threshold for using *light* to identify

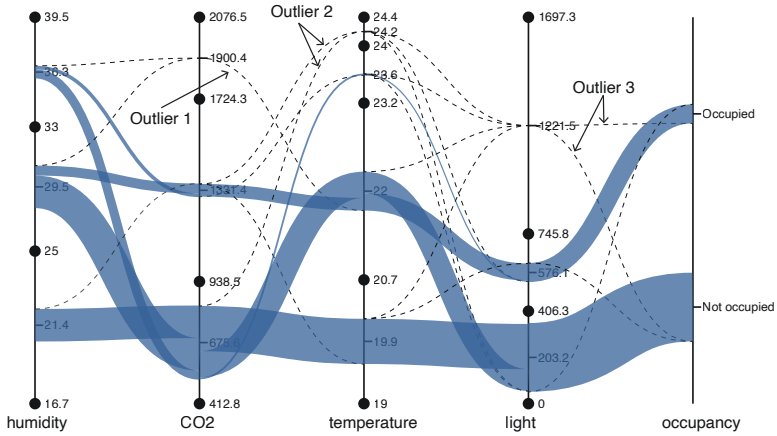


Figure 4.5: The visualization of the office dataset with the proposed EB method and the user-adjusted clusters.

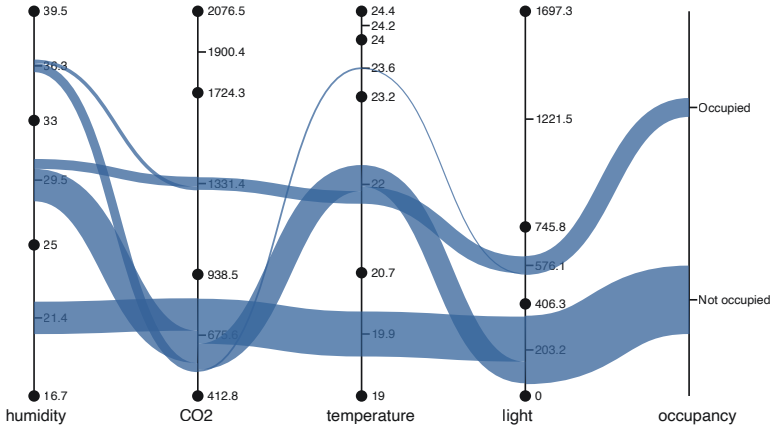


Figure 4.6: The visualization of the office dataset with the proposed EB method, which hides the outliers to highlight the major trends in the data.

whether the room is occupied.

- **Pattern 2.** When the temperature is lower than 20.7°, the room is considered unoccupied.

- **Pattern 3.** In Figure 4.6 (d), the bundles over the axes of CO_2, *temperature* and *light* indicates the data of these three axes are moderately correlated.

- **Pattern 4.** The data of *humidity* has a low correlation with CO_2, *temperature* and *light*. This implies that using different combinations of features can have an impact on the occupancy estimation.

In addition, with the visualization in Figure 4.5, the outliers in the data are detected, which are listed as follows:

- **Outlier 1.** The data records with CO_2 higher than 1724.3 are detected as outliers, which accounts for up to 0.9‰ (1 dashed curves) of the data.

- **Outlier 2.** The data records with *temperature* higher than 24 are detected as the outliers, which accounts for up to 1.8‰ (2 dashed curves) of the data.

- **Outlier 3.** The data records with *light* higher than 745.8 are detected as the outliers, which accounts for up to 1.8‰ (2 dashed curves) of the data.

By comparing the proposed EB method with classic PC plots, the case study shows that the proposed EB method not only reduces visual clutter and overplotting in PC plots, but also reveals the distribution, patterns and correlations of the data, and detects outliers in the data. With the interactions described in Section 3.1.3, the proposed EB method allows the user to visually analyze multidimensional big data in edge-bundled PC plots using their knowledge and judgments.

Comparison with Two Edge-Bundled PCP

Figure 4.7 (a) shows the visualization of the cars dataset with the edge-bundled PC plot of Palmas et al. [87]. The bundling method in [87] uses Gaussian kernel density estimation to create several 3-tuple-defined clusters on each axis independently.

The clusters are represented as the intervals on the axes. Within each interval, the clustered data is rendered as several polygonal strips between the adjacent axes. As highlighted in the rectangles in Figure 4.7, all the polygonal strips belonging to the same cluster have the same starting position, which makes the strips overplotted in the area near the axes. In addition, the strips are rendered in grayscale to reflect the number of data points in the strips. This leads to two shortcomings: 1) in some circumstances, the grayscale cannot accurately reflect the difference in the number of data points in the strips, 2) the user may ignore the strips that contain a small number of data points. For example, in Figure 4.7, it is difficult for the user to distinguish the difference in the number of data points in the red and black strips. In addition, the strips marked by the circles are likely to be ignored by the user.

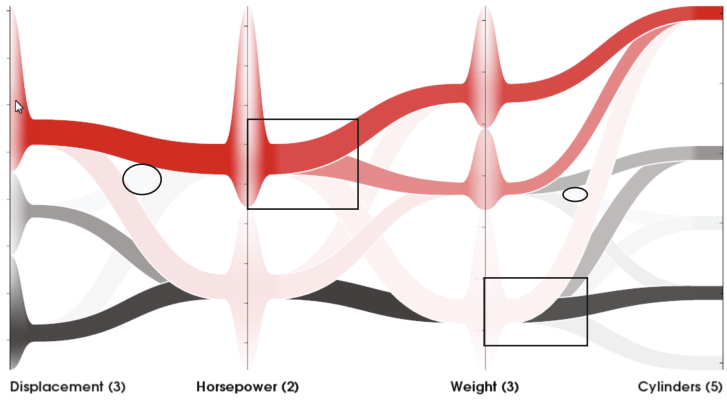


Figure 4.7: The visualization of the cars dataset with the edge-bundled PC plot of Palmas et al [87].

Figure 4.8 shows the cars dataset visualized using the proposed EB method with the similar clustering result in Figure 4.7. As shown in Figure 4.8, double-clicking on the bundle between the axes of *displacement* and *horsepower* selects all the data points in this bundle and highlights the bundles that contain the selected data points in red over the axes. In contrast to [87], the proposed EB method 1) eliminates the overplotting of the bundles in the area near the axes by aligning them around the centers of the intervals, 2) scales the widths of the bundles according to

their frequencies to reveal the distribution of the data, 3) renders the bundles with extremely low frequencies as dashed curves to highlight them. For example, as highlighted in the rectangles in Figure 4.8 (b), the bundles with different widths are aligned in the intervals. In addition, the bundles marked by the circles are rendered as dashed curves.

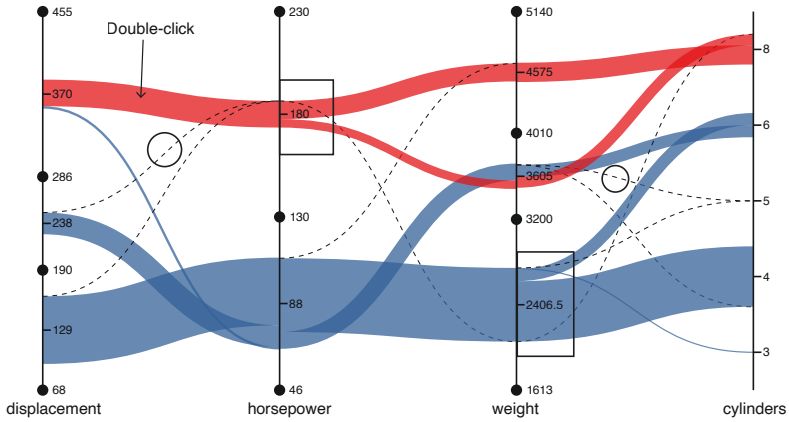


Figure 4.8: The visualization of the cars dataset with the proposed EB method.

Figure 4.9 shows the visualization of the cars dataset with the edge-bundled PC plot of Lima et al. [88]. The bundling method in [88] uses DBSCAN (density-based spatial clustering of applications with noise) to create multidimensional clusters and visually encodes the clusters information of each dimension, such as variance and means, into the curvature of the curves in the area near the axes. Between the axes, the curves belonging to the same cluster are rendered as a single merged curve to reduce visual clutter. However, DBSCAN leads to two shortcomings to the method: 1) as highlighted in the rectangles in Figure 4.9, the clusters are overlapped because DBSCAN groups some data records into two or more clusters, 2) data records with small-scale features in the data set cannot be visualized because DBSCAN removes the non-reachable data points during the clustering process. In Figure 4.9, 8.7% of the data are removed from the cars data set.

Figure 4.10 shows the user-adjusted visualization based on Figure 4.8. In Fig-

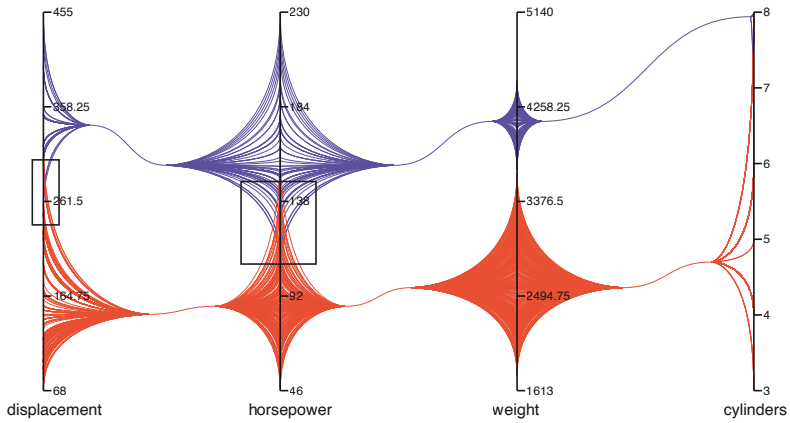


Figure 4.9: The visualization of the cars dataset with the edge-bundled PC plot of Lima et al [88].

ure 4.10, the two wider bundles marked by the circles show the similar clustering result in Figure 4.9. The red bundles are highlighted by double-clicking the bundle between the axes of *horsepower* and *weight*. In contrast to [88], the proposed EB method 1) displays the intervals with the clear boundaries, 2) visualizes data records with small-scale features in the dataset as thinner bundles or dashed curves. For example, as highlighted in the rectangles in Figure 4.10, the overlapped clusters in Figure 4.9 are separated by the boundary points. The not-displayed data records in Figure 4.9 are visualized as the thinner bundles and dashed curves in Figure 4.10.

Besides the visualization effectiveness, the proposed EB method shows the advantages over the methods in [87, 88] in scalability and user interactions. Table 4.1 shows the comparison of the proposed EB method with [87, 88] in terms of scalability and user interactions. To support the interaction of changing the number of clusters, the method in [87] pre-computes the clustering for different numbers of clusters per dimension. The pre-computation takes approximately 60 seconds for one dimension with 10^5 data points. The method in [88] requires multiple time-consuming iterations of DBSCAN to achieve usable clustering results. One computation of DBSCAN takes approximately 32 seconds to cluster 10^4 data records

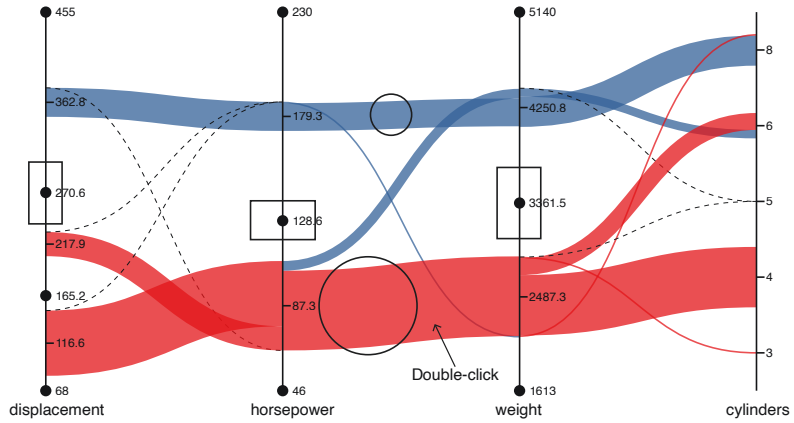


Figure 4.10: The user-adjusted visualization of the cars dataset based on Figure 4.8.

with 4 dimensions. As a comparison, our method takes approximately 1 second to cluster 10^6 data records with 4 dimensions.

Table 4.1: Comparison of Scalability and User Interactions of the Different Edge-Bundled PC plots.

Method	Scalability		Interactions on Clusters[2]
	Data volume	Running time[1]	
[87]	1×10^5	60	Change the number of clusters per variable.
[88]	4×10^4	32	No interactions described.
Our method	4×10^6	1	Change the number and range of intervals per variable.

[1] The approximate running time (in seconds) required for clustering the corresponding data set on desktop hardware.
[2] Only the interactions that change clustering result are compared.

In summary, the case study shows that the proposed EB method addresses the issues of the overplotted clusters and bundles in existing edge-bundled PC plots, and reveals the distribution of the data. In addition, with the high scalability and

interactions, it can well support IVA of multidimensional big data with PC plots.

4.1.3 User Study

I conducted a comparative user study to assess the user performance in two visualization tasks within the proposed EB method and the methods in [87, 88]. The study was performed using the static images of Figure 4.7 and Figure 4.8 without the highlight in red, and Figure 4.9 on a web questionnaire. For comparison, each participant did each task with both the proposed EB method and the methods in [87, 88]. The two tasks are described as follows:

- **T1: Tracing Subsets.** This task asks the participants to trace the subsets and count the number of the subsets over the axes. A subset is a path that contains the same data points over the axes. For example, in Figure4.8, the red bundles are two subsets/paths over the axes.

- **T2: Tracing Bundles.** This task asks the participants to trace a bundle and identify its range on an axis, which contains two types of the questions:

  - **Q1.** Tracing the bundle with the highest frequency and identifying its range on an axis. For example, with Figure 4.8, the participants were asked to trace the bundle with the highest frequency and identify its range of on the axis of *horsepower*.

  - **Q2.** Using the ranges on the axes to trace a bundle and identify its range on another axis. For example, with Figure 4.8, the participants were asked to trace the bundle with the range of 266 - 455 on the axis of *displacement* and the range of 4010 - 5140 on the axis of *weight* and identify its range on the axis of *horsepower*.

26 students and researchers at several universities participated in the user study. Each task was carefully explained to the participants. To avoid learning effects, on the web questionnaire, the images and questions of the three methods were randomly presented to each participant.

To assess the user performance in the two tasks, I calculated the mean error of each task and the correct rate for each type of question in task **T2** for each method as follows:

- **Mean Error.** I first calculated the absolute difference between the response and the ground-truth answer as the error of each question. Then, I calculated the mean error of each task for each method.

- **Correct Rate.** For each type of question in task **T2**, I counted the number of the correct responses and divided it by the total number of the responses to compute the correct rate for each method. It is important to note that a response of a smaller range than the ground-truth answer but identifies the same bundle is considered as a correct response for computing the correct rate.

Figure 4.11 shows the results of the user study. For comparison, I performed two paired t-tests for the proposed method and the methods in [87, 88] respectively for each task. As shown in Figure 4.11a, in task **T1**, the proposed EB method achieved the best performance with the smallest mean error (0.19). The mean errors of the proposed EB method and the method in [87] were close to zero, between which there was no statistically significant difference. However, the mean error of the proposed EB method was 95% smaller than that of the method in [88]. This difference was statistically significant at $p < 0.01$ ($t = 9.28$).

For question **Q1**, the proposed EB method achieved the best performance with the smallest mean error (33.5) and the highest correct rate (88%) (see in Figure 4.11b and Figure 4.11c). The mean error of the proposed EB method is 67% smaller than that of the method in [87] and 65% smaller than that of the method in [88]. Both the differences are statistically significant at $p < 0.01$ with $t = 3.58$ and $t = 2.94$ respectively.

As shown in Figure 4.11b, for question **Q2**, although the mean error of the proposed EB method was the largest (9.54), the mean errors of the three methods were

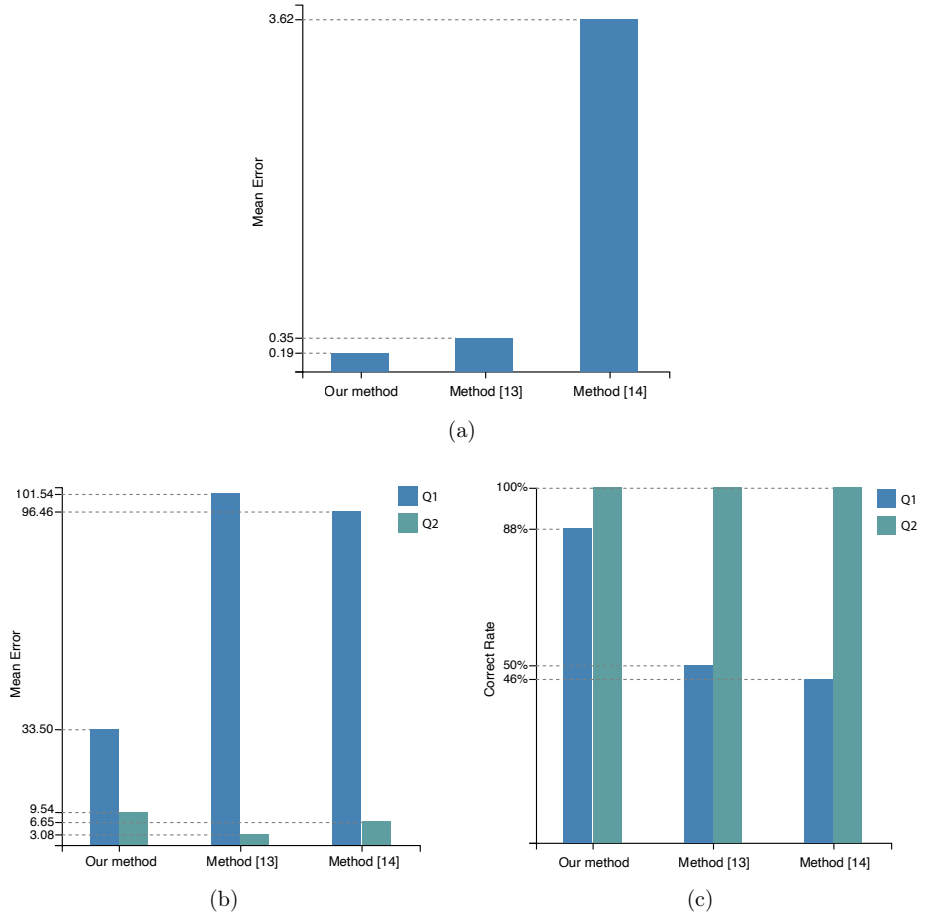


Figure 4.11: The results of the user study. (a) Mean error of task **T1** for each method. (b) Mean error of task **T2** for each method. (c) Correct rate of task **T2** for each method.

sufficiently small for tracing the bundles. Furthermore, for question **Q2**, there were no statistically significant differences between the mean errors of the proposed EB method and the methods in [87] and [88].

It is important to note that the correct rates of the three methods were 100% in

question **Q2** (see in Figure 4.11c). This indicates that, for the three methods, the participants may incorrectly underestimate the range of a bundle even though they traced the correct bundle by the intervals on the axes. For the proposed EB method and the method in [87], since the width of the bundle is smaller than the range (interval) of the bundle on the axis, the participants may be misled by the width of the bundle to underestimate the range of the bundle. For the method in [88], the intervals of different bundles on the axis may overlap each other and lead the participants to incorrect readings of the range of the bundle.

The user study shows that the proposed EB method achieves sufficiently well user performance in tracing bundles by the range (**Q2**) and better user performance than the methods in [87] and [88] in tracing subsets (**T1**) and tracing bundles by the frequency (**Q1**).

4.2 ColorPCA

4.2.1 Scalability Analysis

To examine the scalability of ColorPCA, the run-time analysis of ColorPCA with several synthesized large multidimensional datasets were performed. Each dataset was generated by several multivariate Gaussian distributions with different means and the same deviation [127]. The machine used in the tests has an Intel Core i5 Processor of 3.1 GHz and 8 Gigabytes of memory. Datasets were fully stored in the memory to avoid hard disk access.

ColorPCA contains three computation processes, including the computation of the dimension opacity, the color mapping, and the color contrast enhancement. According to Section 3.2.1, the computation of the dimension opacity has the same time complexity as the computation of PCA. Therefore, for an m-dimensional dataset S that contains n data points, the time complexity of the computation of the dimension opacity is $\mathcal{O}(nm^2+m^3)$, if PCA is computed by eigendecomposition of the data covariance matrix. The color mapping process is based on the linear accumulation process of ray casting, which has time complexity $\mathcal{O}(mn)$. The color contrast enhancement process uses the same computation to convert color space

for the composite color of each data point. Its time complexity is $\mathcal{O}(n)$.

According to the emission-absorption model, the first few dimensions will be enough for colorizing a high-dimensional dataset because later dimensions have less virtual light to pass through and thus contribute less to the composite color (see the case study in Section 4.2.2). This can limit m to a constant and reduce the time complexity of the color mapping process to $\mathcal{O}(n)$.

The computation of the dimension opacity is the most time-consuming process of ColorPCA. However, its time complexity can be reduced by replacing eigen-decomposition of the data covariance matrix with more efficient PCA algorithms [128, 129, 130]. More importantly, for a given dataset, the computation of the dimension opacity only needs to run once because the dimension opacity is a constant for each dimension. With ColorPCA, adjusting the color scheme and the color contrast is the key for users to explore the data using their perception of color. Each adjustment only requires re-running the color mapping and the color contrast enhancement processes. Therefore, as a one-time data pre-processing, the running time of the computation of the dimension opacity is not included in the running time of ColorPCA. And the time complexity of ColorPCA is $\mathcal{O}(mn + n)$, which increases linearly with m or n.

Figure 4.12 and Figure 4.13 show the approximate lines for the running times of ColorPCA, in seconds, for datasets with different values of m and n. Figure 4.12 shows that, for different values of n, the running time of ColorPCA increases linearly with m. Figure 4.13 shows that, for different values of m, the running time of ColorPCA increases linearly with n. As shown in Figure 4.12 and Figure 4.13, ColorPCA takes 0.97 seconds to map a 50-dimensional dataset with one million data points into colors and 1.41 seconds to map a $1,000$-dimensional dataset with $100,000$ data points into colors. Furthermore, if m is limited to a constant (for example, 10), ColorPCA can map high-dimensional datasets with millions of data points into colors in about 1 second. Therefore, the scalability of ColorPCA allows users to interactively explore multidimensional big data based on their perception of color.

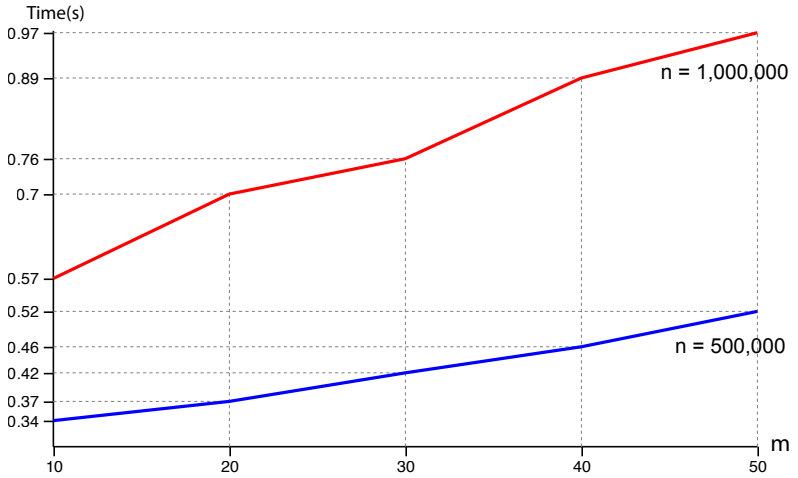


Figure 4.12: Running times (in seconds) of ColorPCA for multidimensional data sets with different values of m and n.

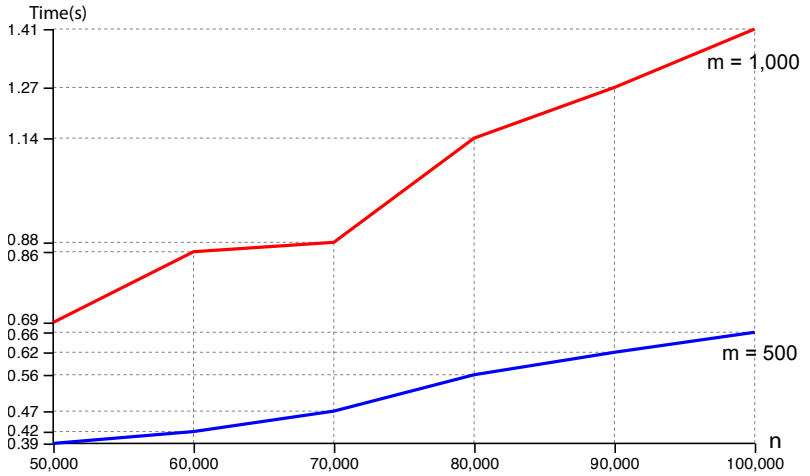


Figure 4.13: Running times (in seconds) of ColorPCA for high-dimensional data sets with different values of m and n.

4.2.2 Case Studies

Two case studies were conducted with three widely used multidimensional datasets to assess the usefulness of ColorPCA.

Case 1: Impact of the Color Schemes and Contrast on ColorPCA

The seed dataset [131] is a dataset of grain geometric features obtained by the X-ray technique. The dataset is 7-dimensional (seven measurements of geometrical properties of the kernel of wheat) with three classes defined by three varieties of wheat: Kama, Rosa, and Canadian. It contains 70 data points per class.

Figure 4.14 shows the PCA plots of the seed data with different colorization methods. In Figure 4.14a, the data is colorized by the first principal component (PC1) with the sequential color scheme shown in figure. Figure 4.14b and Figure 4.14c are colorized by ColorPCA with the sequential color scheme **CS1** and the categorical color scheme **CS2**, respectively. The first seven colors in the color schemes are assigned to the dimensions of the seed data in the descending order of the standard deviations of the dimensions. In Figure 4.14d, the data is colorized by the label, which shows the ground truth of the classes in the data.

The comparison of Figure 4.14a and Figure 4.14d shows that colorizing the data by PC1 cannot reveal the classes in the data by color. The comparison of Figure 4.14c and Figure 4.14d shows that although ColorPCA cannot guarantee the 100% accuracy in classifying the data points by color, ColorPCA reveals the three classes in the unlabeled seed data by automatically mapping the data points of the same class (with similar attribute values) into nearly the same color and the data points of different classes into distinguishable colors. Figure 4.14b shows better colorization result than Figure 4.14a, in which the data points for the three classes are mainly colorized in brown, dark red and light red, respectively. However, in contrast to Figure 4.14c, Figure 4.14b shows that the sequential color scheme can reduce the ability of ColorPCA to reveal the classes in the data. This is because the sequential color scheme reduces the differences between the colors assigned to the dimensions and thus reduces the variances between the dimensions mapped into the color space.

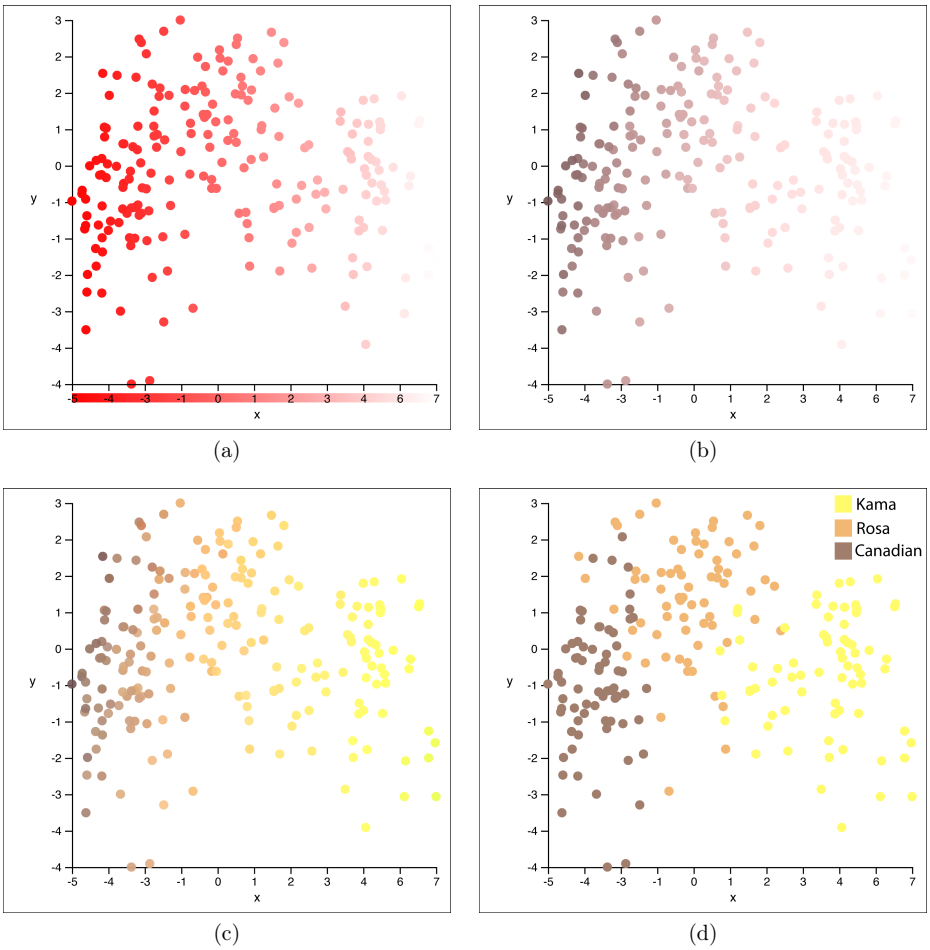
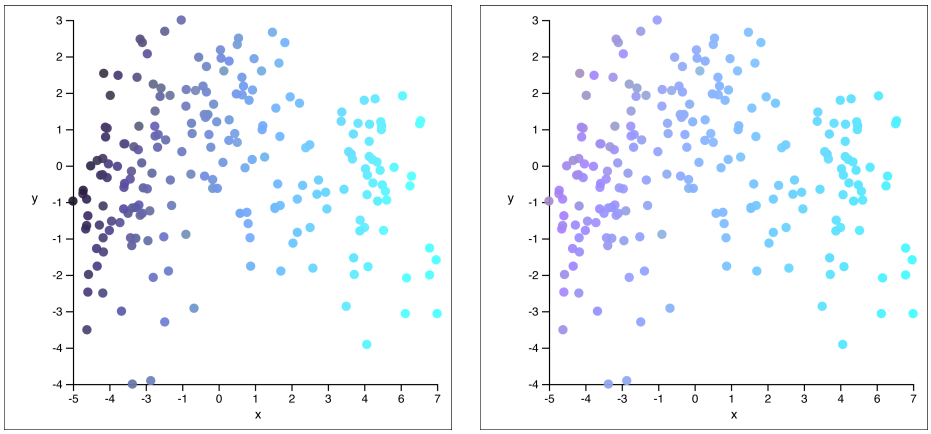


Figure 4.14: The PCA plots of the seed data. (a) Colorized by the first principal component (PC1) with the sequential color scheme. (b) Colorized by ColorPCA with **CS1**. (c) Colorized by ColorPCA with **CS2**. (d) Colorized by the label of the data.

The scalability of ColorPCA allows the user to adjust the range of the luminance and opacity in real-time to enhance the color contrast of the data. Figure 4.15 shows two PCA plots of the seed data colorized by ColorPCA with the automatically generated color scheme **CS3**. Figure 4.15a is colorized with the default



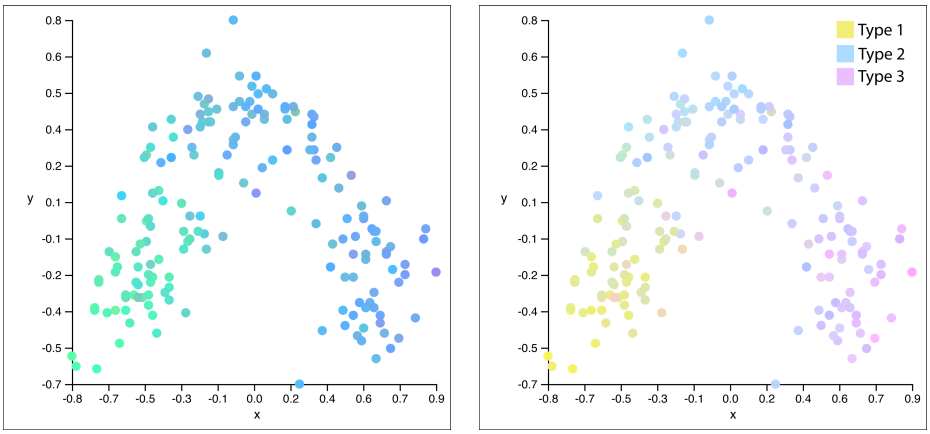
(a) (b)

Figure 4.15: Two PCA plots of the seed data colorized by ColorPCA with **CS3**. (a) Luminance: [0.01, 0.99]. (b) Luminance: [0.3-0.8].

luminance range $[0.01, 0.99]$. Figure 4.15b is colorized with the user-adjusted luminance range $[0.3, 0.99]$. In Figure 4.15a, the data points for the three classes are mainly colorized in dark purple (close to black), blue and cyan, respectively. In Figure 4.15b, the data points for the three classes are mainly colorized in purple, blue and cyan, respectively. In contrast to Figure 4.15a, Figure 4.15b shows the enhanced color contrast with the user-adjusted luminance range.

With ColorPCA, the colorization of the data with a default color scheme may not fully reveal the classes in unlabeled multidimensional data. However, ColorPCA allows users to adjust the color scheme based on their perception to explore the data by re-colorizing the data in real-time. It is to be noted that a color can be repeated assigned to different dimensions.

Figure 4.16 shows the PCA plots of the wine dataset [132]. The wine dataset is 13-dimensional (the quantities of thirteen constituents of the wines) with three classes derived from three different cultivars. Each class contains 59, 71, and 48 data points respectively. In Figure 4.16a, the data is colorized by ColorPCA with **CS3**, in which the data points of type 1 are mainly colorized in turquoise, and the



(a) (b)

Figure 4.16: Two PCA plots of the wine data colorized by ColorPCA. (a) With **CS3**. (b) With the user-adjusted **CS3**.

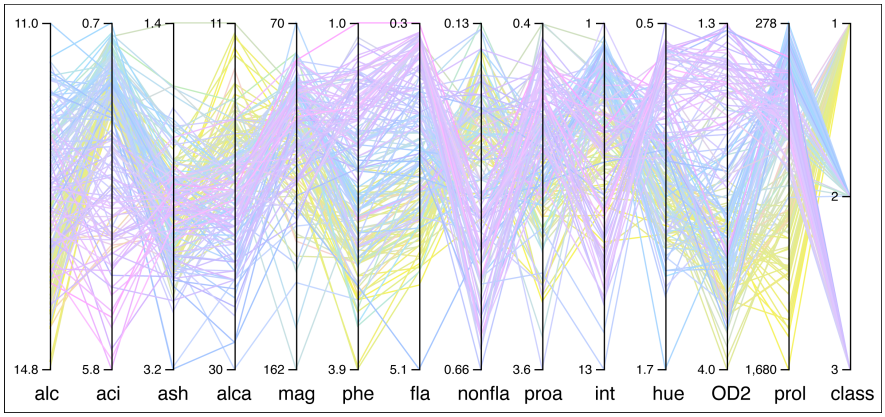


Figure 4.17: The PCP of the wine data.

data points of the other two types are both mainly colorized in blue. This hinders the user to distinguish the data points for type 2 and type 3 in the projection. Figure 4.16b is colorized by ColorPCA with the user-adjusted **CS3**, in which the data points of each type are mainly colorized in yellow-green, blue and purple respectively. For Figure 4.16b, the user adjusts **CS3** based on their perception

of color by two operations: 1) replace the first color in **CS3** with the seventh color in **CS3** and 2) replace the second color in **CS3** with the third color in **CS3**. The comparison of Figure 4.16a and Figure 4.16b shows that ColorPCA can help the users discover classes in the unlabeled multidimensional data by adjusting the color scheme based on their perception of color.

In addition to enhancing lower-dimensional projections of unlabeled multidimensional data, ColorPCA can be used to enhance other mappings of the data. Figure 4.17 shows the parallel coordinates plot (PCP) of the wine data, which is colorized by ColorPCA with the same setting of Figure 4.16b. In Figure 4.17, the polylines form three bundles in the three colors, which reveal the distribution and the relational patterns between each two adjacent axes for each type of the wine data.

Case 2: Apply ColorPCA to Unlabeled High-dimensional Big Data

Tuning parameters of DR algorithms is difficult and time-consuming, especially for unlabeled multidimensional big data. ColorPCA can help the user to find suitable parameters that balance the running time and the projection result of DR algorithms by automatically colorizing unlabeled multidimensional big data. This case study shows several t-SNE plots of the MNIST dataset. All the t-SNE plots are computed with the parallel Barnes-Hut implementation [133] of the t-SNE algorithm on a machine with an Intel Core i5 Processor of 3.1 GHz.

Perplexity is a tunable parameter of t-SNE, which balances the attention between local and global aspects of the data. It has large effects on the resulting plot and the running time of t-SNE, especially for big data. Figure 4.18 shows the different t-SNE plots of the data for digits 0 and 1 in the MNIST dataset, which are computed with different perplexity parameters. The plots in Figure 4.18 are colorized by ColorPCA with **CS3**. 784 colors are generated with **CS3** and assigned to the dimensions of the MNIST dataset in the descending order of the standard deviations of the dimensions.

For a given dataset, the t-SNE algorithm with the same parameters does not always produce similar plots. However, ColorPCA will always map the data to

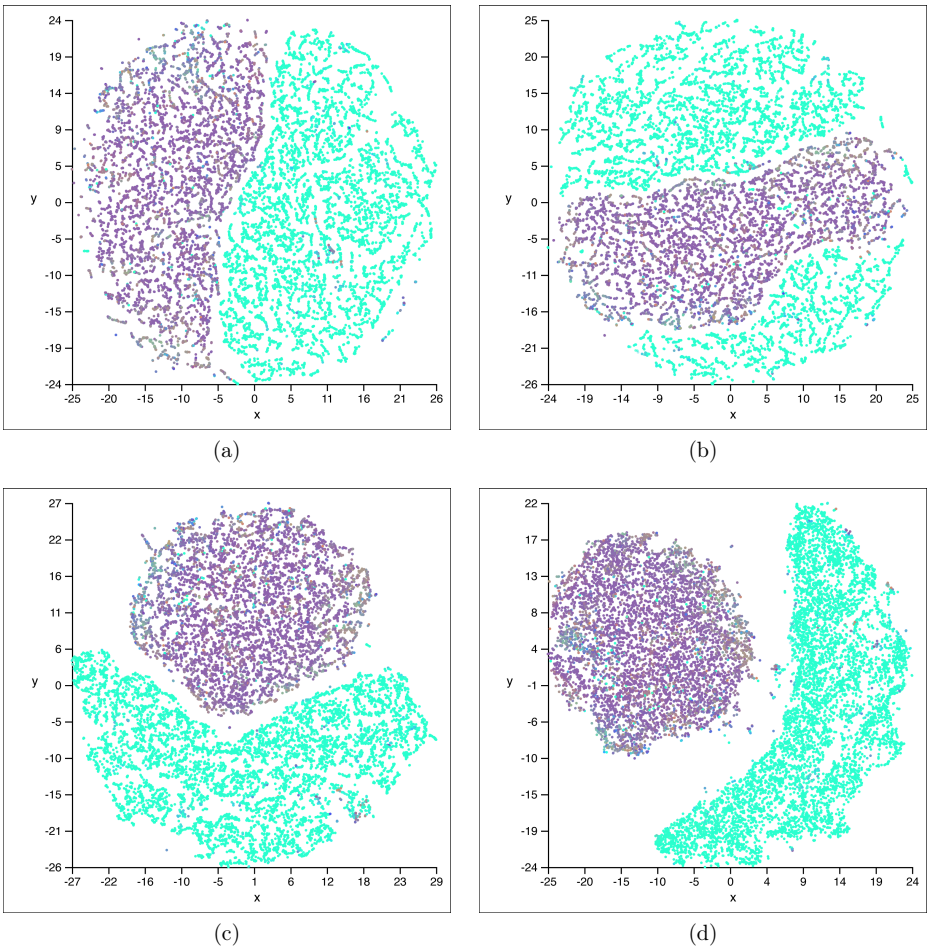


Figure 4.18: The t-SNE plots of the data for digits 0 and 1 in the MNIST dataset. (a) Perplexity: 5, step: 1000. (b) Perplexity: 5, step: 1000. (c) Perplexity: 20, step: 1000. (a) Perplexity: 100, step: 1000.

the same colorization using a given color scheme and the range of the luminance and opacity. In Figure 4.18, the data points of the two digits are colorized in purple (0) and turquoise (1) respectively by ColorPCA, which provides a hint to help the user find a suitable perplexity parameter. Figure 4.18a and Figure 4.18b show the

two different t-SNE plots, which are computed with the same perplexity of 5. The colorization in Figure 4.18a and Figure 4.18b show that, with the perplexity of 5, t-SNE cannot guarantee the data points for the two digits are always correctly projected into the two clusters. Therefore 5 is not a good choice for the perplexity. Figure 4.18c and Figure 4.18d are computed with the perplexity of 20 and 100 respectively. The colorization in Figure 4.18c and Figure 4.18d show that, with the perplexity of 20 and 100, t-SNE produces similar clustering results where the data points for the two digits are correctly projected into the two clusters. However, for the perplexity of 20 and 100, t-SNE takes 45.73 and 89.77 seconds to compute the data, respectively. Therefore, 20 is a better choice for the perplexity, which produces a sufficiently well t-SNE plot of the data and saves 49% of the running time compared to the perplexity of 100.

For high-dimensional data, ColorPCA can colorize the data with only the first few dimensions to discover classes in the data because later dimensions have less virtual light to pass through and thus contribute less to the composite color. Figure 4.19 shows the different t-SNE plots of the data for the digits 0 to 5 in the MNIST dataset, which are computed with different perplexity parameters. The plots in Figure 4.19 are colorized by ColorPCA with the first nine dimensions of the data and **CS2**. The dimensions are in the descending order of their standard deviations.

ColorPCA maps the data points with similar attributes to close positions in the color space. Therefore, ColorPCA may colorize the data points that belong to different classes but have similar attribute values in close colors. Furthermore, due to the limited number of the colors that can be easily distinguished in the color space, ColorPCA may not always colorize the data points for each class in a unique color. However, within the lower-dimensional projection of the data, ColorPCA can colorize the clusters belonging to the same class with a unique color combination to help the user discover classes in the data. And the user can tune parameters of DR algorithms with the following strategy:

- Merging the clusters that are colorized in the same color or the same color combination.

- Color combinations that consist of the same colors but with different proportions of each color are considered two different color combinations.

As shown in Figure 4.19a, three clusters are colorized in green and two clusters are colorized with the same color combination (CC) consisting of lots of purple and a little green and orange, which means that the data points for two classes in the dataset are incorrectly projected to three and two separated clusters, respectively. In contrast to Figure 4.19a, by using a larger perplexity parameter, in Figure 4.19b, the data points colorized with CC are merged into a cluster and the data points in green are still incorrectly projected to two separated clusters. In Figure 4.19c and Figure 4.19d, each cluster is colorized in a unique color or color combination. As marked in Figure 4.19c, the cluster for digit 0 and 1 are colorized in orange and green respectively, the cluster for digit 3 is mainly colorized in yellow and green, the cluster for digit 5 is colorized in yellow, green, purple and orange, the clusters for digit 2 and 4 are colorized in purple, green and orange with different proportions of green and orange. The colorization in Figure 4.19c and Figure 4.19d show that, with the perplexity of 30 and 50, t-SNE produces similar clustering results where the data points for the six digits are correctly projected into the six clusters. However, for the perplexity of 30 and 50, t-SNE takes 2.7 and 3.5 minutes to compute the data, respectively. Therefore, 30 is a better choice for the perplexity, which produces a sufficiently well t-SNE plot of the data and saves 23% of the running time compared to the perplexity of 50.

Using larger perplexity parameters does not ensure that the t-SNE algorithm can produce better clustering results. Figure 4.20 shows the t-SNE plots of the whole MNIST dataset, which are computed with different perplexity parameters. The plots in Figure 4.20 are colorized by ColorPCA with the first nine dimensions of the data and **CS3**. In Figure 4.20a, with the perplexity of 5, cluster 1 and 2 are colorized in the same color (turquoise), which indicates that these two clusters are incorrectly projected. As marked in Figure 4.20b, each cluster is colorized in a unique color or color combination, which shows that, with the perplexity of 10, the data points for the ten digits are correctly projected into ten clusters. The plots in Figure 4.20c and Figure 4.20d are computed with the perplexity

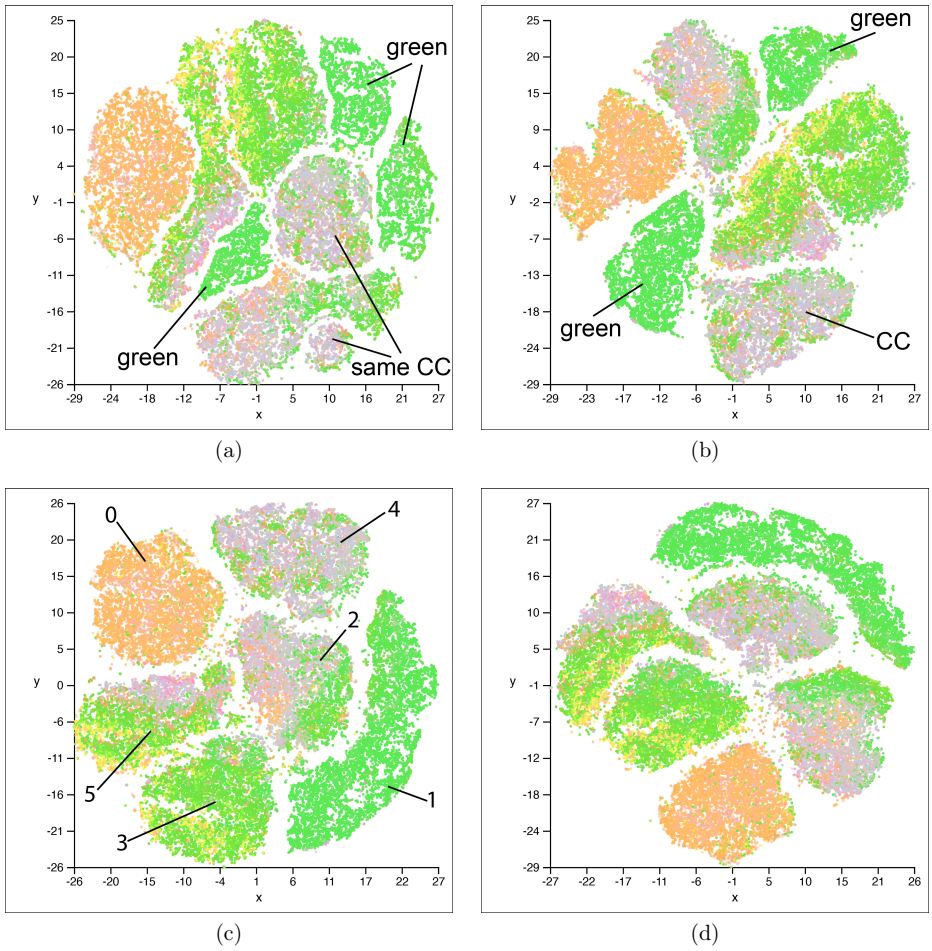
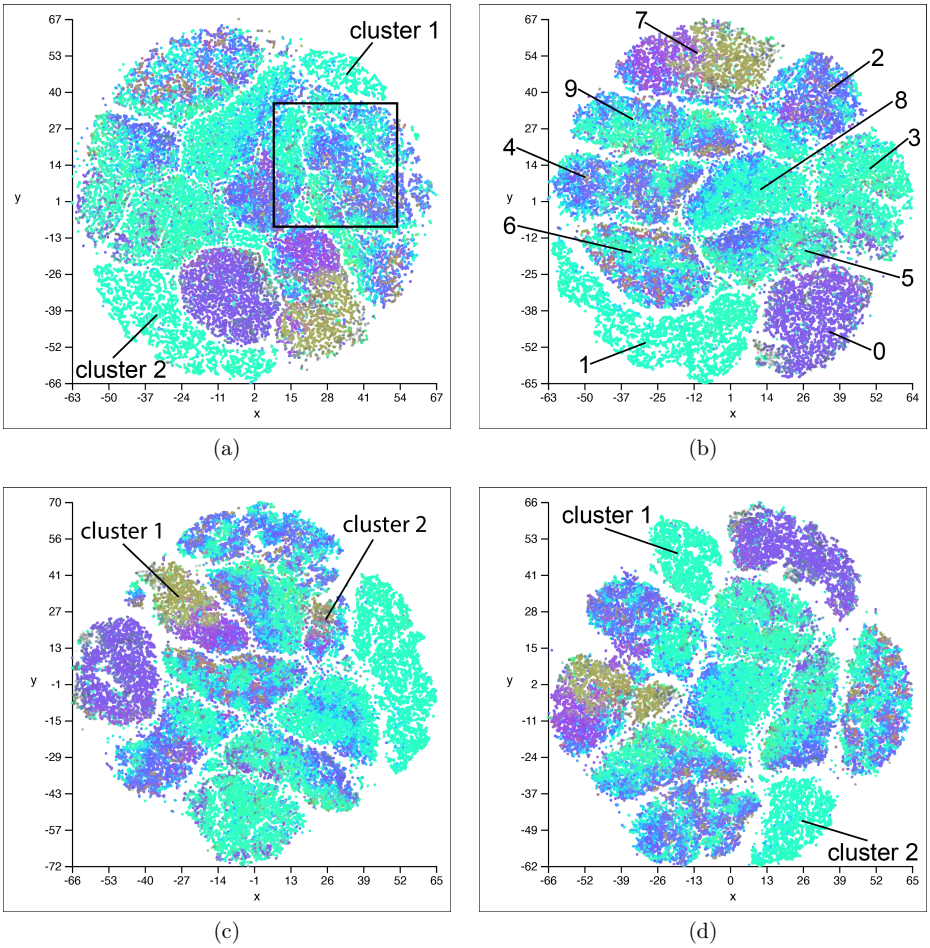


Figure 4.19: The t-SNE plots of the data for digits 0 to 5 in the MNIST dataset. (a) Perplexity: 10, step: 1000. (b) Perplexity: 25, step: 1000. (c) Perplexity: 30, step: 1000. (a) Perplexity: 50, step: 1000.

of 15 and 20, respectively. The comparison of Figure 4.20b, Figure 4.20c and Figure 4.20d shows that, with larger perplexity parameters, t-SNE can produce worse clustering results. For example, in Figure 4.20c, cluster 1 and 2 are colorized in similar color combination. In Figure 4.20d, cluster 1 and 2 are colorized in the

same color (turquoise). Moreover, in contrast to Figure 4.20b, t-SNE takes much more computation time to produce the plots in Figure 4.20c and Figure 4.20d.



(a) (b)

(c) (d)

Figure 4.20: The t-SNE plots of the MNIST dataset. (a) Perplexity: 5, step: 5000. (b) Perplexity: 10, step: 5000. (c) Perplexity: 15, step: 5000. (a) Perplexity: 20, step: 5000.

With the increasing number of classes in the data, ColorPCA may colorize the data points for several different classes with similar color combinations. For example,

in Figure 4.20a, the clusters in the rectangle area are colorized with similar color combinations. In Figure 4.20b, the clusters for digits 4, 6, and 9 are colorized in similar color combinations that consist of the same colors but with different proportions of each color. Similar color combinations may hinder the user to discover classes in the data, which is discussed in the user study. However, as shown in Figure 4.20, ColorPCA still provides a hint to help the user tune parameters of DR algorithms, even though the data points for several different classes are colorized in similar combinations.

4.2.3 User Study

A comparative user study was conducted to evaluate the effectiveness of ColorPCA in discovering classes/clusters in unlabeled multidimensional data by colorizing the projection of the data. The user study was performed with totally 18 static images on a web questionnaire, which includes Figure 4.18a, 4.18b, 4.18c, Figure 4.19a, 4.19b, 4.19c (without the annotations), and Figure 4.20a, 4.20b, 4.20d (without the annotations), and their corresponding uncolored figures. Figure 4.18d and a part of Figure 4.20c were used as examples to explain how to identify clusters in the figures using their colors or color combinations. To avoid learning effects, on the web questionnaire, the 9 uncolored images are presented first to participants, and both the colored and uncolored images are randomly presented to participants.

In the user study, the participants were asked to estimate the number of clusters/classes in each figure. To assess the user performance, the mean error and the correct rate for each figure were calculated as:

- **Mean Error.** First, the absolute difference between the participants' answers and the ground-truth number of the classes in the dataset was calculated as the error for each figure. Then, the mean error of each figure was calculated.

- **Correct Rate.** The number of the correct answers was counted and divided by the total number of the answers to compute the correct rate for each figure.

47 valid answers were collected and analyzed for each figure. Table 4.2 shows the results of the user study. For comparison, the paired t-tests were performed between the colored and uncolored figures for each dataset. The results of Figure 4.18a and Figure 4.18b show that, by colorizing the data, ColorPCA reduced the mean errors of the uncolored figures to 0 and improved the correct rates of the uncolored figures to 100%. The differences between the colored and uncolored Figure 4.18a and Figure 4.18b are statistically significant at $p < 0.01$ with $t = 6.952828$ and $t = 16.682326$, respectively. In Figure 4.18a and Figure 4.18b, ColorPCA colorized the data for each class in a unique color, which reveals that, in Figure 4.18b, the data are incorrectly projected to three clusters. Therefore, the correct answers of the uncolored Figure 4.18b were obtained coincidentally by the participants with the misleading projection of the data. Combined with the results of the case study, the user study results of Figure 4.18c show that ColorPCA helped the user find the suitable perplexity parameter for the dataset, of which all the participants correctly discovered the two classes in the data with the uncolored Figure 4.18c.

Table 4.2: The results of the user study of ColorPCA.

Figures	Uncolored		Colored by ColorPCA	
	Mean error	Correct rate	Mean error	Correct rate
Figure 4.18a	0.72	42.6%	0	100%
Figure 4.18b	0.94	10.6%	0	100%
Figure 4.18c	0	100%	0	100%
Figure 4.19a	2.98	0%	0.81	40.4%
Figure 4.19b	1	0%	0.19	78.7%
Figure 4.19c	0	100%	0.06	93.6%
Figure 4.20a	1.45	0%	2.52	4.3%
Figure 4.20d	1.09	0%	1.98	6.4%
Figure 4.20b	0	100%	0.83	61.7%

For the dataset containing 6 classes, the results of Figure 4.19a and Figure 4.19b show that, by colorizing the data, ColorPCA reduced the mean errors of the uncolored figures by 72.8% and 81% and improved the correct rates of the uncolored figures by 40.4% and 78.7%, respectively. The differences between the colored and uncolored Figure 4.19a and Figure 4.19b are statistically significant at $p < 0.01$ with $t = 16.308199$ and $t = 13.046072$, respectively. The results of the colored and uncolored Figure 4.19c show that, by colorizing the data, ColorPCA slightly increased the mean error of the uncolored figure to 0.06 and decreased the correct rate of the uncolored figure by 6.4%, which is statistically significant at $p < 0.05$ with $t = 1.770978$. This is because ColorPCA colorized the clusters for two classes into similar color combinations that consist of the same colors but with different proportions of each color and a few participants mistakenly grouped the clusters in the similar color combinations into the same class. However, the case study results show that the perplexity parameter used for the uncolored Figure 4.19c was selected based on the colorization of the data. The results of the uncolored Figure 4.19c show that ColorPCA still helped the user find the suitable perplexity parameter for the dataset, of which all the participants correctly discovered the classes in the data.

The results of Figure 4.20a and Figure 4.20d show that, by colorizing the data, ColorPCA increased the mean errors of the uncolored figures to 2.52 and 1.98 but also improved the correct rates of the uncolored figures to 4.3% and 6.4%, respectively. The differences between the colored and uncolored Figure 4.20a and Figure 4.20d are statistically significant at $p < 0.01$ with $t = 6.25645$ and $t = 4.467586$, respectively. For the dataset containing 10 classes, ColorPCA colorized several clusters for different classes into similar color combinations. Most participants underestimated the number of classes by mistakenly grouping the clusters in similar color combinations into the same class. And a few participants correctly distinguished the clusters for each class by distinguishing the similar color combinations based on the different proportions of each color. Therefore, the colored Figure 4.20a and Figure 4.20d increased both the mean errors and correct rates of the corresponding uncolored figures. The results of Figure 4.20b show that, due to the clusters in similar color combinations, ColorPCA increased the mean error of the uncolored

Figure 4.20b to 0.83 and decreased the correct rate of the uncolored Figure 4.20b by 38.3%, which is statistically significant at $p < 0.01$ with $t = 4.802952$. With the uncolored Figure 4.20b, all the participants correctly discovered the cluster for each class. The perplexity parameter used for Figure 4.20b was selected based on the colorization of the data, which means ColorPCA helped the user find the suitable perplexity parameter for the dataset.

The comparison of the correct rates of the colored Figure 4.19c and Figure 4.20b shows that, due to the limited number of distinguishable colors in the color space, as the number of classes in the dataset increases, ColorPCA will colorize more clusters into similar color combinations, which will reduce the correct rates of discovering classes in the datasets. However, ColorPCA still provides a hint to help the user tune parameters of DR algorithms even though the clusters of different classes may be colorized into similar color combinations. The increasing correct rates of the colored Figure 4.19a, 4.19b, 4.19c, and colored Figure 4.20a, 4.20b, 4.20d show that, during the process of tuning the parameter of the t-SNE algorithm, using ColorPCA on the projection of the data improved the correct rates of discovering classes in the data.

4.2.4 Discussion

ColorPCA aims to address the challenge of automatically colorizing unlabeled multidimensional big data for discovering classes in the data. It integrates PCA and ray casting to map the data into the RGBA color space, where different classes of the data are mapped to different locations in the color space and thus colorized with different colors or color combinations. By colorizing the data, it provides a fast way to enhance lower-dimensional projections of unlabeled multidimensional big data to help the user discover classes in the data and find suitable parameters of DR algorithms to balance the running time and the projection results.

ColorPCA is well scalable for unlabeled multidimensional big data. Based on the linear accumulation process of ray casting, the running time of ColorPCA increases linearly with the number of data points or the number of dimensions. The scalability analysis shows that, after a one-time pre-processing of the data,

ColorPCA takes 0.97 seconds to map a 50-dimensional dataset with one million data points into colors and 1.41 seconds to map a $1,000$-dimensional dataset with $100,000$ data points into colors. Furthermore, by colorizing the data with its first few dimensions (for example, 10), ColorPCA can map high-dimensional datasets with millions of data points into colors in about 1 second.

With the default color scheme and the range of the luminance and opacity, ColorPCA may not fully reveal the classes in unlabeled multidimensional big data by colorizing the data. However, the scalability of ColorPCA allows the user to adjust the color scheme and the range of the luminance and opacity based on their perception to explore the data by re-colorizing the data in real-time.

For a unlabeled multidimensional dataset with a small number of classes (for example, with 2, 3 or 6 classes), ColorPCA can colorize the data of each class with a unique color or color combinations to reveal the classes in the data. For a unlabeled multidimensional dataset with multiple classes, ColorPCA may colorize the data of several classes with similar combinations that consist of the same colors but with different proportions of each color, which will hinder the user to discover the classes in the data. However, the case study and the user study show that, in contrast to uncolored projections of unlabeled multidimensional big data, ColorPCA still provides a hint to help the user tune parameters of DR algorithms even though the clusters of different classes may be colorized into similar color combinations. And, during the process of tuning parameters of DR algorithms, using ColorPCA on the projection of unlabeled multidimensional big data can improve the correct rate of discovering classes in the data.

Chapter 5

Conclusion and Future Work

The main contributions and future directions of this research work are summarized and discussed in this chapter.

5.1 Contributions

This thesis aims to propose methods and build web-based applications to support IVA of multidimensional big data. The research of this thesis mainly includes three aspects: 1) a systematic review of VA; 2) a scalable lightweight EB method for PC; 3) a scalable method for colorizing unlabeled multidimensional big data. In the systematic review, I have proposed a novel taxonomy of VA applications with a focus on the dimensionality of data. The results of the systematic review show that PC and DR are two of the most widely used techniques for **multidimensional-transformation-2D** and **multidimensional-reduction-2D** VA applications for multidimensional data, respectively. In addition, I have discussed the scalability and interactions challenges and future directions of PC and DR-based VA for multidimensional big data. For PC-based VA, I have proposed a scalable lightweight EB method to support IVA of multidimensional big data. For DR-based VA, I have proposed a scalable method, named ColorPCA, to provide an efficient way for the users to explore unlabeled multidimensional big data and configure parameters and constraints of DR algorithms using their perception of color.

Furthermore, I have built two web-based VA applications based on the proposed EB method and ColorPCA for supporting IVA of multidimensional big data. The web-based applications can break time and space constraints in communication

and collaboration. The usefulness and effectiveness of the two proposed methods in this thesis were verified by the case studies and user studies based on the two applications.

To sum up, the main contributions of this thesis are:

- Conducted a systematic review of VA with a focus on the dimensionality of data, which examines over 200 publications, presents a novel taxonomy of VA applications and discusses challenges and future directions of VA applications.

- Proposed a scalable lightweight EB method for PC and built a web-based VA application based on the proposed method to support PC-based IVA of multidimensional big data.

- Proposed a scalable method ColorPCA for automatically colorizing unlabeled multidimensional big data and built a web-based VA application based on ColorPCA to support DR-based IVA of multidimensional big data.

5.1.1 Summary of Main Findings

This thesis developed two algorithms to address the challenges of IVA of multidimensional big data. To explain the main findings of the work, the summarized answers of the research questions proposed in Section 1.3 are as follows:

- **RQ1: What are the state-of-the-art techniques for interactive visual analysis of multidimensional data?**

  In the systematic review, about 80 VA applications have been examined with a focus on the dimensionality of data. The results show that PC and DR are the two most widely used techniques for IVA of multidimensional data. In addition, for PC, various EB methods have been proposed to reduce visual clutter in PC plots.

- **RQ2: What are the challenges for interactive visual analysis of**

multidimensional big data?

PC renders multidimensional data records as polylines across multiple parallel axes, which can be seen as trail-sets. Therefore, for multidimensional big data, PC plots suffer from the same visual clutter and overplotting as large graphs. More importantly, due the limited scalability of PC, the time-consuming process of rendering multidimensional big data as polylines can significantly delay the visual response of the **exploratory-oriented** interactions of PC-based VA applications, which hinders the users from gaining insight into data by exploring the patterns emerging from the visualization of data. A number of EB methods that bundle the data with different clustering algorithms and render the bundled data in different forms of visualizations have been proposed for reducing visual clutter and overplotting in PC plots. However, due to the limited scalability of the underlying clustering algorithms, existing EB techniques for PC only support limited **exploratory-oriented** interactions based on the pre-computation of the data. More importantly, existing EB techniques do not support the **expressive-oriented** interactions in PC that require re-clustering of the data, such as changing the bundling results based on users' perception and axes-reordering, which reduces the usability and usefulness of EB techniques for PC-based VA applications, especially for multidimensional big data.

For multidimensional big data, the limited scalability is the major challenge of DR algorithms, which can significantly increase the computation time and the time required for tuning parameters and constrains. For DR-based VA application, this can delay the visual response of **expressive-oriented** interactions that require re-computation of the data. For labeled multidimensional data, colorizing the data by label can provide a hint to help the users tune parameters and constraints of DR algorithms. However, automatically colorizing unlabeled multidimensional data to discover classes in the data and enhance the visualization of the data in lower-dimensional space has not been sufficiently investigated, especially for unlabeled multidimensional big data.

- **RQ3: For parallel coordinates, how to make the edge bundling process scalable by simplifying the underlying clustering algorithms and integrating human perception and judgments into the clustering process for supporting interactive visual analysis of multidimensional big data?**

  I have proposed a scalable lightweight EB method for PC. It simplifies the underlying clustering process by using two-dimensional data binning to cluster the data between each two adjacent axes independently. The proposed method is well scalable for bundling multidimensional big data in PC plots. The scalability analysis of the clustering process show that its running time increases linearly with the size of the data. It can cluster/bundle 1 million data records with 6 dimensions in about 1 second. It integrates human perception and judgments into the clustering process by novel interactions that allow the users to divide, adjust, and merge the intervals on each axis.

- **RQ4: For parallel coordinates, how to accelerate the rendering process of multidimensional big data without hardware-accelerated rendering using scalable edge bundling methods?**

  The proposed EB method uses the frequency-based representation to render the clusters as histogram-like bundles to reveal the distribution of the data, eliminate visual clutter and accelerate the rendering process. The scalability analysis of the rendering process show that it makes the rendering time independent of the size of the data. For example, it can render the bundling result of millions of multidimensional data points in real-time in web-based visualization without hardware-accelerated rendering.

- **RQ5: How to automatically colorize unlabeled multidimensional big data to discover classes in the data and enhance the visualization of the data in lower-dimensional space?**

  I have proposed a scalable method, named ColorPCA, to address the chal-

lenge of automatically colorizing unlabeled multidimensional big data for discovering classes in the data. It integrates PCA and ray casting to map the data into the RGBA color space, where different classes of the data are mapped to different locations in the color space and thus colorized with different colors or color combinations. By colorizing the data, it provides a fast way to enhance lower-dimensional projections of unlabeled multidimensional big data to help the user discover classes in the data and find suitable parameters of DR algorithms to balance the running time and the projection results. The case study and the user study show that, in contrast to uncolored projections of unlabeled multidimensional big data, during the process of tuning parameters of DR algorithms, using ColorPCA on the projection of unlabeled multidimensional big data can improve the correct rate of discovering classes in the data.

- **RQ6: How to support interactive visual analysis of unlabeled multidimensional big data by combining human perception of color and the automatic colorization of the data?**

  ColorPCA is well scalable for unlabeled multidimensional big data. Based on the linear accumulation process of ray casting, the running time of ColorPCA increases linearly with the number of data points or the number of dimensions. The scalability analysis shows that, after a one-time pre-processing of the data, ColorPCA takes 0.97 seconds to map a 50-dimensional dataset with one million data points into colors and 1.41 seconds to map a $1,000$-dimensional dataset with $100,000$ data points into colors. Furthermore, by colorizing the data with its first few dimensions (for example, 10), ColorPCA can map high-dimensional datasets with millions of data points into colors in about 1 second. The scalability of ColorPCA allows the user to adjust the color scheme and the range of the luminance and opacity based on their perception to explore the data by re-colorizing the data in real-time. Furthermore, I have built a web-based VA application based on ColorPCA. With this application, the users can explore the data by interactively adjust the colorization of the data.

5.2 Conclusion

A systematic review of VA has been conducted, which proposed a novel taxonomy of VA applications and discussed the challenges and future directions for IVA of multidimensional big data. The results of the systematic review lead to the two practical research of this thesis.

A scalable lightweight EB method is proposed for PC to support IVA of multidimensional big data. A web-based application is built based on the proposed EB method. The evaluation results show that the proposed EB method and the application work well for supporting IVA of multidimensional big data.

ColorPCA is proposed for automatically colorizing unlabeled multidimensional big data to help the users discover classes in the data and find suitable parameters of DR algorithms to balance the running time and the projection results. A web-based application is built based on ColorPCA. The evaluation results show that ColorPCA and the application work well for supporting IVA of unlabeled multidimensional big data.

5.3 Future Work

To improve IVA of multidimensional big data, the following extensions of the topics would be worthwhile to investigate for future research.

- For the proposed EB method, it would be worthwhile to investigate the use of statistical information of the clusters, such as means and variances, to position and render the bundles. For example, anchoring the bundles at the means of the clusters, and changing the color and transparency of the bundles according to the variances of the clusters. This may reveal more information in the data. However, this requires more computation and could reduce the scalability of the method.

- For ColorPCA, it would be worthwhile to investigate the use of ColorPCA for different types of unlabeled multidimensional data, such as revealing iso-

lated and grouped hot spots as well as uneventful areas in a colored map of unlabeled spatio-temporal multidimensional data.

- Extending the web-based systems with big data infrastructures and/or out-of-core algorithms would be worthwhile to investigate for further scaling, such as addressing the bottleneck of the disk I/O and network bandwidth for datasets that larger than the memory and accelerating the pre-computation process of ColorPCA.

Bibliography

[1] D. Thornton, R. M. Mueller, P. Schoutsen, and J. van Hillegersberg, "Predicting healthcare fraud in medicaid: A multidimensional data model and analysis techniques for fraud detection," *Procedia Technology*, vol. 9, pp. 1252–1264, 2013.

[2] M. Larance and A. I. Lamond, "Multidimensional proteomics for cell biology," *Nature reviews Molecular cell biology*, vol. 16, no. 5, pp. 269–280, 2015.

[3] R. Clarke, H. W. Ressom, A. Wang, J. Xuan, M. C. Liu, E. A. Gehan, and Y. Wang, "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," *Nature reviews cancer*, vol. 8, no. 1, pp. 37–49, 2008.

[4] W. Li and S. Wang, "Polarglobe: A web-wide virtual globe system for visualizing multidimensional, time-varying, big climate data," *International Journal of Geographical Information Science*, vol. 31, no. 8, pp. 1562–1582, 2017.

[5] M. Kreuseler, "Visualization of geographically related multidimensional data in virtual 3d scenes," *Computers & Geosciences*, vol. 26, no. 1, pp. 101–108, 2000.

[6] C. G. Healey, "Combining perception and impressionist techniques for nonphotorealistic visualization of multidimensional data," in *Proc. ACM SIGGRAPH*, vol. 1, pp. 20–52, 2001.

[7] S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker, "Visual exploration of high dimensional scalar functions," *IEEE Transactions on Visualization*

and Computer Graphics, vol. 16, no. 6, pp. 1271–1280, 2010.

[8] F. Kamalabadi, "Multidimensional image reconstruction in astronomy," *IEEE Signal Processing Magazine*, vol. 27, no. 1, pp. 86–96, 2010.

[9] A. Robotham and D. Obreschkow, "Hyper-fit: fitting linear models to multi-dimensional data with multivariate gaussian uncertainties," *Publications of the Astronomical Society of Australia*, vol. 32, 2015.

[10] M. Morháč, V. Matoušek, I. Turzo, and J. Kliman, "Daqprovis, a toolkit for acquisition, interactive analysis, processing and visualization of multi-dimensional data," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 559, no. 1, pp. 76–80, 2006. Proceedings of the X International Workshop on Advanced Computing and Analysis Techniques in Physics Research.

[11] S. Jesse, P. Maksymovych, and S. V. Kalinin, "Rapid multidimensional data acquisition in scanning probe microscopy applied to local polarization dynamics and voltage dependent contact mechanics," *Applied Physics Letters*, vol. 93, no. 11, p. 112903, 2008.

[12] A. Inselberg, *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009.

[13] A. Inselberg and B. Dimsdale, "Parallel coordinates: A tool for visualizing multi-dimensional geometry," in *Proceedings of the 1st Conference on Visualization '90*, VIS '90, (Washington, DC, USA), p. 361–378, IEEE Computer Society Press, 1990.

[14] E. J. Wegman, "Hyperdimensional data analysis using parallel coordinates," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 664–675, 1990.

[15] A. Inselberg, "The plane with parallel coordinates," *The Visual Computer*,

vol. 1, no. 2, pp. 69–91, 1985.

[16] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[17] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 3, pp. 1249–1268, 2017.

[18] H. Abdi and L. J. Williams, "Principal component analysis," *WIREs Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[19] M. A. A. Cox and T. F. Cox, "Multidimensional scaling," in *Handbook of data visualization*, pp. 315–347, Springer, 2008.

[20] A. J. Izenman, "Linear discriminant analysis," in *Modern multivariate statistical techniques*, pp. 237–280, Springer, 2013.

[21] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 07 1998.

[22] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[23] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[24] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003.

[25] Z. Zhang and J. Wang, "Mlle: Modified locally linear embedding using multiple weights," in *Advances in Neural Information Processing Systems*

(B. Schölkopf, J. Platt, and T. Hoffman, eds.), vol. 19, MIT Press, 2006.

[26] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, pp. 1373–1396, 06 2003.

[27] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, no. 60, pp. 1783–1816, 2005.

[28] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2018.

[29] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[30] J. Wang, X. Liu, H.-W. Shen, and G. Lin, "Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 81–90, 2017.

[31] C. A. Steed, J. E. Swan, T. Jankun-Kelly, and P. J. Fitzpatrick, "Guided analysis of hurricane trends using statistical processes integrated with interactive parallel coordinates," in *2009 IEEE Symposium on Visual Analytics Science and Technology*, pp. 19–26, 2009.

[32] H. Hauser, F. Ledermann, and H. Doleisch, "Angular brushing of extended parallel coordinates," in *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, pp. 127–130, 2002.

[33] R. C. Roberts, R. S. Laramee, G. A. Smith, P. Brookes, and T. D'Cruze, "Smart brushing for parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 3, pp. 1575–1590, 2019.

[34] L. F. Lu, M. L. Huang, and T.-H. Huang, "A new axes re-ordering method in parallel coordinates visualization," in *2012 11th International Conference on Machine Learning and Applications*, vol. 2, pp. 252–257, 2012.

[35] M. Blumenschein, X. Zhang, D. Pomerenke, D. A. Keim, and J. Fuchs, "Evaluating reordering strategies for cluster identification in parallel coordinates," *Computer Graphics Forum*, vol. 39, no. 3, pp. 537–549, 2020.

[36] J. Heinrich and D. Weiskopf, "State of the Art of Parallel Coordinates," in *Eurographics 2013 - State of the Art Reports* (M. Sbert and L. Szirmay-Kalos, eds.), The Eurographics Association, 2013.

[37] J. Choo, H. Lee, J. Kihm, and H. Park, "ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction," in *2010 IEEE Symposium on Visual Analytics Science and Technology*, pp. 27–34, 2010.

[38] N. Pezzotti, B. P. F. Lelieveldt, L. v. d. Maaten, T. Höllt, E. Eisemann, and A. Vilanova, "Approximated and user steerable tsne for progressive visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1739–1752, 2017.

[39] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, S. Leman, and C. North, "Towards a systematic combination of dimension reduction and clustering in visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 131–141, 2018.

[40] T. Fujiwara, Shilpika, N. Sakamoto, J. Nonaka, K. Yamamoto, and K.-L. Ma, "A visual analytics framework for reviewing multivariate time-series data with dimensionality reduction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1601–1611, 2021.

[41] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Transactions on*

Visualization and Computer Graphics, vol. 13, no. 6, pp. 1224–1231, 2007.

[42] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "Visual interaction with dimensionality reduction: A structured literature analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 241–250, 2017.

[43] T. Fujiwara, X. Wei, J. Zhao, and K.-L. Ma, "Interactive dimensionality reduction for comparative analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 758–768, 2022.

[44] S. Johansson and J. Johansson, "Interactive dimensionality reduction through user-defined combinations of quality metrics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 993–1000, 2009.

[45] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28 – 39, 2016.

[46] J. Johansson and C. Forsell, "Evaluation of parallel coordinates: Overview, categorization and guidelines for future research," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 579–588, 2016.

[47] A. Lhuillier, C. Hurter, and A. Telea, "State of the art in edge and trail bundling techniques," *Computer Graphics Forum*, vol. 36, no. 3, pp. 619–645, 2017.

[48] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[49] G.-D. Sun, Y.-C. Wu, R.-H. Liang, and S.-X. Liu, "A survey of visual analytics techniques and applications: State-of-the-art research and future chal-

lenges," *Journal of Computer Science and Technology*, vol. 28, no. 5, pp. 852–867, 2013.

[50] S. Chen, C. Wang, Z. Liu, Z. Wang, Z. Wang, Z. Miao, and X. Yuan, "Visual analytics support for collecting and correlating evidence for intelligence analysis," in *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pp. 319–320, IEEE, 2014.

[51] G. Andrienko, N. Andrienko, H. Bosch, T. Ertl, G. Fuchs, P. Jankowski, and D. Thom, "Thematic patterns in georeferenced tweets through space-time visual analytics," *Computing in Science & Engineering*, vol. 15, no. 3, pp. 72–82, 2013.

[52] M. Bögl, W. Aigner, P. Filzmoser, T. Lammarsch, S. Miksch, and A. Rind, "Visual analytics for model selection in time series analysis," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2237–2246, 2013.

[53] G. Andrienko, N. Andrienko, and S. Wrobel, "Visual analytics tools for analysis of movement data," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 38–46, 2007.

[54] N. Andrienko and G. Andrienko, "Visual analytics of movement: An overview of methods, tools and procedures," *Information Visualization*, vol. 12, no. 1, pp. 3–24, 2013.

[55] C. Donalek, S. G. Djorgovski, A. Cioc, A. Wang, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, A. Drake, *et al.*, "Immersive and collaborative data visualization using virtual reality platforms," in *Big Data (Big Data), 2014 IEEE International Conference on*, pp. 609–614, IEEE, 2014.

[56] J. Choo, H. Lee, J. Kihm, and H. Park, "ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction," in *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium*

on, pp. 27–34, IEEE, 2010.

[57] Y. Wu, S. Liu, K. Yan, M. Liu, and F. Wu, "Opinionflow: Visual analysis of opinion diffusion on social media," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1763–1772, 2014.

[58] J. C. Roberts, "State of the art: Coordinated amp; multiple views in exploratory visualization," in *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, pp. 61–71, 2007.

[59] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan, "Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection," in *Visualization Symposium (PacificVis), 2011 IEEE Pacific*, pp. 163–170, IEEE, 2011.

[60] E. Achtert, H.-P. Kriegel, E. Schubert, and A. Zimek, "Interactive data mining with 3d-parallel-coordinate-trees," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, (New York, NY, USA), p. 1009–1012, Association for Computing Machinery, 2013.

[61] J. Johansson, M. Cooper, and M. Jern, "3-dimensional display for clustered multi-relational parallel coordinates," in *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pp. 188–193, IEEE, 2005.

[62] K. Kurzhals and D. Weiskopf, "Space-time visual analytics of eye-tracking data for dynamic stimuli," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2129–2138, 2013.

[63] A. Endert, "Semantic interaction for visual analytics: Toward coupling cognition and computation," *IEEE computer graphics and applications*, vol. 34, no. 4, pp. 8–15, 2014.

[64] J. Heer and B. Shneiderman, "Interactive dynamics for visual analysis,"

Queue, vol. 10, no. 2, p. 30, 2012.

[65] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North, "Observation-level interaction with statistical models for visual analytics," in *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 121–130, 2011.

[66] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang, "ipca: An interactive system for pca-based visual analytics," in *Computer Graphics Forum*, vol. 28, pp. 767–774, Wiley Online Library, 2009.

[67] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen, "Data visualization with multidimensional scaling," *Journal of Computational and Graphical Statistics*, vol. 17, no. 2, pp. 444–472, 2008.

[68] G. Robertson, D. Ebert, S. Eick, D. Keim, and K. Joy, "Scale and complexity in visual analytics," *Information Visualization*, vol. 8, no. 4, pp. 247–253, 2009.

[69] S. G. Eick and A. F. Karr, "Visual scalability," *Journal of Computational and Graphical Statistics*, vol. 11, no. 1, pp. 22–43, 2002.

[70] P. C. Wong, H.-W. Shen, C. R. Johnson, C. Chen, and R. B. Ross, "The top 10 challenges in extreme-scale visual analytics," *IEEE computer graphics and applications*, vol. 32, no. 4, pp. 63–67, 2012.

[71] R. Ball and C. North, "Realizing embodied interaction for visual analytics through large displays," *Computers & Graphics*, vol. 31, no. 3, pp. 380–400, 2007.

[72] H. Schmauder, M. Burch, C. Muller, and D. Weiskopf, "Distributed visual analytics on large-scale high-resolution displays," in *Big Data Visual Analytics (BDVA), 2015*, pp. 1–8, IEEE, 2015.

[73] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in visual data analysis," in *Information Visualization, 2006. IV 2006. Tenth International Conference on*, pp. 9–16, IEEE, 2006.

[74] A. Malik, R. Maciejewski, B. Maule, and D. S. Ebert, "A visual analytics process for maritime resource allocation and risk assessment," in *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pp. 221–230, IEEE, 2011.

[75] A. Motamedi, A. Hammad, and Y. Asen, "Knowledge-assisted bim-based visual analytics for failure root cause detection in facilities management," *Automation in construction*, vol. 43, pp. 73–83, 2014.

[76] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale, "Beyond mouse and keyboard: Expanding design considerations for information visualization interactions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2689–2698, 2012.

[77] M. Spindler, S. Stellmach, and R. Dachselt, "Paperlens: advanced magic lens interaction above the tabletop," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pp. 69–76, ACM, 2009.

[78] J. Browne, B. Lee, S. Carpendale, N. Riche, and T. Sherwood, "Data analysis on interactive whiteboards through sketch-based interaction," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pp. 154–157, ACM, 2011.

[79] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz, "Uncovering clusters in crowded parallel coordinates visualizations," in /, pp. 81–88, IEEE, 2004.

[80] J. Heinrich and D. Weiskopf, "Continuous parallel coordinates," *IEEE Transactions on Visualization & Computer Graphics*, no. 6, pp. 1531–1538, 2009.

[81] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. Netto, and R. Buyya,

"Big data computing and clouds: Trends and future directions," *Journal of Parallel and Distributed Computing*, vol. 79, pp. 3–15, 2015.

[82] F. J. Newbery, "Edge concentration: A method for clustering directed graphs," *SIGSOFT Softw. Eng. Notes*, vol. 14, no. 7, p. 76–85, 1989.

[83] R. Pienta, J. Abello, M. Kahng, and D. H. Chau, "Scalable graph exploration and visualization: Sensemaking challenges and opportunities," in *2015 International Conference on Big Data and Smart Computing (BIGCOMP)*, pp. 271–278, 2015.

[84] A. Lhuillier, C. Hurter, and A. Telea, "FFTEB: Edge bundling of huge graphs by the fast fourier transform," in *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 190–199, 2017.

[85] X. Huang and C. Huang, "NGD: Filtering graphs for visual analysis," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 381–395, 2018.

[86] M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf, "Parallel edge splatting for scalable dynamic graph visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2344–2353, 2011.

[87] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauf, "An edge-bundling layout for interactive parallel coordinates," in *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pp. 57–64, IEEE, 2014.

[88] R. S. do Amor Divino Lima, C. G. R. dos Santos, S. de Paula Mendonça, J. M. a. de Morais, and B. S. Meiguins, "Understanding data dimensions by cluster visualization using edge bundling in parallel coordinates," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, (New York, NY, USA), p. 640–647, Association for Computing Machinery, 2018.

[89] M. Dickerson, D. Eppstein, M. T. Goodrich, and J. Y. Meng, "Confluent

drawings: Visualizing non-planar diagrams in a planar way," in *Graph Drawing* (G. Liotta, ed.), (Berlin, Heidelberg), pp. 1–12, Springer Berlin Heidelberg, 2004.

[90] B. Bach, N. H. Riche, C. Hurter, K. Marriott, and T. Dwyer, "Towards unambiguous edge bundling: Investigating confluent drawings for network visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, pp. 541–550, Jan 2017.

[91] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner, "Hierarchical parallel coordinates for exploration of large datasets," in *Proceedings Visualization '99 (Cat. No.99CB37067)*, (San Francisco, CA, USA, USA), pp. 43–508, IEEE Computer Society, 1999.

[92] M. Novotny and H. Hauser, "Outlier-preserving focus+context visualization in parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 893–900, Sep. 2006.

[93] K. T. McDonnell and K. Mueller, "Illustrative parallel coordinates," *Computer Graphics Forum*, vol. 27, no. 3, pp. 1031–1038, 2008.

[94] J. Johansson, P. Ljung, M. Jern, and M. Cooper, "Revealing structure within clustered parallel coordinates displays," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 125–132, 2005.

[95] M. van der Zwan, V. Codreanu, and A. Telea, "Cubu: Universal real-time bundling for large graphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2550–2563, 2016.

[96] V. Peysakhovich, C. Hurter, and A. Telea, "Attribute-driven edge bundling for general graphs with applications in trail analysis," in *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 39–46, 2015.

[97] A. Perrot and D. Auber, "Cornac: Tackling huge graph visualization with big

data infrastructure," *IEEE Transactions on Big Data*, vol. 6, no. 1, pp. 80–92, 2020.

[98] J. Sansen, G. Richer, T. Jourde, F. Lalanne, D. Auber, and R. Bourqui, "Visual exploration of large multidimensional data using parallel coordinates on big data infrastructure," *Informatics*, vol. 4, no. 3, p. 21, 2017.

[99] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *SIGMOD Rec.*, vol. 25, p. 103–114, June 1996.

[100] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz, "Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping," vol. 14, no. 3, pp. 564–575, 2008.

[101] J. Poco, R. Etemadpour, F. Paulovich, T. Long, P. Rosenthal, M. Oliveira, L. Linsen, and R. Minghim, "A framework for exploring multidimensional data with 3d projections," *Computer Graphics Forum*, vol. 30, no. 3, pp. 1111–1120, 2011.

[102] R. Etemadpour, R. Motta, J. G. d. S. Paiva, R. Minghim, M. C. F. de Oliveira, and L. Linsen, "Perception-based evaluation of projection methods for multidimensional data visualization," vol. 21, no. 1, pp. 81–94, 2015.

[103] P. Ready and P. Wintz, "Information extraction, snr improvement, and data compression in multispectral imagery," vol. 21, no. 10, pp. 1123–1131, 1973.

[104] J. Lawrence, S. Arietta, M. Kazhdan, D. Lepage, and C. O'Hagan, "A user-assisted approach to visualizing multidimensional images," vol. 17, no. 10, pp. 1487–1498, 2011.

[105] S. Cheng, W. Xu, and K. Mueller, "Colormapnd: A data-driven approach and tool for mapping multivariate data to color," vol. 25, no. 2, pp. 1361–1377, 2019.

[106] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley, "Dna visual and analytic data mining," in *Proceedings. Visualization '97 (Cat. No. 97CB36155)*, pp. 437–441, 1997.

[107] S. Cheng and K. Mueller, "Improving the fidelity of contextual data layouts using a generalized barycentric coordinates framework," in *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 295–302, 2015.

[108] A. Zeileis, K. Hornik, and P. Murrell, "Escaping rgbland: Selecting colors for statistical graphics," *Computational Statistics & Data Analysis*, vol. 53, no. 9, pp. 3259–3270, 2009.

[109] P. V. Kerm, "Adaptive kernel density estimation," *The Stata Journal*, vol. 3, no. 2, pp. 148–156, 2003.

[110] P. Sabella, "A rendering algorithm for visualizing 3d scalar fields," *SIGGRAPH Comput. Graph.*, vol. 22, p. 51–58, June 1988.

[111] B. Tang, G. Sapiro, and V. Caselles, "Color image enhancement via chromaticity diffusion," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 701–707, 2001.

[112] L. Lucchese, S. Mitra, and J. Mukherjee, "A new algorithm based on saturation and desaturation in the xy chromaticity diagram for enhancement and re-rendition of color images," in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, vol. 2, pp. 1077–1080 vol.2, 2001.

[113] S.-C. Pei, Y.-C. Zeng, and C.-H. Chang, "Virtual restoration of ancient chinese paintings using color contrast enhancement and lacuna texture synthesis," *IEEE Transactions on Image Processing*, vol. 13, no. 3, pp. 416–429, 2004.

[114] L. Lucchese and S. Mitra, "Filtering color images in the xyy color space,"

in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, vol. 3, pp. 500–503 vol.3, 2000.

[115] S. Liapis and G. Tziritas, "Color and texture image retrieval using chromaticity histograms and wavelet frames," *IEEE Transactions on Multimedia*, vol. 6, no. 5, pp. 676–686, 2004.

[116] M. Harrower and C. A. Brewer, "Colorbrewer.org: an online tool for selecting colour schemes for maps," *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.

[117] 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," vol. 13, no. 6, pp. 1270–1277, 2007.

[118] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.

[119] N. Max, "Optical models for direct volume rendering," vol. 1, no. 2, pp. 99–108, 1995.

[120] S. Westland, *CIE Chromaticity Coordinates (xyY)*, pp. 137–140. Springer, 2016.

[121] IEC, "61966-2-1: 1999 multimedia systems and equipment-colour measurement and management-part 2-1: Colour management-default rgb colour space-srgb," *International Electrotechnical Commission*, 1999.

[122] T. Chen, A.-X. Zhu, M. Wu, M. Chen, M. Zhang, W. Jiang, Y. Lu, and H. Wang, "A harmony-based approach to generating sequential color schemes for maps," *Color Research & Application*, vol. 45, no. 2, pp. 303–314, 2020.

[123] A. Light and P. J. Bartlein, "The end of the rainbow? color schemes for improved data graphics," *Eos, Transactions American Geophysical Union*, vol. 85, no. 40, pp. 385–391, 2004.

[124] C. L. Anderson and A. C. Robinson, "Affective congruence in visualization design: Influences on reading categorical maps," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2021.

[125] H. Fang, S. Walton, E. Delahaye, J. Harris, D. A. Storchak, and M. Chen, "Categorical colormap optimization with visualization case studies," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 871–880, 2017.

[126] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28 – 39, 2016.

[127] F. Iglesias, T. Zseby, D. Ferreira, and A. Zimek, "MDCGen: Multidimensional Dataset Generator for Clustering," *Journal of Classification*, vol. 36, pp. 599–618, Oct. 2019.

[128] S. Roweis, "Em algorithms for pca and spca," in *Advances in Neural Information Processing Systems* (M. Jordan, M. Kearns, and S. Solla, eds.), vol. 10, MIT Press, 1997.

[129] C. Chatterjee, Z. Kang, and V. Roychowdhury, "Algorithms for accelerated convergence of adaptive pca," *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 338–355, 2000.

[130] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust pca via gradient descent," in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[131] M. Charytanowicz, J. Niewczas, P. Kulczycki, P. A. Kowalski, S. Łukasik, and S. Żak, "Complete gradient clustering algorithm for features analysis of x-ray images," in *Information Technologies in Biomedicine* (E. Piętka and J. Kawa, eds.), pp. 15–24, Springer, 2010.

[132] S. Aeberhard, D. Coomans, and O. de Vel, "Comparative analysis of statistical pattern recognition methods in high dimensional settings," *Pattern Recognition*, vol. 27, no. 8, pp. 1065–1077, 1994.

[133] L. van der Maaten, "Accelerating t-sne using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, no. 93, pp. 3221–3245, 2014.

Appendixes

Appendix A

Paper 1

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 10, 2019, accepted June 3, 2019, date of publication June 19, 2019, date of current version July 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2923736

Visual Analytics: A Comprehensive Overview

WENQIANG CUI
Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering, NTNU–Norwegian University of Science and Technology, 6009 Ålesund, Norway

e-mail: wenqiang.cui@ntnu.no

This work was supported by the NTNU–Norwegian University of Science and Technology.

ABSTRACT With the ever-increasing amount of data, the world has stepped into the era of ''Big Data''. Presently, the analysis of massive and complex data and the extraction of relevant information, have been become essential tasks in many fields of studies, such as health, biology, chemistry, social science, astronomy, and physics. However, compared with the development of data storage and management technologies, our ability to gain useful information from the collected data does not match our ability to collect the data. This gap has led to a surge of research activity in the field of visual analytics. Visual analytics employs interactive visualization to integrate human judgment into algorithmic data-analysis processes. In this paper, the aim is to draw a complete picture of visual analytics to direct future research by examining the related research in various application domains. As such, a novel categorization of visual-analytics applications from a technical perspective is proposed, which is based on the dimensionality of visualization and the type of interaction. Based on this categorization, a comprehensive survey of visual analytics is performed, which examines its evolution from visualization and algorithmic data analysis, and investigates how it is applied in various application domains. In addition, based on the observations and findings gained in this survey, the trends, major challenges, and future directions of visual analytics are discussed.

INDEX TERMS Visual analytics, information visualization, interactive visualization, data analysis, analytical reasoning, knowledge representations, visual data mining, perception, cognition, sense-making, high-dimensional data.

I. INTRODUCTION

We are living in the age of data and advanced analytics. With recent advances in computing resources and data management technologies, our ability to generate, collect and store a wide variety of large and complex data sets continues to grow. According to the International Data Corporation's (IDC's) Digital Universe forecasts, the overall created and copied data volume worldwide will rise to approximately 40 zettabytes (ZB, 44 trillion GB) by 2020 [1]. This rapidly increasing amount of data has triggered an information revolution and enormous challenges that in turn will bring incredible scientific and industrial opportunities.

Nowadays, the analysis of massive amounts of data, which are typically messy, inconsistent and complex, as well as the subsequent extraction of relevant information, is becoming an essential task in numerous field of studies, such as health, biology, chemistry, social science, astronomy, and physics [2]. However, our ability to collect and store massive amounts of data far outstrips our ability to analyze the collected data [3], [4]. This has led to the well-known problem of ''information overload'' [5] (or the so-called ''data deluge'' [6]) in the age of information.

To address this data deluge, new technologies and methods have been investigated in many disciplines, such as visualization, statistics-based data analysis, machine learning, data mining, and perceptual and cognitive sciences, to extract useful information and generate reliable knowledge from unexplored data. However, it is questionable whether these sub-specialties are adequate to simply and effectively extract information from the ever-increasing massive data. Keim *et al.* indicated that ''approaches, which work either on a purely analytical or on a purely visual level, do not sufficiently help to filter substantial information from fast-growing complex data sets and to communicate it to humans in an appropriate way'' [7].

To generate knowledge and discover hidden opportunities from massive and complex data, James (Jim) Joseph Thomas (March 26, 1946 – August 6, 2010) created, promoted and established the visual-analytics field [8]–[10]. Visual analytics is ''the science of analytical reasoning facilitated by interactive visual interfaces'', which uses visualization and interaction techniques to integrate expert human judgment in the data analysis process [2], [3]. Such an approach requires

The associate editor coordinating the review of this manuscript and approving it for publication was Feng Xia.

TABLE 1. Some key terms related to visual analytics.

Key terms	Explanation
Visualization	Refers to the theories and techniques of creating visual representations of data, which includes information visualization and scientific visualization [11].
Information visualization	Refers to the theories and techniques that use (interactive) visual computing and representations to amplify human cognition with abstract information [12], [13].
Scientific visualization	Refers to the theories and techniques that use visual display and realistic renderings to gain information from spatial data associated with scientific processes [14].
Interactive visualization	Refers to the techniques that visualize and explore data by manipulating the color, brightness, size, and shape of its visual representation.
Human-computer interaction	Refers to "the study of the way in which computer technology influences human work and activities" [15].
Data analysis	Refers to the process of analyzing data with the goal of discovering useful information by applying statistical procedures and/or logical techniques [16]. In statistics, data analysis is divided into confirmatory and exploratory data analyses.
Confirmatory data analysis	Refers to the statistical process of evaluating pre-specified hypotheses (assumptions) on existing data sets (evidence) through a statistical hypothesis test [17].
Exploratory data analysis	Refers to approaches to analyzing data sets that reveal hidden and unknown information from data beyond the formal modeling or hypothesis-testing task.
Visual data mining	Refers to "the process of interaction and analytical reasoning with visual representations of data that leads to the visual discovery of robust patterns in these data that form the information and knowledge utilized in informed decision making" [18].
Visual analytics	Refers to the science of analytical reasoning supported by interactive visual interfaces [19].

the integration of algorithmic data analysis methods, innovative interactive techniques and data visualization, which allows decision makers to optimize the analytical-reasoning process and make sound decisions by their human flexibility, creativity, and background knowledge.

As information visualization has changed our view on databases, the ways of analyzing data and filtering information is being made transparent for an analytics discourse by visual analytics [4]. This article presents a complete picture of visual analytics to direct future research by examining the related research in various application domains. It gives an in-depth understanding of "*what is visual analytics*", "*how visual analytics is applied in various application domains*", "*the state of the art of visual analytics*", and "*what are the challenges and opportunities of visual-analytics research*".

There are several key terms, such as visualization, information visualization, scientific visualization, interactive visualization, human-computer interaction, data analysis, confirmation data analysis, exploratory data analysis, visual data mining and visual analytics, which are highly connected, easily confused and are also the key terms widely used in this article. Table 1 lists and explains them.

A. RECENT SURVEY STUDIES ON VISUAL ANALYTICS

Table 2 lists and compares recent surveys on visual analytics. It shows that existing surveys mainly concentrated on one aspect of visual analytics, such as its challenges, opportunities, techniques or applications in a specific field. This leaves a gap between its theory and applications when applying visual analytics in different application domains.

According to Table 2, although a few articles and references have discussed visual analytics from a theoretical perspective, they are generally narrowed to a single or two specific aspects of visual analytics, such as its definition, scope or processes. Additionally, they lack a connection between the theory and its applications. For instance, Keim *et al.* [20] compared the differences between visual analytics and information/scientific visualization from several aspects, including data analysis, perception and cognition, and human-computer interaction. However, the authors did not discuss these differences in related applications.

On the other hand, some visual-analytics surveys mainly focused on the techniques and applications without relating it to a theoretical background. Additionally, they are commonly limited to a single type of data or a specific application domain. For instance, Andrienko and Andrienko [21] presented a survey of the state-of-the-art visual-analytics techniques that support the analysis and understanding of various aspects of movement data. Caban and Gotz [22] and West *et al.* [23] presented systematic reviews of visual-analytics approaches which have been proposed to explore complex clinical data.

B. RESEARCH OBJECTIVES

Visual analytics has been applied in many different application domains, such as economics, bioinformatics, health, and social media. The ultimate purpose of this article is to draw a complete picture of visual analytics to direct future research by examining the related research in various application domains. It aims to bridge the gap between theory and practice when applying visual analytics in different application domains.

Sun *et al.* [28] classified visual-analytics applications into a set of categories, including *space and time*, *multivariate*,

TABLE 2. Recent surveys related to visual analytics.

Survey	Area of Interest	Pros	Limitations
[20] [2]	- Definition - Scope - Process - Challenges	An overview on visual analytics that compares and distinguishes visual analytics and information visualization from a theoretical perspective.	- Lack of discussion on applications. - Limited application challenges.
[24]	- Information in time and space. - Spatio-temporal analysis.	An overview of spatio-temporal visual analytics and its open issues.	- Lack of visual analytics background and its technical challenges. - Limited overview of applications.
[25]	- Commercial systems and frameworks.	An survey of the state-of-the-art commercial visual-analytics systems and frameworks for big data.	- Lack of visual analytics background. - Limited overview of commercial systems.
[26]	- Open source toolkits. - Visualization functions. - Analysis capabilities.	An overview of open-source visual-analytics toolkits.	- Lack of visual analytics background. - Limited overview of open-source toolkits.
[21] [27]	- Movement data. - Methods, tools and procedures.	A survey of visual-analytics techniques and applications for analyzing movement data.	- Lack of visual-analytics background and its challenges. - Discusses visual analytics on a single data type.
[28]	- Visual-analytics techniques and applications.	An overview of the state-of-the-art visual-analytics techniques and applications.	- Limited visual-analytics background. - Limited classification of applications. - Lack of technical challenges.
[29]	- Social media data.	A survey of the state-of-the-art visual-analytics techniques for analyzing social media data.	- Lack of visual-analytics background and its challenges. - Discusses visual analytics on a single data type.

text, *graph and network*, and *other applications*. This classification naturally differentiates visual-analytics applications to a specific data type or application domain. However, visual-analytics applications with different data types can share a common technique. For example, Jeong *et al.* [30] and El-Assady *et al.* [31] used the same visualization technique (parallel coordinates plots, PCPs) within multivariate and textual data, separately.

To avoid limiting this survey to a specific data type or application domain, a novel categorization of visual-analytics applications from a technical perspective is proposed, which is based on the dimensionality of visualization and the type of interactions. Based on this categorization, in this article an organized overview of visual analytics is constructed, which discusses the theory and evolution of visual analytics, and investigates how visual analytics is applied in various application domains. It aims to bridge the gap between the challenges of discovering knowledge in large and complex data sets and visual-analytics solutions by investigating state-of-the-art visual-analytics applications. In addition, the major challenges and future directions of visual analytics are targeted. To the best of our knowledge, this article is the first to classify visual-analytics applications from a technical perspective. By sharing the observations and findings gained in this survey, it is expected that this article could direct future research of visual analytics in different application domains.

In this survey, to demonstrate the proposed categorization and how visual analytics is applied in various disciplines, a careful examination of papers from premier conferences and journals that are related to visual analytics, such as Computer Graphics Forum (CGF), ACM SIGKDD Explorations, ACM Transactions on Graphics (TOG), IEEE Transactions on Visualization and Computer Graphics (TVCG), IEEE Visual Analytics Science and Technology (VAST), IEEE Information Visualization (InfoVis), EG/VGTC Conference on Visualization (EuroVis), and IEEE Pacific Visualization Symposium (PacificVis), is presented. The papers are filtered and analyzed within the Web of Science and Google Scholar according to the proposed categorization.

The remainder of the survey is organized as follows. In Section II, the evolution of visual analytics from data analysis and visualization is tracked, which addresses the fundamental question: "*What is visual analytics?*". In Section III, state-of-the-art visual-analytics techniques and applications are introduced. In particular, these applications are classified into eight categories according to the dimensionality of visualization and the type of interaction. In Section IV, the challenges and future research directions of visual analytics are discussed. Finally, in Section V the conclusions of this work are summarized.

II. FROM DATA ANALYSIS, VISUALIZATION TO VISUAL ANALYTICS

In this section, the question "*What is visual analytics?*" is addressed by investigating the evolution of visual analytics from visualization and algorithmic data analysis. The definition, model and process of visual analytics are discussed as the fundamentals of the proposed categorization of visual-analytics applications.

A. THE VISUAL ANALYTICS JOURNEY

Visual analytics is an outgrowth of the fields of scientific and information visualization. It is likely that the first appearance of "visual analytics" as a term in the literature was in the

"Guest Editors' Introduction-Visual Analytics" [32] of a special issue of IEEE Computer Graphics and Applications (CG&A) in 2004. In that introduction, *visual analytics* was defined as "the formation of abstract visual metaphors in combination with a human information discourse (interaction) that enables detection of the expected and discovery of the unexpected within massive, dynamically changing information spaces." Recently, by the combination of algorithmic data analysis and visualization, visual analytics started utilizing visualization as a medium and interaction as a means to involve human judgment in the data analysis process [33].

1) VISUALIZATION

Visualization can be broadly classified into scientific and information visualization. "Scientific visualization evolved first in the late 1980s, while information visualization matured in the mid-1990s" [34]. Scientific visualization focuses on visual display and realistic renderings of spatial data associated with scientific processes, for example, three-dimensional (3D) phenomena (architectural, meteorological, medical, biological, etc.) [35]. Information visualization examines visual representations of abstract and non-inherently spatial data which includes both numerical and non-numerical data, such as textual and geographical information [36], [37]. In information visualization, during the last decade, novel visualization techniques, such as parallel coordinates and its numerous extensions [38], tree-maps [39], Glyph- [40] and Pixel- [41] based visual data representations, have been developed to map a variety of abstract data to display space. Although scientific and information visualization have different research focuses and priorities, both of these subfields of visualization have the same goal: the visual communication of valuable data with understandable meaning. Accordingly, most research efforts in visualization have concentrated on producing different views.

2) DATA ANALYSIS

Data analysis is a process of modeling and exploring data with the goal of discovering useful information and supporting decision making by applying statistical procedures and/or logical techniques [16]. In statistical applications, data analysis is divided into confirmatory data analysis (CDA) and exploratory data analysis (EDA) [42]. CDA is a statistical process that evaluates pre-specified hypotheses (assumptions) on existing data sets (evidence) through a statistical hypothesis test [43]. It uses the traditional statistical tools of inference, significance, and confidence. In contrast, EDA is a quantitative process of isolating patterns and features of data, and revealing hidden and unknown information from data when little or no statistical hypotheses exist [44]. It is an approach which employs a variety of techniques (mostly visual methods) to summarize characteristics of data sets. It was first utilized in the statistics research community by Tukey in 1977 [45].

3) JOURNEY TO VISUAL ANALYTICS

As data volumes grow dramatically in a wide variety of fields, knowledge discovery in databases (KDD) was proposed at the first "Knowledge Discovery and Data Mining" workshop in 1989 [46]. KDD is the process of discovering understandable patterns in data, which emphasizes that knowledge is the end-product of the process [47]. With the goal of extracting useful information (knowledge) from data, KDD has evolved from the intersection of many research fields including statistics, pattern recognition, machine learning, artificial intelligence, and data visualization.

Before EDA was proposed, "data analysis techniques such as statistics and data mining developed independently from visualization and interaction techniques" [48]. Unlike CDA where visualization is used to present results, EDA employs visualization to interact with data. Therefore, moving from CDA to EDA is one of the most important steps in forming the research field of visual analytics.

In the information-visualization research community, with improvements in graphical user interfaces, "they recognized the potential of integrating the user in the KDD process through effective and efficient visualization techniques, interaction capabilities and knowledge transfer leading to visual data exploration or visual data mining" [48]. This implies a certain overlap between interactive visualization and visual analytics. However, interactions in interactive visualization are mainly used to present different views by manipulating graphical elements. In interactive visualization, much less has been discussed on interactions with data itself rather than interactions with graphical elements because data analysis is not "a must". To explore the relationship between visual data representation, data analysis and the knowledge discovery process, visual data mining was proposed and defined as "a step in the KDD process that utilizes visualization as a communication channel between the computer and the user to produce novel and interpretable patterns" [49]. Visual data mining is the process of interaction and analytical reasoning based on data visualization to discover understandable patterns (knowledge) in data [18].

Visual data mining considerably widened the scope of both the information-visualization and data-mining research fields. More importantly, as an important technique for visual analytics, visual mining data supports the formation of visual analytics by combining a collection of information-visualization metaphors and techniques with algorithmic data analyses through human information discourses (interactions) [50]. In 2004, visual analytics was first proposed by Wong and Thomas [32]. A year later, visual analytics was defined, illustrated and discussed in the book "Illuminating the path: The research and development agenda for visual analytics" [19]. More recently, visual analytics has become a multidisciplinary field that combines various research fields including visualization, human-computer interaction, data analysis, statistics, perception and cognition, and analytical reasoning. Figure 1 summarizes the visual

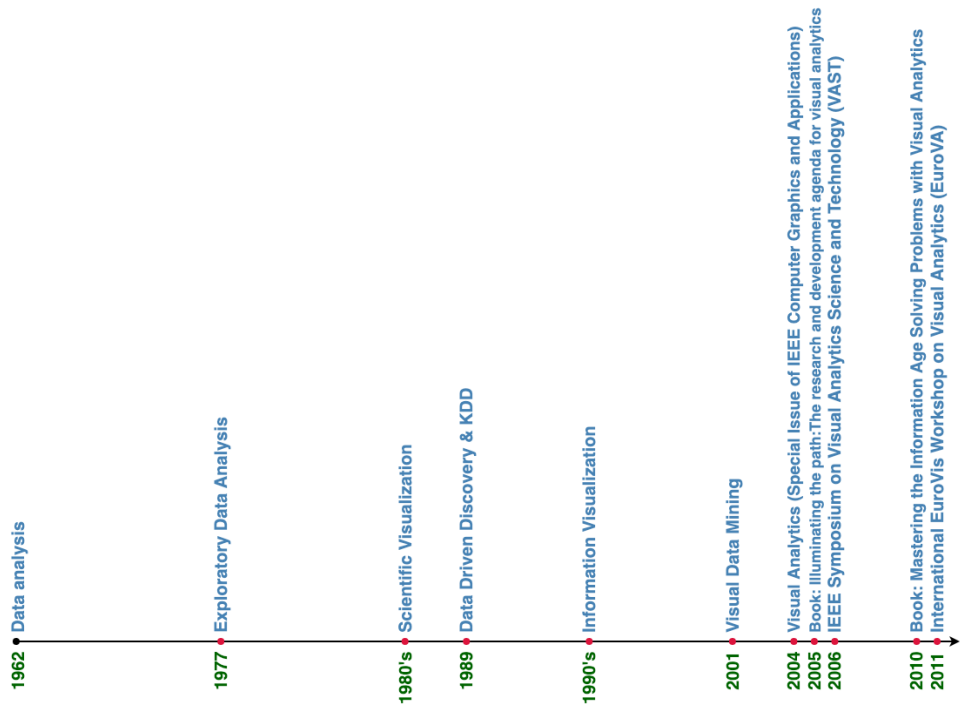


FIGURE 1. Visual analytics journey with respect to key events.

analytics journey. It presents the evolution of visual analytics in terms of representative moments, events and major aspects of its disciplinary development.

B. DEFINITION OF VISUAL ANALYTICS

Visual analytics was proposed to turn the information overload into an opportunity by creating tools and techniques to facilitate human judgment in the KDD process. In the book "*Illuminating the path: The research and development agenda for visual analytics*", visual analytics was first defined as:

Definition 1: Visual analytics is "the science of analytical reasoning facilitated by interactive visual interfaces" [19].

Nowadays, visual analytics, as an integrated approach combining visualization, algorithmic data analysis, human-computer interaction, and analytical reasoning, has attracted increasing interest from a wide range of domains and disciplines. With its development, researchers from different backgrounds have given detailed definitions of it with different focuses:

Definition 2: Visual analytics is "a method to synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data; detect the expected and discover the unexpected; provide timely, defensible, and understandable assessments; and communicate assessment effectively for action" [3].

Definition 3: Visual analytics "combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets" [20].

Built on the evolution of visual analytics from visualization and data analysis, a more detailed and comprehensive definition of visual analytics to emphasize its research goals is presented in this review:

Definition 4: Visual analytics is a multidisciplinary research field mainly based on visualization, algorithmic data analysis and analytical reasoning, which takes advantage of visualization and interactions as suitable tools to integrate human judgment into the KDD process to visually discover explainable patterns (knowledge) and to gain insight into large and complex data sets.

According to Definition 4, visual analytics has the same ultimate research goal as EDA, which is to discover knowledge and gain insight from data sets. However, visual analytics exploits visualization as a tool to integrate human cognition, perception abilities, and human intelligence into the data-analysis process to obtain explainable results. Relative to visualization, visual analytics places higher priority on analyzing data and discovering knowledge in data, rather than just presenting and understanding the data. Meanwhile, based on visualization, visual analytics addresses the challenge in data analysis that the discovered complex patterns could be hard to interpret in an intuitive and meaningful manner.

C. HUMAN INFORMATION DISCOURSE IN VISUAL ANALYTICS

According to Definition 1, as a science of analytical reasoning, the core idea of visual analytics is to integrate human cognitive, perceptual and reasoning abilities, and their knowledge into an analysis process to gain insight from data that is difficult to explore with pure visualization or analysis techniques. Analytical reasoning encompasses different kinds of reasoning, such as deductive, inductive, and analogical, which is based on a rational, logical analysis and evaluation of data [51]. Pohl *et al.* [52] discussed several theories, including sense-making theories, gestalt theories, distributed cognition, graph comprehension theories and skill-rule-knowledge models, and their relevance to visual analytics. In visual analytics, analytical reasoning is facilitated by creating appropriate visualizations and interactions that maximize human capacity to perceive and explore data. It adapts existing analysis processes by integrating visualization and algorithmic data analysis, which was discussed by [48].

Visual analytics is built upon an understanding of the reasoning process, as well as an understanding of the underlying cognitive and perceptual principles when applying human judgment to reach conclusions from data [3]. Human judgment is an integral part of the visual-analytics process, which relies on human-in-the-loop (HITL, a model that requires human interaction [53]) based interactions. In visual analytics, the interaction is not only a means to an end of finding a good representation of data, but also a valuable exploration process to apply human judgment and reveal insight from data [54]. Since interactions affect users' understanding of visually presented data, human-factor-based designs are the basis of visual analytics [55]. To study human factors in visual analytics, Green *et al.* [56] proposed a modeling framework of human "higher cognition". Miksch and Aigner [57] proposed a design triangle for visual-analytics methods that focused on time and time-oriented data. Dasgupta *et al.* [58] proposed a trust-augmented design of the visual-analytics system that explicitly took into account domain-specific tasks, conventions, and preferences.

Furthermore, recent work has emphasized that visual-analytics theories must move beyond HITL to "human-*is*-the-loop" analytics in order to integrate human cognition and reasoning process with analytics [59]. Figure 2 illustrates how human cognition, perception, and reasoning are employed in visual analytics. It shows that human judgment (perceptive skills, cognitive reasoning and domain knowledge) and algorithmic data analyses are effectively coupled through interactive visual representations in the visual-analytics process to gain insight from data.

D. THE VISUAL-ANALYTICS PROCESS

Shneiderman's celebrated mantra "Overview first, Filter and zoom, Details on demand" clearly emphasized the role of visualization in the knowledge-discovery process [60]. As visual analytics is an outgrowth of the fields of scientific and information visualization, to give an overview of the visual-analytics process, inspired by Shneiderman's mantra, a mantra is created here that focuses toward visual analytics: "Analyze/Overview first, interaction and visualization repeatedly, insights into data".

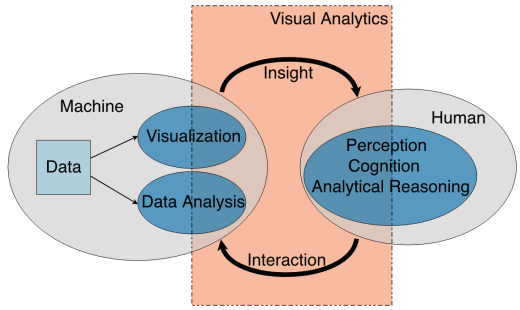


FIGURE 2. Visual analytics as the interplay between data analysis, visualization, and human analytical reasoning.

Based on the observations gained in this survey, the typical steps in the visual-analytics process are summarized as follows:

- *Step* 1 Preprocess (clean, transform, integrate) the data in order to prepare it for further processing.
- *Step* 2 Apply algorithmic analysis methods to the data.
- *Step* 3 Visualize the (processed) data with appropriate visualization techniques.
- *Step* 4 Users generate insightful knowledge through human perception, cognition, and reasoning activities.
- *Step* 5 Users make new hypotheses and integrate the newly generated knowledge into the analysis and visualization through interactions.
- *Step* 6 Regenerate an updated visualization based on the interactions to reflect the user's understanding of the data.

In many visual-analytics scenarios, heterogeneous data sources need to be integrated before algorithmic analysis methods or visualization can be applied. Therefore, the first step of the visual-analytics process is to preprocess data. The typical tasks in *Step* 1 could be data cleaning, normalization, transformation, grouping, and/or integration of the heterogeneous data into a common schema. In the visual-analytics process, knowledge can be gained from each step. However, the initial algorithmic analysis (*Step* 2) and visualization (*Step* 3) of the data are often not sufficient for problem-solving and decision making. Accordingly, human perception, cognition and reasoning activities are performed in *Step* 4 to generate insightful knowledge. Meantime, the knowledge is used for making new hypotheses. In *Step* 5, new knowledge and hypotheses are integrated into the data-analysis and visualization processes through interactions made by the user. Then, the data-analysis algorithms and visualizations are updated according to the user's interactions in *Step* 6. After the first loop of the visual-analytics process, it continuously iterates from *Step* 4 to *Step* 6 until enough

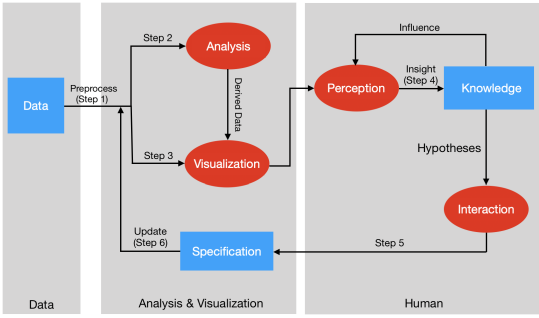


FIGURE 3. The visual-analytics process as a sense-making loop.

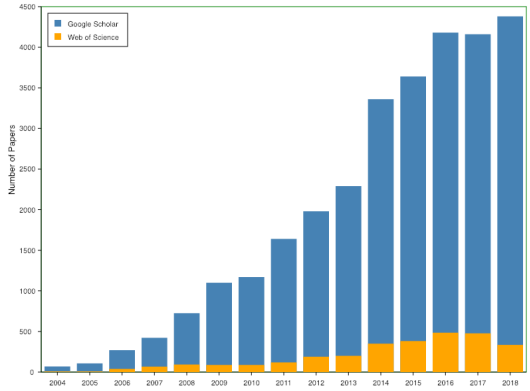


FIGURE 4. Trends in visual-analytics research based on the statistics of related papers. *Note:* The data were collected on 26 December 2018.

insight has been gained into the data for making decisions or solving the problems associated with the data. In some visual-analytics applications, *Step* 2 may be removed since it is not a must for all types of data and scenarios. This iterative process well illustrates the ''human-is-the-loop'' philosophy described in Section II-C. The generated knowledge is stored in the visual-analytics process through the feedback loop in *Step* 5, which enables the user to continuously draw faster and better conclusions and gain insight from the data. Figure 3 illustrates this visual-analytics process as a sense-making loop, in which each step is labeled. Figure 3 is composed and adapted from several diagrams, including analytical processes in visual analytics [61], visualization models [62], knowledge conversion processes [63] and knowledge generation models for visual analytics [64].

E. TRENDS IN THE FIELD OF VISUAL ANALYTICS

To discuss the trends in the field of visual analytics, the related academic papers in the Web of Science and Google Scholar, which are well-known academic database and search engine, are analyzed. In the Web of Science, all the papers which took visual analytics as the topic were counted and grouped by publication years from 2004 to 2018. Within Google Scholar, all the papers which discussed ''visual analytics'' as a term were searched, counted and grouped by publication years from 2004 to 2018.

Figure 4 illustrates the search results of the Web of Science and Google Scholar. Although the analysis of published papers in visual analytics does not reflect the full picture of the field, Figure 4 indicates the following: (1) Visual analytics is a relatively new research (it was created in 2004) area compared to other research fields, such as data analysis and visualization. (2) Visual analytics is a continuously and rapidly growing research field. In the field of visual analytics, the number of published papers in 2018 was six times larger than the corresponding number a decade previous.

The papers in the Web of Science were also analyzed according to the Web of Science Categories, as shown in Figure 5 as a tree-map. The figure shows that visual analytics is widely applied in different disciplines such as telecommunications, optics, cybernetics, geography, mathematical computational biology, education, medical informatics, remote sensing, etc.

III. VISUAL-ANALYTICS TECHNIQUES AND APPLICATIONS

Visual analytics has been applied in many different application domains, such as economics, bioinformatics, health, and social media. In this section, state of the art in visual-analytics applications are examined.

Sun *et al.* [28] identified five categories of visual-analytics applications according to the type of considered data. In their research, visual-analytics applications were classified as *space and time*, *multivariate*, *text*, *graph and network*, and *other applications*. However, they did not provide a comprehensive classification. Firstly, it is difficult to categorize a visual-analytics application which deals with several different types of data at the same time. For example, Chen *et al.* [65] proposed a visual-analytics system to analyze and explore multiple types of data and correlate them for intelligence analysis. The data analyzed in their system included GPS logs, which contained spatial and temporal data, news and email headers, which are textual data, and transaction logs which contained network data. Secondly, a complex data set may have two or more characteristics so that the corresponding visual-analytics application will be classified into two or more categories simultaneously. For example, Andrienko *et al.* [66] analyzed streaming-tweets data which consisted of geographical coordinates, time of tweeting, and the tweet text itself in their visual-analytics system. According to the classifications of [28], this visual-analytics application can be classified into the category *space and time* as well as the category *text*. Furthermore, with the rapid development of visual analytics in different domains, increasing numbers of visual-analytics applications can be classified into the category *other applications*.

To address the challenges arising from the limitations of the classification scheme of [28], and direct the future research of visual analytics, in this survey, a new comprehensive

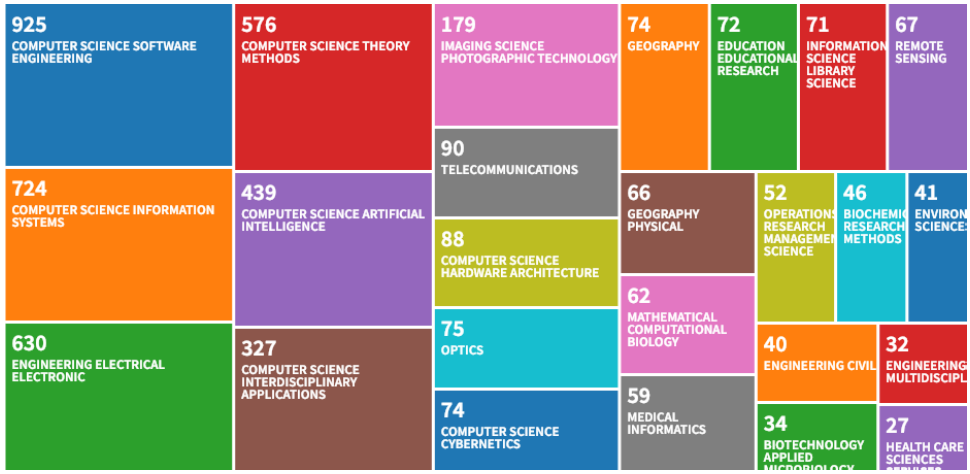


FIGURE 5. Application domains of visual analytics based on the statistics of related papers. *Note:* The data were collected on 26 December 2018.

categorization of visual-analytics applications from a technical perspective is proposed. According to the process of visual analytics summarized in Section II-D and the "human-is-the-loop" philosophy described in Section II-C, two technical components of visual analytics are identified: visualization and interactions. To integrate human judgment into the data-analysis process in visual analytics, users gain insight from data through visualization and apply their judgment to the data through interactions, such as zooming in different visualization areas, changing visualization methods, modifying the parameters of data models, and investigating different visual views on data. Therefore, visual-analytics applications can be categorized according to the dimensionality of visualization and the type of interaction.

A. VISUALIZATION-BASED CLASSIFICATION

According to the dimensionality of the data and visualization techniques, four categories of visual-analytics applications are identified: *2D-to-2D*, *multi-dimensional-reduction-2D*, *multi-dimensional-to-2D*, and *multi-dimensional-to-3D*.

1) 2D-TO-2D

Two-dimensional (2D) visualization is the most common way to visualizing data in information visualization. Within 2D visualization, binary data is naturally visualized in 2D space through the Cartesian coordinate system. A visual-analytics application will be classified as *2D-to-2D* if it fulfills the following requirements:

- The data is 2D.
- The data are visualized in 2D visualization.

For visual-analytics applications in this category, users will gain insight from data by performing analytical reasoning on 2D data through 2D visualization. For example, Bögl *et al.* [67] developed a *2D-to-2D* visual-analytics application (TiMoVA) to guide domain experts in model-selection tasks based on user stories and iterative expert feedback on users experiences. It closely combined human perception and analytical reasoning and automated computation. Figure 6 shows an overview of TiMoVA.

In addition, *2D-to-2D* visual analytics are commonly used for another type of 2D data: movement data. The research of [21], [68] used different visualization techniques, such as mapping and clustering movement data on 2D maps, to analyze and explore various aspects of movement through visual analytics.

2) MULTI-DIMENSIONAL-REDUCTION-2D

With the ever-increasing amount of data sets, multi-dimensional data show up in numerous fields of study, such as economics, biology, chemistry, political science, astronomy, and physics [69]. In this survey, multi-dimensional data are defined as:

Definition 5: A data set that has more than three dimensions/attributes.

However, the high dimensionality of a multi-dimensional data set represents a critical obstacle: humans are biologically optimized to see the world and the patterns in it in three dimensions [70]. This challenge and the wide availability of multi-dimensional data have led to new opportunities for visual analytics.

A visual-analytics application will be classified as *multi-dimensional-reduction-2D* if it fulfills the following requirements:

- The data is multi-dimensional.
- The dimensionality of the data is reduced by algorithmic approaches to two dimensions.
- The processed data are visualized in 2D visualization.

FIGURE 6. TiMoVA Overview. A *2D-to-2D* visual-analytics application for finding an adequate model for a given time-oriented data set [67].

To break the physical limitations of the human visual system, a variety of analysis-centric dimension-reduction methods have been investigated for reducing the dimensions of multi-dimensional data, such as principal component analysis (PCA), multi-dimensional scaling (MDS) and linear discriminant analysis. However, it is usually difficult to understand and interpret the result of these algorithmic approaches in an intuitive and meaningful manner. To address this challenge, *multi-dimensional-reduction-2D* visual-analytics applications integrate dimension-reduction approaches into the human analytical reasoning process to reduce the data items presented in the visualization. For example, Choo *et al.* [71] presented a *multi-dimensional-reduction-2D* visual-analytics system (iVisClassifier) for classifications based on a supervised dimension-reduction approach, which is shown in Figure 7.

Wu *et al.* [72] introduced a *multi-dimensional-reduction-2D* visual-analytics system (OpinionFlow) to empower analysts to detect opinion-propagation patterns and glean insights, which is shown in Figure 8. OpinionFlow uses an information diffusion model to reduce the dimension of the social-media data.

3) MULTI-DIMENSIONAL-TRANSFORMATION-2D

Another category of visual-analytics applications is *multi-dimensional-transformation-2D*, which visualizes multi-dimensional data without analysis-centric dimension-reduction approaches. A visual-analytics application will be classified as *multi-dimensional-transformation-2D visualization* if it fulfills the following requirements:

- The data is multi-dimensional.
- The multi-dimensional data is transformed and mapped in 2D visualization.
- The dimension of the data is not reduced by algorithmic approaches.

Within *multi-dimensional-transformation-2D* visual-analytics applications, multi-dimensional data is transformed and mapped in 2D space, which encodes data to different representations, such as PCPs and coordinated multiple views (CMVs). PCPs align axes parallel to each other and data points are mapped to lines intersecting the axes at the respective values. They allow the simultaneous display of a number of dimensions by embedding the corresponding number of parallel axes into a plane to reveal trends and patterns in the data. CMVs encompass a specific exploratory visualization technique that uses two or more distinct views to support the investigation of a single conceptual entity [73]. Guo *et al.* [74] presented a *multi-dimensional-transformation-2D* visual-analytics system, Triple Perspective Visual Trajectory Analytics (TripVista), for exploring and analyzing complex traffic trajectory data, which was mainly based on a parallel coordinate plot and coordinated multiple views. TripVista is shown in Figure 9.

4) MULTI-DIMENSIONAL-TO-3D

Three-dimensional (3D) visualization was developed for converting 3D objects/phenomena into 2D images through a computer-graphics process. Presently, 3D visualization is widely used in scientific visualization to graphically illustrate scientific data, which enables scientists to understand and illustrate the data. Moreover, 3D visualization is often integrated with a variety of approaches to visually analyze multi-dimensional data. For example Achtert *et al.* [75] and Johansson *et al.* [76] visualized parallel coordinates in 3D space to explore the complicated relationships between the axes, which arranged more than two neighboring axes around the central attribute.

Multi-dimensional-to-3D visual-analytics applications are based on the 3D visualization of multi-dimensional data. A visual-analytics application will be classified as *multi-dimensional-to-3D* if it fulfills the following requirements:

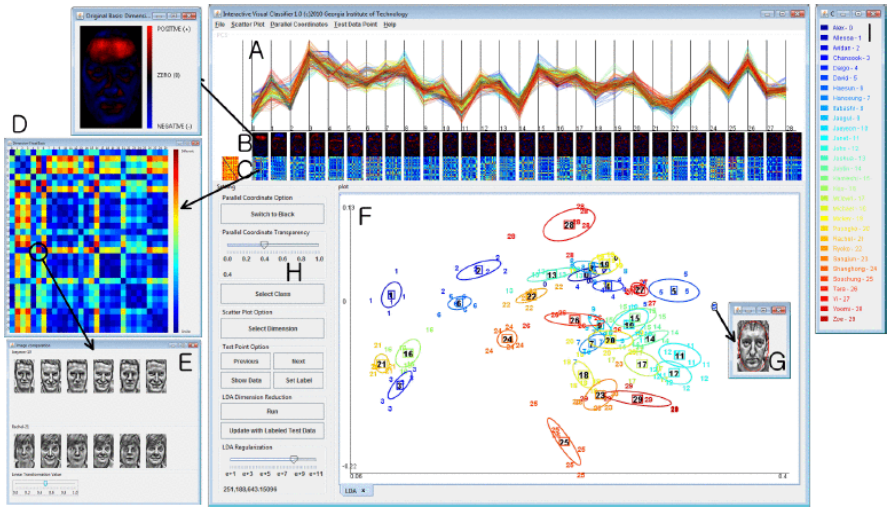


FIGURE 7. An overview of iVisClassifier [71].

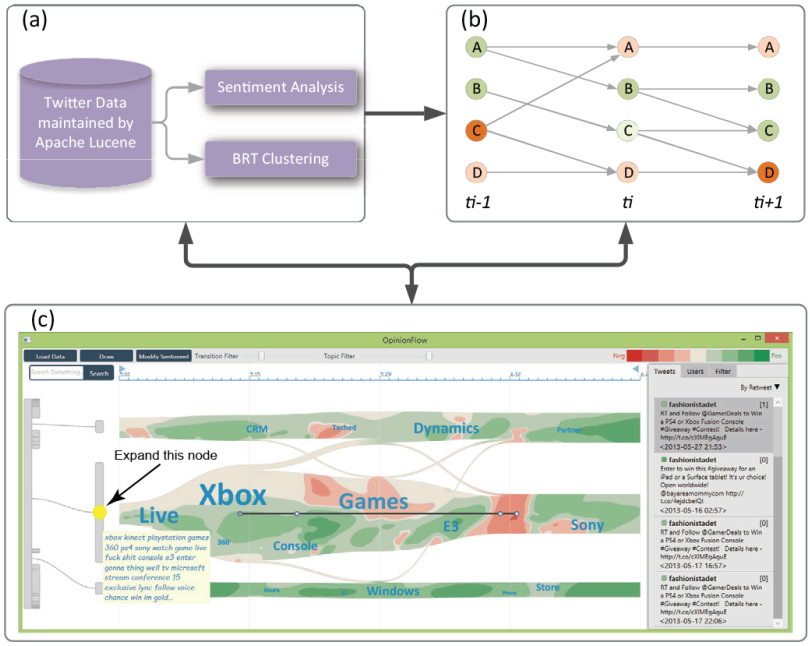


FIGURE 8. Three major parts of OpinionFlow: (a) Data preprocessing, (b) diffusion modeling, and (c) interactive visualization [72].

– The data is multi-dimensional.
– The multi-dimensional data is transformed and mapped in 3D visualization.

For example, Kurzhals and Weiskopf [77] introduced a *multi-dimensional-to-3D* visual-analytics method to analyze eye-tracking data recorded for dynamic stimuli such as video or animated graphics, which is shown in Figure 10.

B. INTERACTION-BASED CLASSIFICATION

In visual analytics, the analytical-reasoning process is facilitated by interactive visual exploration of data through various interaction techniques. According to the visual-analytics process (II-D), users can directly interact with data, algorithms, and visualization [78]. Heer and Shneiderman [79] gave a taxonomy of interactive dynamics for visual analysis,

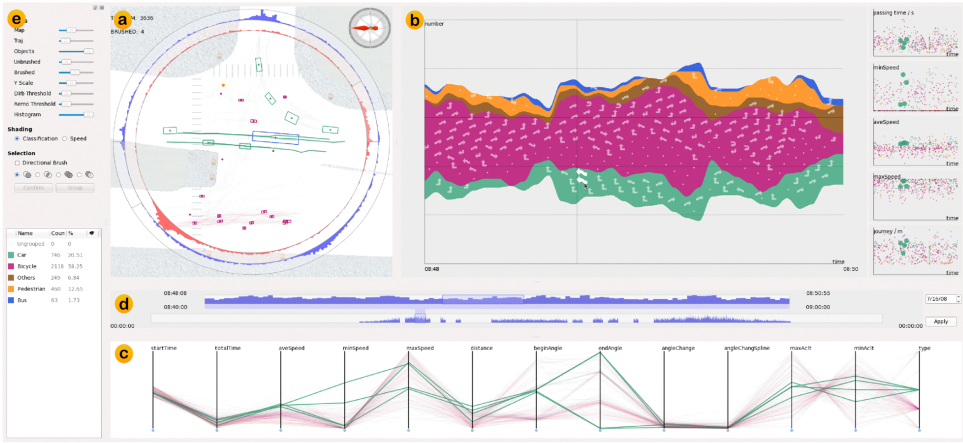


FIGURE 9. TripVista overview [74].

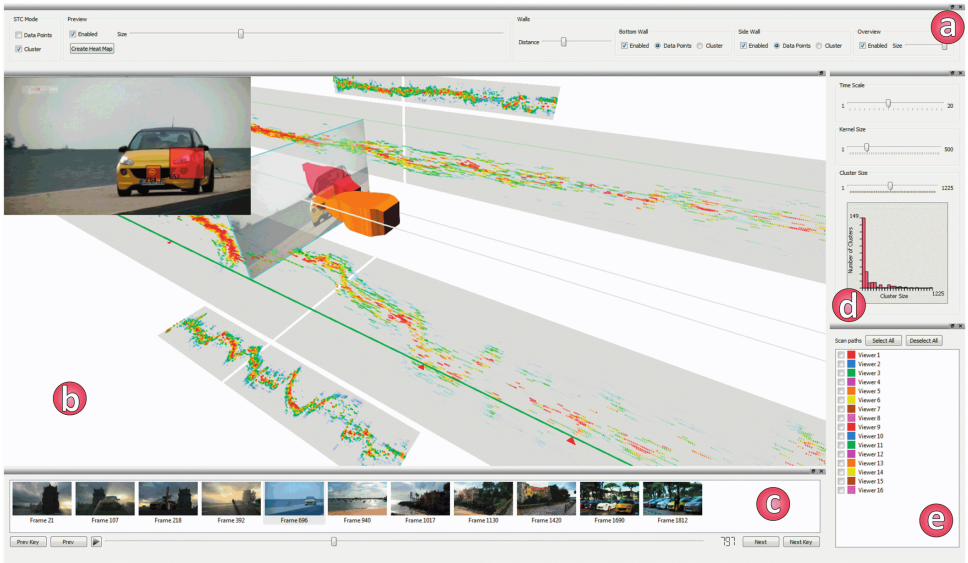


FIGURE 10. A *multi-dimensional-to-3D* visual-analytics application for eye-tracking data [77].

which included data and view specifications (filtering, sorting, deriving values or models from source data, etc.), view manipulations (selecting, navigation, etc.), and processes and provenances (recording, guiding or sharing, etc.). Endert *et al.* [80] divided interactions into two categories *exploratory* and *expressive* from observation-level.

In this survey, the taxonomy of [79] and the classifications of [80] are combined to classify visual-analytics applications from the interaction perspective. Visual-analytics applications are classified into two categories: *exploratory-oriented* and *expressive-oriented*, based on their interactions. An application will be classified as *exploratory-oriented* if its interactions are designed to explore data and visualization space. For example, the interactions of selecting different encoding, modifying zoom levels and of filtering data are considered as *exploratory-oriented*. Within *exploratory-oriented* visual-analytics applications, users gain insight from data by observing how data reacts during interactions in a dynamic visual representation.

An application will be classified as *expressive-oriented* if its interactions are designed to change the algorithms for rendering the visualization or the underlying models for data analysis. The interactions of modifying the parameters of the underlying mathematical models or rendering algorithms,

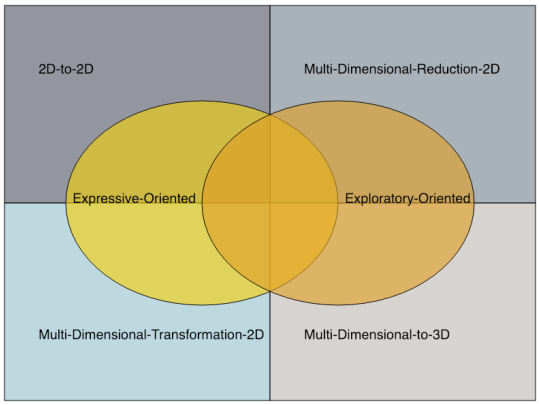


FIGURE 11. The complete classification of visual-analytics applications.

and deriving values or models from source data, are considered as *expressive-oriented*. Within *expressive-oriented* visual-analytics applications, interactions are therefore commonly coupled with the statistics-based data-analysis process. For example, Interactive Principal Component Analysis (iPCA) [30] changes the weight for each dimension in calculating the direction of projection using multiple sliders through user interactions. Also, for visual-analytics applications using MDS [81], the dissimilarities in the calculation of the stress function can be weighted through visual controls.

C. A COMPLETE CATEGORIZATION OF VISUAL-ANALYTICS APPLICATIONS

For the visualization-based classification of a visual-analytics application, 3D data are not considered for two reasons: 1) 3D data can be naturally classified as multi-dimensional data, and 2) 3D data can be easily visualized in several 2D visualizations. Therefore, it is not necessary to create a category for 3D data. Furthermore, both types of interaction can be used by an application at the same time. Accordingly, there is an overlap between the categories of *exploratory-oriented* and *expressive-oriented* in the interaction-based classification scheme.

From a technical perspective, visualization- and interaction-based classifications form a complete categorization of visual-analytics applications. Figure 11 illustrates the relationship of two classifications of visual-analytics applications. It covers all technical components of state-of-the-art visual-analytics applications, including 2D- and 3D-visualization techniques, algorithmic dimension reduction and data-analysis methods, and exploratory and expressive interactions. Therefore, this categorization can direct researchers toward selecting the appropriate techniques for applying visual analytics and building an application on complex data sets. Table 3 shows the categorizations of the visual-analytics applications examined in this survey.

IV. VISUAL ANALYTICS: CHALLENGES AND FUTURE DIRECTIONS

Visual analytics has made great progress over the past 15 years. The inevitable trend of visual analytics brought us not only opportunities but also challenges. In this section, these challenges characterized by the scalability, interaction, infrastructure, and evaluation from both technical and application perspectives are discussed. In addition, the future directions accompanying these challenges in an effort to provide a stimulus for research are presented.

A. CHALLENGES

1) SCALABILITY

The explosion of data presents a significant challenge for exploring large and complex data sets. Visual-analytics techniques need to be able to scale with both the size and dimension of the data. However, there is a growing mismatch between data size/complexity and the human ability to explore and interact with the data [144], which makes scalability a fundamental challenge of visual analytics.

The scalability of visual analytics is defined as "its capability to effectively display large data sets in terms of either the number or the dimension of individual data elements" [145]. Presently, most research in improving the scalability of visual analytics is primarily focused on investigating visualization devices [146]. For example, with the growing availability of large-scale high-resolution displays, large high-resolution displays [147], [148] and power wall display [149] have been investigated to display more overview and detail for large data sets in visual-analytics research. However, compared with the amount of data which is continuously growing at a rapid pace, the number of pixels on current displays has remained rather constant. In this case, the amount of data still commonly exceeds the limited amount of pixels of a display by several orders of magnitude. In addition, although it is possible to build ever-larger and higher-resolution displays, human visual acuity is limited to match the extreme large-screen approach. Meantime, algorithmic dimension-reduction techniques have been investigated to improve the scalability of visual analytics, especially for multi-dimensional data sets. For example, both linear and non-linear dimension-reduction algorithms, such as PCA [150] and MDS [151], have been applied to visualize multi-dimensional data sets. However, the use of these algorithms has been somewhat limited in visual analytics because they have been too slow for interactive use when the number of dimensions is scaled up [144]. This significantly hinders the integration of human judgment into the data-analysis process. More importantly, more dimension reduction and a higher rate of compression of data on displays mean more abstract representations and more lost details [152], which requires additional interpretation when performing analytical reasoning.

The scalability challenge of visual analytics involves both human and machine limitations. It is expected that the

TABLE 3. Complete categories of the examined papers.

Category		Paper
2D-to-2D	exploratory-oriented	[68] [82] [83] [84] [21]
	expressive-oriented	[85] [67]
multi-dimensional-reduction-2D	exploratory-oriented	[86] [87] [71] [88] [89] [90] [72] [91] [92] [93] [94] [31]
	expressive-oriented	[81] [30] [71] [95] [96] [97]
multi-dimensional-transformation-2D	exploratory-oriented	[98] [99] [100] [101] [102] [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [66] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [65] [128] [129] [130] [131] [132] [133] [134] [135] [136]
	expressive-oriented	[137] [138] [74] [139] [140] [116] [126] [141]
multi-dimensional-to-3D	exploratory-oriented	[111] [113] [77] [142] [143]
	expressive-oriented	[77] [142]

integration of algorithms and visualization techniques for large data in visual analytics can help reduce the mismatch between data size/complexity and human ability.

2) INTERACTION

Interaction is a fundamental component of visual analytics. The grand challenge of interaction is to develop a taxonomy to describe and clarify the interaction design space since there is hardly ever an explanation of what the benefits of interaction actually are as well as how and why they work [19], [153]. There are several taxonomies [154]–[156] that have been devised for describing and structuring interaction space. However, it is still a challenge to develop a comprehensive taxonomy that captures all possible interactions that may be performed, which includes an explication of the cognitive and perceptual impact of each individual interaction [157].

In this survey, interaction methods in visual analytics are classified into two categories: *exploratory-oriented* and *expressive-oriented*. Both of them are equally important to visual analytics. However, according to Table 3, compared with *exploratory-oriented* interactions, *expressive-oriented* interactions are used much less in recent visual-analytics applications. Only a few applications have tried to use these two different kinds of interactions together, such as [71], [77], [110], [142]. One of most possible reasons for this situation is that *expressive-oriented* interactions are associated with the modification of the underlying mathematical models or rendering algorithms, which may delay the response of the interactions when the size and complexity (dimension) of the input data are scaled up.

In addition, there have been rapid advances in interaction technologies; however, their advantages have not been fully investigated as most visual-analytics applications are still based on the traditional desktop, mouse, and keyboard setup of WIMP (Windows, Icons, Menus, and a Pointer) interfaces [158]. A few researchers have focused on new possibilities in interaction technologies in visual analytics; however, they have only been tested with simple data sets and scenarios. For example, PaperLens [159] uses a handheld lens and a tracked sheet of paper to navigate the 3D virtual information spaces above a tabletop. Interactive Whiteboards [160] leverages hand-drawn input for exploring data through simple charts. Ball and North [147] discussed embodied interactions, such as physical navigation, by physically interacting with large-scale visualizations for improving performance times on analytics tasks through an empirical study.

Therefore, in visual analytics, the challenge of interaction is to investigate its cognitive and perceptual impacts for integrating human judgment in the data-analysis process, as well as developing novel interactions by taking advantage of new algorithms and devices.

3) INFRASTRUCTURE

Based on the observation gained in this survey, in the field of visual analytics, there is an urgent need for a common framework to accelerate the research and development of new techniques. This has neither been fully valued nor discussed in recent research.

A few frameworks have been proposed for various purposes in visual analytics. For example, Aigner *et al.* [161] proposed a conceptual visual-analytics framework specifically for time and time-oriented data. Garg *et al.* [162] describe a visual analytic framework which uses logic programming as the underlying computing machinery to encode the relations as rules and facts and compute with them. Chen *et al.* [95], Brennan *et al.* [99] and Aragon *et al.* [163] proposed three frameworks for collaborative visual analytics with different focuses. However, these frameworks were designed for a specific domain/problem or data type. None of them can be reused as a common framework, which hinders the rapid development of visual-analytics techniques, and communications in the visual-analytics research community.

More importantly, the lack of a visual-analytics framework that works on high-performance computing platforms, such as Elasticsearch [164], Apache Kafka [165], and Apache Spark [166], is especially frustrating for visual analytics of large-scale data.

Therefore, in visual analytics, the infrastructure challenge is to develop reusable libraries and frameworks for common research questions, such as heterogeneous-data fusion, collaborative analytics, information sharing, and large-scale data processing, to accelerate the research of visual analytics, and facilitate communications in the research community. Such libraries and frameworks must support multiple levels of abstraction, including unwrapping the logic within the products, adding new reasoning and facts, and turning the results into new products.

4) EVALUATION

As cognition, perception and analytical reasoning are significant factors in the visual-analytics process, human information discourse constitutes a challenge for evaluating the utility, effectiveness, and trustworthiness of visual-analytics applications. Some methods have been investigated for evaluating visual-analytics applications, for example, insight- and task-based methodologies for evaluating spatiotemporal visual-analytics applications [167]. However, in various application domains, the complexity of a visual-analytics application still makes its evaluation a challenge.

For visual-analytics applications in different problem domains, such as biology, medical, astrophysics, and geography, three methods that adapted from the field of information visualization are used, including case studies, user studies based on controlled experiments, and expert reviews. However, each of these methods has its own strengths and weaknesses. For example, Tory and Moller [168] indicated that expert reviews can quickly assess usability, however, they may miss important issues in their evaluations due to a lack of user involvement. More importantly, these methods are mainly focused on evaluating the usability and effectiveness of the visualization components of visual-analytics applications, which lack an evaluation of the data analysis components, such as accuracy and efficiency.

In addition, during the visual-analytics process, the uncertainties in data may arise, propagate and compound, which results in impaired decision making, misleading analysis results, and misinterpretations [169]. This challenges the trustworthiness of visual-analytics applications, which is one of the most important evaluation criteria. Sacha *et al.* [170] illustrated the relationship between human's perceptual and cognitive biases and the trustworthiness of visual-analytics applications, in which the user's awareness of the uncertainties in the data is influenced by their perceptual and cognitive biases. Presently, techniques, such as uncertainty modeling and visualization, have been proposed to quantitatively characterize and intuitively display the uncertainty information in data sets [171], [172]. However, due to the complexity of the visual-analytics process, there are still no widely accepted evaluation techniques to ensure the trustworthiness of the visual-analytics process.

Therefore, we need science, support structures and data to perform encompassing evaluations of visual-analytics applications. The challenge of proposing a theoretically founded evaluation framework for visual analytics is expected to gain more interest in the field.

B. FUTURE DIRECTIONS

In spite of all the challenges, the rapid development of visual analytics will lead to numerous opportunities for making progress in many fields. Several future directions are discussed in this section to tackle many challenges and open issues with visual analytics.

To address the scalability challenge of visual analytics, investigating novel visualization algorithms and methods for large-scale data is one significant research direction. In the field of information visualization, most methods are focused on relatively small data sets. For example, various studies [173]–[175] on PCPs for visualizing high-dimensional data are limited when the size of the data is scaled up. Therefore, re-designing these methods specifically for large-scale data would be a potential solution for visual analytics of large-scale data. In addition, since there are no strict boundaries among the proposed categorization of visual-analytics applications, combining techniques from different categories is a potential research direction. For example, combining algorithmic dimension-reduction methods and parallel coordinates would be a possible way for visual analytics of high-dimensional data.

To facilitate collaboration and information sharing in visual analytics, building a web-based framework for visual analytics is a potential research direction. A web-based framework could break temporal and spatial constraints in communication and collaboration. Moreover, it could also facilitate the integration of visual-analytics applications with other big data platforms, since most recent big-data platforms provide web services for accessing and processing the data stored within them [176]. This will not only address the scalability challenge of visual analytics but also will accelerate the research and development of visual analytics. Another future research direction of visual analytics is to extend it into immersive and stereoscopic visualization (virtual reality) environments. Although several devices, such as consumer-grade 3D displays and immersive head-mounted displays enable immersive and stereoscopic visualization environments, the related visualization techniques have not been explored extensively for information visualization and visual analytics [177]. Investigating these new devices and related visualization techniques could provide potential solutions that address the scalability and interaction challenges of visual analytics. In addition, to address the evaluation challenges, developing evaluation standards for visual analytics by selecting and combining proper evaluation methods from the fields of visualization and algorithmic data analysis [178] is another possible direction.

The challenges and future directions discussed in this section were selected based on the observations made in this survey. For the entire field of visual analytics, there are

many more challenges and opportunities than what have been discussed in this section. Readers are hence encouraged to use these as guides to deeper investigation and prospective thinking toward future possibilities of visual analytics.

V. SUMMARY

Visual analytics is a fast-growing field of research combining strengths from visualization, data analysis, knowledge discovery, data management, analytical reasoning, cognition, perception, and human-computer interaction. Its goal is to discover knowledge and gain insight from large and complex data sets through integrating human judgment into the data-analysis process.

This survey has drawn a complete picture of visual analytics to direct future research by examining the related research in various application domains. To avoid limiting this survey to a specific data type or applications domain, a novel categorization of visual analytics applications from a technical perspective was proposed. Based on this categorization, an organized overview of visual analytics in over 200 publications was constructed, which discussed the theory, evolution, and trends of visual analytics, and how visual analytics is applied in various application domains was investigated. To better understand visual analytics, the human-information discourse of visual analytics was discussed, a formal model of the visual analytics process was summarized, which provided a detailed definition of visual analytics, and the visual analytics mantra "Analyze/Overview first, interaction and visualization repeatedly, insights in data" was presented. Under the proposed categorization, state-of-the-art techniques and applications of visual analytics in different application domains that can bridge the gap between the challenges of discovering knowledge in large and complex data sets and visual analytics solutions were presented. Finally, an overview of the major challenges and future directions of visual analytics was given.

REFERENCES

[1] N. Al-Qirim, A. Tarhini, and K. Rouibah, "Determinants of big data adoption and success," in *Proc. Int. Conf. Algorithms, Comput. Syst.*, 2017, pp. 88–92.

[2] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, "Visual analytics: Scope and challenges," in *Visual Data Mining* (Lecture Notes in Computer Science), vol. 4404, S. J. Simoff, M. H. Böhlen, and A. Mazeika, Eds. Berlin, Germany: Springer, 2008.

[3] J. J. Thomas and K. A. Cook, "A visual analytics agenda," *IEEE Comput. Graph. Appl.*, vol. 26, no. 1, pp. 10–13, Jan. 2006.

[4] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, Eds., *Mastering the Information Age: Solving Problems With Visual Analytics*. Goslar, Germany: Eurographics Association, 2010.

[5] C. C. Yang, H. Chen, and K. Hong, "Visualization of large category map for Internet browsing," *Decis. Support Syst.*, vol. 35, no. 1, pp. 89–102, 2003.

[6] A. J. Hey and A. E. Trefethen, "The data deluge: An e-science perspective," in *Grid Computing: Making the Global Infrastructure a Reality*, F. Berman, G. C. Fox, and A. J. G. Hey, Eds. Hoboken, NJ, USA: Wiley, 2003, pp. 809–824.

[7] D. A. Keim, F. Mansmann, D. Oelke, and H. Ziegler, "Visual analytics: Combining automated discovery with interactive visualizations," in *Proc. Int. Conf. Discovery Sci.* Springer, 2008, pp. 2–14.

[8] D. S. Ebert, J. Dill, and D. J. Kasik, "In memoriam: Illuminating our paths–James (Jim) Joseph Thomas," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2010, pp. 1–14.

[9] J. Kielman, J. Thomas, and R. May, "Foundations and frontiers in visual analytics," *Inf. Vis.*, vol. 8, no. 4, pp. 239–246, 2009.

[10] C. Chen, H. Hou, Z. Hu, and S. Liu, "An illuminated path: The impact of the work of Jim Thomas," in *Expanding the Frontiers of Visual Analytics and Visualization*, J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. Wong, Eds. London, U.K.: Springer, 2012.

[11] M. C. F. D. Oliveira and H. Levkowitz, "From visual data exploration to visual data mining: A survey," *IEEE Trans. Vis. Comput. Graphics*, vol. 9, no. 3, pp. 378–394, Jul. 2003.

[12] M. Card, *Readings in Information Visualization: Using Vision to Think*. San Mateo, CA, USA: Morgan Kaufmann, 1999.

[13] S. Card, J. D. Mackinlay, and B. Shneiderman, "Information visualization," in *Human-Computer Interaction: Design Issues, Solutions, and Applications*, vol. 181. London, U.K.: Taylor & Francis, 2009.

[14] G. M. Nielson, H. Hagen, and H. Müller, *Scientific Visualization: Overviews, Methodologies, and Techniques*. Los Alamitos, CA, USA: IEEE Computer Society, 1997.

[15] A. Dix, "Human-computer interaction," in *Encyclopedia of Database Systems*, L. Liu and M. T. ÖZsu, Eds. Boston, MA, USA: Springer, 2009.

[16] A. Azzalini and B. Scarpa, *Data Analysis and Data Mining: An Introduction*. New York, NY, USA: Oxford Univ. Press, 2012.

[17] A. O'Hagan and J. J. Forster, *Kendall's Advanced Theory of Statistics: Bayesian Inference*, vol. 2B, 2nd ed. London, U.K.: Arnold, 2004, p. 496.

[18] S. J. Simoff, M. H. Böhlen, and A. Mazeika, "Visual data mining: An introduction and overview," in *Visual Data Mining* (Lecture Notes in Computer Science), vol. 4404, S. J. Simoff, M. H. Böhlen, and A. Mazeika, Eds. Berlin, Germany: Springer, 2008.

[19] K. A. Cook and J. J. Thomas, "Illuminating the path: The research and development agenda for visual analytics," Pacific Northwest Nat. Lab., Richland, WA, USA, Tech. Rep. PNNL-SA-45230, 2005.

[20] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, "Visual analytics: Definition, process, and challenges," in *Information Visualization* (Lecture Notes in Computer Science), vol. 4950, A. Kerren, J. T. Stasko, J. D. Fekete, and C. North, Eds. Berlin, Germany: Springer, 2008.

[21] N. Andrienko and G. Andrienko, "Visual analytics of movement: An overview of methods, tools and procedures," *Inf. Vis.*, vol. 12, no. 1, pp. 3–24, 2012.

[22] J. J. Caban and D. Gotz, "Visual analytics in healthcare—Opportunities and research challenges," *J. Amer. Med. Inform. Assoc.*, vol. 22, no. 2, pp. 260–262, Mar. 2015. doi: 10.1093/jamia/ocv006.

[23] V. L. West, D. Borland, and W. E. Hammond, "Innovative information visualization of electronic health record data: A systematic review," *J. Amer. Med. Inform. Assoc.*, vol. 22, no. 2, pp. 330–339, 2014.

[24] G. Andrienko, N. Andrienko, U. Demsar, D. Dransch, J. Dykes, S. I. Fabrikant, M. Jern, M.-J. Kraak, H. Schumann, and C. Tominski, "Space, time and visual analytics," *Int. J. Geograph. Inf. Sci.*, vol. 24, no. 10, pp. 1577–1600, 2010.

[25] L. Zhang, A. Stoffel, M. Behrisch, S. Mittelstadt, T. Schreck, R. Pompl, S. Weber, H. Last, and D. Keim, "Visual analytics for the big data era—A comparative review of state-of-the-art commercial systems," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2012, pp. 173–182.

[26] J. R. Harger and P. J. Crossno, "Comparison of open-source visual analytics toolkits," *Proc. SPIE*, vol. 8294, Jan. 2012, Art. no. 82940E.

[27] G. Andrienko, N. Andrienko, I. Kopanakis, A. Ligtenberg, and S. Wrobel, "Visual analytics methods for movement data," in *Mobility, Data Mining and Privacy*, F. Giannotti and D. Pedreschi, Eds. Berlin, Germany: Springer, 2008.

[28] G.-D. Sun, Y.-C. Wu, R.-H. Liang, and S.-X. Liu, "A survey of visual analytics techniques and applications: State-of-the-art research and future challenges," *J. Comput. Sci. Technol.*, vol. 28, no. 5, pp. 852–867, 2013.

[29] Y. Wu, N. Cao, D. Gotz, Y.-P. Tan, and D. A. Keim, "A survey on visual analytics of social media data," *IEEE Trans. Multimedia*, vol. 18, no. 11, pp. 2135–2148, Nov. 2016.

[30] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang, "iPCA: An interactive system for PCA-based visual analytics," *Comput. Graph. Forum*, vol. 28, no. 3, pp. 767–774, 2009.

[31] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins, "Progressive learning of topic modeling parameters: A visual analytics framework," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 382–391, Aug. 2018.

[32] P. C. Wong and J. Thomas, "Guest editors' introduction–visual analytics," *IEEE Comput. Graph. Appl.*, vol. 24, no. 5, pp. 20–21, Sep. 2004.
[33] D. A. Keim, F. Mansmann, A. Stoffel, and H. Ziegler, "Visual analytics," in *Encyclopedia of Database Systems*, L. Liu and M. T. ÖZsu, Eds. Boston, MA, USA: Springer, 2009.
[34] T.-M. Rhyne, M. Tory, T. Munzner, M. Ward, C. Johnson, and D. H. Laidlaw, "Information and scientific visualization: Separate but equal or happy together at last," in *Proc. 14th IEEE Vis. (VIS)*, Oct. 2003, p. 115.
[35] M. Friendly and D. J. Denis. (2001). *Milestones in the History of Thematic Cartography, Statistical Graphics, and Data Visualization*. [Online]. Available: http://www.datavis.ca/milestones
[36] B. B. Bederson and B. Shneiderman, *The Craft of Information Visualization: Readings and Reflections*. San Mateo, CA, USA: Morgan Kaufmann, 2003.
[37] R. Spence, *Information Visualization: An Introduction*, 3rd ed. Springer, 2014.
[38] J. Johansson and C. Forsell, "Evaluation of parallel coordinates: Overview, categorization and guidelines for future research," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 579–588, Aug. 2016.
[39] B. Shneiderman, "Tree visualization with tree-maps: A 2-D space-filling approach," *ACM Trans. Graph.*, vol. 11, no. 1, pp. 92–99, 1992.
[40] 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," in *Proc. Eurographics*, 2013, pp. 39–63.
[41] D. A. Keim, "Designing pixel-oriented visualization techniques: Theory and applications," *IEEE Trans. Vis. Comput. Graphics*, vol. 6, no. 1, pp. 59–78, Jan. 2000.
[42] J. W. Tukey, "We need both exploratory and confirmatory," *Amer. Stat.*, vol. 34, no. 1, pp. 23–25, 1980.
[43] A. Gelman, "Exploratory data analysis for complex models," *J. Comput. Graph. Statist.*, vol. 13, no. 4, pp. 755–779, 2004.
[44] M. L. Vigni, C. Durante, and M. Cocchi, "Exploratory data analysis," in *Data Handling in Science and Technology*, vol. 28. Amsterdam, The Netherlands: Elsevier, 2013, pp. 55–126.
[45] J. W. Tukey, *Exploratory Data Analysis*. Reading, MA, USA: Addison-Wesley, 1977.
[46] T. M. Nguyen, A. M. Tjoa, and J. Trujillo, "Data warehousing and knowledge discovery: A chronological view of research challenges," in *Data Warehousing and Knowledge Discovery* (Lecture Notes in Computer Science), vol. 3589, A. M. Tjoa and J. Trujillo, Eds. Berlin, Germany: Springer, 2005.
[47] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Mag.*, vol. 17, no. 3, p. 37, 1999.
[48] D. A. Keim, F. Mansmann, and J. Thomas, "Visual analytics: How much visualization and how much analytics?" *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 2, pp. 5–8, 2010.
[49] M. Ankerst, "Visual data mining," Ph.D. dissertation, 2001, pp. 1–216.
[50] S. J. Simoff, M. H. Böhlen, and A. Mazeika, Eds., *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. Heidelberg, Germany: Springer-Verlag, 2008.
[51] K. Sedig and P. Parsons, "Interaction design for complex cognitive activities with visual representations: A pattern-based approach," *AIS Trans. Hum.-Comput. Interact.*, vol. 5, no. 2, pp. 84–133, 2013.
[52] M. Pohl, M. Smuc, and E. Mayr, "The user puzzle—Explaining the interaction with visual analytics systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2908–2916, Dec. 2012.
[53] W. Karwowski, Ed., *International Encyclopedia of Ergonomics and Human Factors*, vol. 3. Boca Raton, FL, USA: CRC Press, 2001.
[54] P. Rheingans, "Are we there yet? Exploring with dynamic visualization," *IEEE Comput. Graph. Appl.*, vol. 22, no. 1, pp. 6–10, Jan. 2002.
[55] M. Tory and T. Moller, "Human factors in visualization research," *IEEE Trans. Vis. Comput. Graphics*, vol. 10, no. 1, pp. 72–84, Jan./Feb. 2004.
[56] T. M. Green, W. Ribarsky, and B. Fisher, "Visual analytics for complex concepts using a human cognition model," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2008, pp. 91–98.
[57] S. Miksch and W. Aigner, "A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data," *Comput. Graph.*, vol. 38, pp. 286–290, Feb. 2014.
[58] A. Dasgupta, J.-Y. Lee, R. Wilson, R. A. Lafrance, N. Cramer, K. Cook, and S. Payne, "Familiarity vs trust: A comparative study of domain scientists' trust in visual analytics and conventional analysis methods," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 271–280, Aug. 2017.
[59] A. Endert, M. S. Hossain, N. Ramakrishnan, C. North, P. Fiaux, and C. Andrews, "The human is the loop: New directions for visual analytics," *J. Intell. Inf. Syst.*, vol. 43, no. 3, pp. 411–435, 2014.
[60] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proc. IEEE Symp. Vis. Lang.*, Sep. 1996, pp. 336–343.
[61] A. Kerren and F. Schreiber, "Toward the role of interaction in visual analytics," in *Proc. IEEE Winter Simulation Conf. (WSC)*, Dec. 2012, pp. 1–13.
[62] J. J. van Wijk, "The value of visualization," in *Proc. IEEE Vis. (VIS)*, Oct. 2005, pp. 79–86.
[63] X. Wang, D. H. Jeong, W. Dou, S.-W. Lee, W. Ribarsky, and R. Chang, "Defining and applying knowledge conversion processes to a visual analytics system," *Comput. Graph.*, vol. 33, no. 5, pp. 616–623, 2009.
[64] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, "Knowledge generation model for visual analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1604–1613, Dec. 2014.
[65] S. Chen, C. Wang, Z. Liu, Z. Wang, Z. Wang, Z. Miao, and X. Yuan, "Visual analytics support for collecting and correlating evidence for intelligence analysis," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2014, pp. 319–320.
[66] G. Andrienko, N. Andrienko, H. Bosch, T. Ertl, G. Fuchs, P. Jankowski, and D. Thom, "Thematic patterns in georeferenced tweets through space-time visual analytics," *Comput. Sci. Eng.*, vol. 15, no. 3, pp. 72–82, May 2013.
[67] M. Bögl, W. Aigner, P. Filzmoser, T. Lammarsch, S. Miksch, and A. Rind, "Visual analytics for model selection in time series analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2237–2246, Dec. 2013.
[68] G. Andrienko, N. Andrienko, and S. Wrobel, "Visual analytics tools for analysis of movement data," *ACM SIGKDD Explor. Newslett.*, vol. 9, no. 2, pp. 38–46, 2007.
[69] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," in *Proc. Eurograph. Conf. Vis.*, 2015, pp. 1115–1127.
[70] C. Donalek, S. G. Djorgovski, S. Davidoff, A. Cioc, A. Wang, G. Longo, J. S. Norris, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, and A. Drake, "Immersive and collaborative data visualization using virtual reality platforms," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2014, pp. 609–614.
[71] J. Choo, H. Lee, J. Kihm, and H. Park, "iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2010, pp. 27–34.
[72] Y. Wu, S. Liu, K. Yan, M. Liu, and F. Wu, "OpinionFlow: Visual analysis of opinion diffusion on social media," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1763–1772, Dec. 2014.
[73] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky, "Guidelines for using multiple views in information visualization," in *Proc. Work. Conf. Adv. Vis. Interfaces*, 2000, pp. 110–119.
[74] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan, "TripVista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection," in *Proc. IEEE Pacific Vis. Symp. (PacificVis)*, Mar. 2011, pp. 163–170.
[75] E. Achtert, H.-P. Kriegel, E. Schubert, and A. Zimek, "Interactive data mining with 3D-parallel-coordinate-trees," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 1009–1012.
[76] J. Johansson, M. Cooper, and M. Jern, "3-dimensional display for clustered multi-relational parallel coordinates," in *Proc. 9th Int. Conf. Inf. Vis.*, Jul. 2005, pp. 188–193.
[77] K. Kurzhals and D. Weiskopf, "Space-time visual analytics of eye-tracking data for dynamic stimuli," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2129–2138, Dec. 2013.
[78] A. Endert, "Semantic interaction for visual analytics: Toward coupling cognition and computation," *IEEE Comput. Graph. Appl.*, vol. 34, no. 4, pp. 8–15, Jul. 2014.
[79] J. Heer and B. Shneiderman, "Interactive dynamics for visual analysis," *Queue*, vol. 10, no. 2, p. 30, 2012.
[80] A. Endert, C. Han, D. Maiti, L. House, and C. North, "Observation-level interaction with statistical models for visual analytics," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2011, pp. 121–130.

[81] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen, "Data visualization with multidimensional scaling," *J. Comput. Graph. Statist.*, vol. 17, no. 2, pp. 444–472, 2012.

[82] A. Savikhin, R. Maciejewski, and D. S. Ebert, "Applied visual analytics for economic decision-making," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2008, pp. 107–114.

[83] S. Afzal, R. Maciejewski, and D. S. Ebert, "Visual analytics decision support environment for epidemic modeling and response evaluation," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2011, pp. 191–200.

[84] C. Rohrdantz, A. Hautli, T. Mayer, M. Butt, D. A. Keim, and F. Plank, "Towards tracking semantic change by visual analytics," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, 2011, pp. 305–310.

[85] S. Rudolph, A. Savikhin, and D. S. Ebert, "Finvis: Applied visual analytics for personal financial planning," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2009, pp. 195–202.

[86] L. Wilkinson, A. Anand, and R. Grossman, "High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 6, pp. 1363–1372, Nov. 2006.

[87] P. C. Wong, H. Foote, G. Chin, P. Mackey, and K. Perrine, "Graph signatures for visual analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 6, pp. 1399–1413, Nov. 2006.

[88] H. Liu, Y. Gao, L. Lu, S. Liu, H. Qu, and L. M. Ni, "Visual analysis of route diversity," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2011, pp. 171–180.

[89] A. Endert, P. Fiaux, and C. North, "Semantic interaction for visual text analytics," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2012, pp. 473–482.

[90] N. Andrienko, G. Andrienko, H. Stange, T. Liebig, and D. Hecker, "Visual analytics for understanding spatial situations from episodic movement data," *Künstliche Intell.*, vol. 26, no. 3, pp. 241–251, 2012.

[91] J. Wang, L. Bradel, and C. North, "Event-based text visual analytics," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2014, pp. 333–334.

[92] J. Chae, G. Wang, B. Ahlbrand, M. B. Gorantla, J. Zhang, S. Chen, H. Xu, J. Zhao, W. Hatton, A. Malik, S. Ko, and D. S. Ebert, "Visual analytics of heterogeneous data for criminal event analysis VAST challenge 2015: Grand challenge," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2015, pp. 149–150.

[93] W. Hatton, J. Zhao, M. B. Gorantla, J. Chae, B. Ahlbrand, H. Xu, S. Chen, G. Wang, J. Zhang, A. Malik, S. Ko, and D. S. Ebert, "Visual analytics for detecting communication patterns," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2015, pp. 137–138.

[94] S. Liu, J. Yin, X. Wang, W. Cui, K. Cao, and J. Pei, "Online visual analytics of text streams," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 11, pp. 2451–2466, Dec. 2016.

[95] Y. Chen, J. Alsakran, S. Barlowe, J. Yang, and Y. Zhao, "Supporting effective common ground construction in asynchronous collaborative visual analytics," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2011, pp. 101–110.

[96] A. Karami, "A framework for uncertainty-aware visual analytics in big data," in *Proc. AIC*, 2015, pp. 146–155.

[97] N. Pezzotti, B. P. F. Lelieveldt, L. van der Maaten, T. Höllt, E. Eisemann, and A. Vilanova, "Approximated and user steerable tSNE for progressive visual analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 7, pp. 1739–1752, May 2017.

[98] Z. Hu, J. Mellor, J. Wu, and C. DeLisi, "VisANT: An online visualization and analysis tool for biological interaction data," *BMC Bioinf.*, vol. 5, no. 1, p. 17, 2004.

[99] S. E. Brennan, K. Mueller, G. Zelinsky, I. Ramakrishnan, D. S. Warren, and A. Kaufman, "Toward a multi-analyst, collaborative framework for visual analytics," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, Oct. 2006, pp. 129–136.

[100] P. E. Keel, "Collaborative visual analytics: Inferring from the spatial organization and collaborative use of information," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, Oct. 2006, pp. 137–144.

[101] M. C. Hao, U. Dayal, D. A. Keim, D. Morent, and J. Schneidewind, "Intelligent visual analytics queries," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2007, pp. 91–98.

[102] R. Santamaría, R. Therón, and L. Quintales, "A visual analytics approach for understanding biclustering results from microarray data," *BMC Bioinf.*, vol. 9, no. 1, p. 247, 2008.

[103] J.-W. Ahn and P. Brusilovsky, "Adaptive visualization of search results: Bringing user models to visual analytics," *Inf. Vis.*, vol. 8, no. 3, pp. 167–179, 2009.

[104] J. R. Goodall and M. Sowul, "VIAssist: Visual analytics for cyber defense," in *Proc. IEEE Conf. Technol. Homeland Secur. (HST)*, May 2009, pp. 143–150.

[105] N. Diakopoulos, M. Naaman, and F. Kivran-Swaine, "Diamonds in the rough: Social media visual analytics for journalistic inquiry," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2010, pp. 115–122.

[106] R. Maciejewski, S. Rudolph, R. Hafen, A. Abusalah, M. Yakout, M. Ouzzani, W. S. Cleveland, S. J. Grannis, and D. S. Ebert, "A visual analytics approach to understanding spatiotemporal hotspots," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 2, pp. 205–220, Mar. 2010.

[107] P. Isenberg, D. Fisher, M. R. Morris, K. Inkpen, and M. Czerwinski, "An exploratory study of co-located collaborative visual analytics around a tabletop display," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2010, pp. 179–186.

[108] S. Glaßer, U. Preim, K. Tönnies, and B. Preim, "A visual analytics approach to diagnosis of breast DCE-MRI data," *Comput. Graph.*, vol. 34, no. 5, pp. 602–611, 2010.

[109] T. D. Wang, K. Wongsuphasawat, C. Plaisant, and B. Shneiderman, "Extracting insights from electronic health records: Case studies, a visual analytics process model, and design recommendations," *J. Med. Syst.*, vol. 35, no. 5, pp. 1135–1152, 2011.

[110] A. Malik, R. Maciejewski, B. Maule, and D. S. Ebert, "A visual analytics process for maritime resource allocation and risk assessment," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2011, pp. 221–230.

[111] G. Andrienko, N. Andrienko, M. Burch, and D. Weiskopf, "Visual analytics methodology for eye movement studies," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2889–2898, Dec. 2012.

[112] P. Isenberg, D. Fisher, S. A. Paul, M. R. Morris, K. Inkpen, and M. Czerwinski, "Co-located collaborative visual analytics around a tabletop display," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 5, pp. 689–702, Dec. 2012.

[113] T. Von Landesberger, S. Bremm, N. Andrienko, G. Andrienko, and M. Tekusova, "Visual analytics methods for categoric spatio-temporal data," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2012, pp. 183–192.

[114] K. K. Mane, C. Bizon, C. Schmitt, P. Owen, B. Burchett, R. Pietrobon, and K. Gersing, "VisualDecisionLinc: A visual analytics approach for comparative effectiveness-based clinical decision support in psychiatry," *J. Biomed. Inform.*, vol. 45, no. 1, pp. 101–106, 2012.

[115] F. Fischer, F. Mansmann, and D. A. Keim, "Real-time visual analytics for event data streams," in *Proc. 27th Annu. ACM Symp. Appl. Comput.*, 2012, pp. 801–806.

[116] A. Malik, R. Maciejewski, N. Elmqvist, Y. Jang, D. S. Ebert, and W. Huang, "A correlative analysis process in a visual analytics environment," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2012, pp. 33–42.

[117] N. A. Abousalh-Neto and S. Kazgan, "Big data exploration through visual analytics," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2012, pp. 285–286.

[118] C. A. Steed, D. M. Ricciuto, G. Shipman, B. Smith, P. E. Thornton, D. Wang, X. Shi, and D. N. Williams, "Big data visual analytics for exploratory earth system simulation analysis," *Comput. Geosci.*, vol. 61, pp. 71–82, Dec. 2013.

[119] N. Andrienko and G. Andrienko, "A visual analytics framework for spatio-temporal analysis and modelling," *Data Mining Knowl. Discovery*, vol. 27, no. 1, pp. 55–83, Jul. 2013.

[120] S. Liu, J. Pu, Q. Luo, H. Qu, L. M. Ni, and R. Krishnan, "VAIT: A visual analytics system for metropolitan transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1586–1596, Dec. 2013.

[121] P. A. Legg, D. H. S. Chung, M. L. Parry, R. Bown, M. W. Jones, I. W. Griffiths, and M. Chen, "Transformation of an uncertain video search pipeline to a sketch-based visual analytics loop," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2109–2118, Dec. 2013.

[122] B. Broeksema, T. Baudel, A. Telea, and P. Crisafulli, "Decision exploration lab: A visual analytics solution for decision management," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 1972–1981, Dec. 2013.

[123] J. Chae, D. Thom, Y. Jang, S. Y. Kim, T. Ertl, and D. S. Ebert, "Public behavior response analysis in disaster events utilizing visual analytics of microblog data," *Comput. Graph.*, vol. 38, pp. 51–60, Feb. 2014.

[124] D. Gotz and H. Stavropoulos, "DecisionFlow: Visual analytics for high-dimensional temporal event sequence data," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1783–1792, Dec. 2014.
[125] C. D. Stolper, A. Perer, and D. Gotz, "Progressive visual analytics: User-driven visual exploration of in-progress analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1653–1662, Dec. 2014.
[126] D. Sacha, M. Stein, T. Schreck, D. A. Keim, and O. Deussen, "Feature-driven visual analytics of soccer data," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2014, pp. 13–22.
[127] F. Chelaru, L. Smith, N. Goldstein, and H. C. Bravo, "Epiviz: Interactive visual analytics for functional genomics data," *Nature Methods*, vol. 11, no. 9, p. 938, 2014.
[128] F. Fischer, F. Stoffel, S. Mittelstädt, T. Schreck, and D. A. Keim, "Using visual analytics to support decision making to solve the Kronos incident (VAST challenge 2014)," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2014, pp. 301–302.
[129] Y. Zhao, Y. Peng, W. Huang, Y. Li, F. Zhou, Z. Liao, and K. Zhang, "A collaborative visual analytics of trajectory and transaction data for digital forensics: VAST 2014 mini-challenge 2: Award for outstanding visualization and analysis," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2014, pp. 371–372.
[130] S. Malik, F. Du, M. Monroe, E. Onukwugha, C. Plaisant, and B. Shneiderman, "Cohort comparison of event sequences with balanced integration of visual analytics and statistics," in *Proc. 20th Int. Conf. Intell. User Interfaces*, 2015, pp. 38–49.
[131] R. A. Leite, T. Gschwandtner, S. Miksch, E. Gstrein, and J. Kuntner, "Visual analytics for fraud detection and monitoring," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2015, pp. 201–202.
[132] R. C. Basole, A. Qamar, H. Park, C. J. J. Paredis, and L. F. McGinnis, "Visual analytics for early-phase complex engineered system design support," *IEEE Comput. Graph. Appl.*, vol. 35, no. 2, pp. 41–51, Mar. 2015.
[133] X. Huang, Y. Zhao, J. Yang, C. Zhang, C. Ma, and X. Ye, "Trajgraph: A graph-based visual analytics approach to studying urban network centralities using taxi trajectory data," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 160–169, Jan. 2016.
[134] Q. Li, P. Xu, Y. Y. Chan, Y. Wang, Z. Wang, H. Qu, and X. Ma, "A visual analytics approach for understanding reasons behind snowballing and comeback in MOBA games," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 211–220, Aug. 2017.
[135] M. Wagner, D. Slijepcevic, B. Horsak, A. Rind, M. Zeppelzauer, and W. Aigner, "KAVAGait: Knowledge-assisted visual analytics for clinical gait analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 3, pp. 1528–1542, Feb. 2018.
[136] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. Lelieveldt, E. Eisemann, and A. Vilanova, "DeepEyes: Progressive visual analytics for designing deep neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 98–108, Jan. 2018.
[137] E. J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre, "ClusterSculptor: A visual analytics tool for high-dimensional data," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2007, pp. 75–82.
[138] Z. Liao, Y. Yu, and B. Chen, "Anomaly detection in GPS data based on visual analytics," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2010, pp. 51–58.
[139] P. Federico, W. Aigner, S. Miksch, F. Windhager, and L. Zenk, "A visual analytics approach to dynamic social networks," in *Proc. 11th Int. Conf. Knowl. Manage. Knowl. Technol.*, 2011, p. 47.
[140] A. Savikhin, H. C. Lam, B. Fisher, and D. S. Ebert, "An experimental study of financial portfolio selection with visual analytics for decision support," in *Proc. 44th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2011, pp. 1–10.
[141] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl, "VA^2: A visual analytics approach for evaluating visual analytics applications," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 61–70, Aug. 2016.
[142] A. Motamedi, A. Hammad, and Y. Asen, "Knowledge-assisted BIM-based visual analytics for failure root cause detection in facilities management," *Autom. Construct.*, vol. 43, pp. 73–83, Jul. 2014.
[143] S. J. Rysavy, D. Bromley, and V. Daggett, "DIVE: A graph-based visual-analytics framework for big data," *IEEE Comput. Graph. Appl.*, vol. 34, no. 2, pp. 26–37, Mar. 2014.
[144] G. Robertson, D. Ebert, S. Eick, D. Keim, and K. Joy, "Scale and complexity in visual analytics," *Inf. Vis.*, vol. 8, no. 4, pp. 247–253, 2009.
[145] S. G. Eick and A. F. Karr, "Visual scalability," *J. Comput. Graph. Statist.*, vol. 11, no. 1, pp. 22–43, 2002.
[146] P. C. Wong, H.-W. Shen, C. R. Johnson, C. Chen, and R. B. Ross, "The top 10 challenges in extreme-scale visual analytics," *IEEE Comput. Graph. Appl.*, vol. 32, no. 4, pp. 63–67, Jul. 2012.
[147] R. Ball and C. North, "Realizing embodied interaction for visual analytics through large displays," *Comput. Graph.*, vol. 31, no. 3, pp. 380–400, 2007.
[148] D. A. Keim, T. Nietzschmann, N. Schelwies, J. Schneidewind, T. Schreck, and H. Ziegler, "A spectral visualization system for analyzing financial time series data," in *Proc. Eurograp./IEEE TCVG Symp. Vis.*, May 2006, pp. 195–202.
[149] H. Schmauder, M. Burch, C. Müller, and D. Weiskopf, "Distributed visual analytics on large-scale high-resolution displays," in *Proc. Big Data Vis. Anal. (BDVA)*, Sep. 2015, pp. 1–8.
[150] S. Liu, W. Cui, Y. Wu, and M. Liu, "A survey on information visualization: Recent advances and challenges," *Vis. Comput.*, vol. 30, no. 12, pp. 1373–1393, 2014.
[151] S. Ingram, T. Munzner, and M. Olano, "Glimmer: Multilevel MDS on the GPU," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 2, pp. 249–261, Mar. 2009.
[152] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in visual data analysis," in *Proc. 10th Int. Conf. Inf. Vis. (IV)*, Jul. 2006, pp. 9–16.
[153] S. Miksch and G. Santucci, "Understanding the role and value of interaction: First steps," in *Proc. Int. Workshop Vis. Anal. (EuroVA)*, 2011, pp. 17–20.
[154] J. S. Yi, Y. A. Kang, and J. Stasko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 6, pp. 1224–1231, Nov. 2007.
[155] Z. Liu and J. Stasko, "Mental models, visual reasoning and interaction in information visualization: A top-down perspective," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 999–1008, Nov. 2010.
[156] D. Gotz and M. X. Zhou, "Characterizing users' visual analytic activity for insight provenance," *Inf. Vis.*, vol. 8, no. 1, pp. 42–55, 2009.
[157] K. Sedig, P. Parsons, and A. Babanski, "Towards a characterization of interactivity in visual analytics," *JMPT*, vol. 3, no. 1, pp. 12–28, 2012.
[158] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale, "Beyond mouse and keyboard: Expanding design considerations for information visualization interactions," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2689–2698, Dec. 2012.
[159] M. Spindler, S. Stellmach, and R. Dachselt, "PaperLens: Advanced magic lens interaction above the tabletop," in *Proc. ACM Int. Conf. Interact. Tabletops Surfaces*, 2009, pp. 69–76.
[160] J. Browne, B. Lee, S. Carpendale, N. Riche, and T. Sherwood, "Data analysis on interactive whiteboards through sketch-based interaction," in *Proc. ACM Int. Conf. Interact. Tabletops Surf.*, 2011, pp. 154–157.
[161] W. Aigner, A. Bertone, S. Miksch, C. Tominski, and H. Schumann, "Towards a conceptual framework for visual analytics of time and time-oriented data," in *Proc. 39th Conf. Winter Simulation*, Dec. 2007, pp. 721–729.
[162] S. Garg, J. E. Nam, I. Ramakrishnan, and K. Mueller, "Model-driven visual analytics," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2008, pp. 19–26.
[163] C. R. Aragon, S. J. Bailey, S. Poon, K. Runge, and R. C. Thomas, "Sunfall: A collaborative visual analytics system for astrophysics," *J. Phys., Conf. Ser.*, vol. 125, no. 1, 2008, Art. no. 012091.
[164] C. Gormley and Z. Tong, *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. Newton, MA, USA: O'Reilly Media, 2015.
[165] R. Ranjan, "Streaming big data processing in datacenter clouds," *IEEE Cloud Comput.*, vol. 1, no. 1, pp. 78–83, May 2014.
[166] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache spark: A unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.
[167] S. R. Gomez, H. Guo, C. Ziemkiewicz, and D. H. Laidlaw, "An insight- and task-based methodology for evaluating spatiotemporal visual analytics," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2014, pp. 63–72.
[168] M. Tory and T. Moller, "Evaluating visualizations: Do expert reviews work?" *IEEE Comput. Graph. Appl.*, vol. 25, no. 5, pp. 8–11, Sep. 2005.
[169] Y. Wu, G.-X. Yuan, and K.-L. Ma, "Visualizing flow of uncertainty through analytical processes," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2526–2535, Dec. 2012.

[170] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim, "The role of uncertainty, awareness, and trust in visual analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 240–249, Jan. 2016.
[171] A. Slingsby, J. Dykes, and J. Wood, "Exploring uncertainty in geodemographics with interactive graphics," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2545–2554, Dec. 2011.
[172] C. D. Correa, Y.-H. Chan, and K.-L. Ma, "A framework for uncertainty-aware visual analytics," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2009, pp. 51–58.
[173] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz, "Uncovering clusters in crowded parallel coordinates visualizations," in *Proc. IEEE Symp. Inf. Vis.*, Oct. 2004, pp. 81–88.
[174] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauf, "An edge-bundling layout for interactive parallel coordinates," in *Proc. IEEE Pacific Vis. Symp. (PacificVis)*, Mar. 2014, pp. 57–64.
[175] J. Heinrich and D. Weiskopf, "Continuous parallel coordinates," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 6, pp. 1531–1538, Nov. 2009.
[176] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. Netto, and R. Buyya, "Big data computing and clouds: Trends and future directions," *J. Parallel Distrib. Comput.*, vol. 79, pp. 3–15, May 2015.
[177] O.-H. Kwon, C. Muelder, K. Lee, and K.-L. Ma, "A study of layout, rendering, and interaction methods for immersive graph visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 7, pp. 1802–1815, Jul. 2016.
[178] G. L. Andrienko, N. Andrienko, D. Keim, A. M. MacEachren, and S. Wrobel, "Challenging problems of geospatial visual analytics," *J. Vis. Lang. Comput.*, vol. 22, no. 4, pp. 251–256, 2011.

WENQIANG CUI received the B.Eng. degree in software engineering from Northeastern University, China, in 2012, and the M.Sc. degree in computer science from The University of Edinburgh, U.K., in 2013. He is currently pursuing the Ph.D. degree in computer science with the Norwegian University of Science and Technology. His research interests include data visualization, visual analytics, data analysis, and virtual reality.

Appendix B

Paper 2

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

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

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

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

ABSTRACT

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

KEYWORDS

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

1 INTRODUCTION

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

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

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

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

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

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

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

2 SYSTEM AND METHODS

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

2.1 System Overview

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

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

2.2 Data Binning-Based Clustering

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

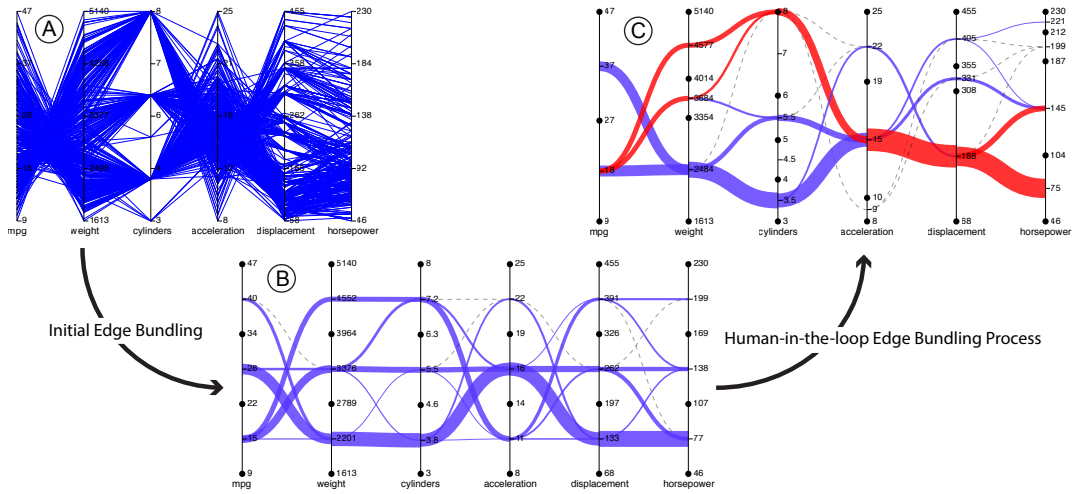


Figure 1: Overview of the system that supports HITL edge bundling in PCPs. A. Visualization of the Cars dataset [1] in a classic PCP. **B.** Edge bundling of the dataset with 3 initial clusters for each dimension. **C.** Interpretable edge bundling of the dataset with a subset highlighted (continuous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

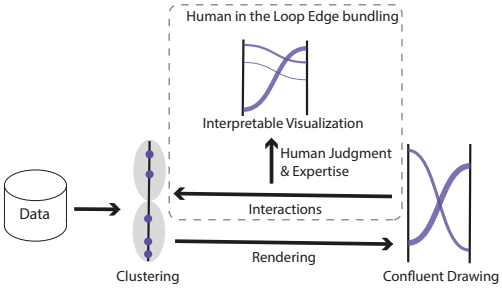


Figure 2: The workflow of the system.

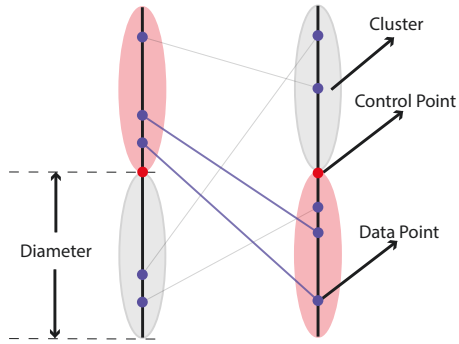


Figure 3: Using 1D binning to cluster data points for each axis in PCPs. The blue points are data points and the red points are control points. An edge between the axes represents two data points that belong to two clusters respectively. Elliptical areas represent clusters in an axis. The initial k is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

2.3 Density-based Confluent Drawing

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

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

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

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

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

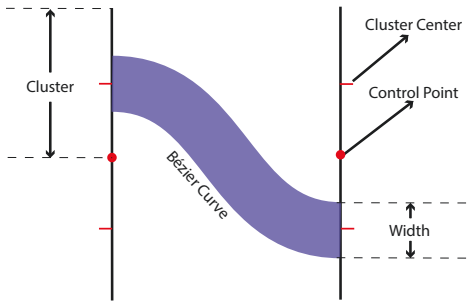


Figure 4: Using the density-based confluent drawing to bundle the edges that belong to a pair of clusters. For a pair of clusters, the bundled edge is rendered as a Bézier curve that starts from the center of a cluster and ends at the center of another. Its width represents the density of the cluster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

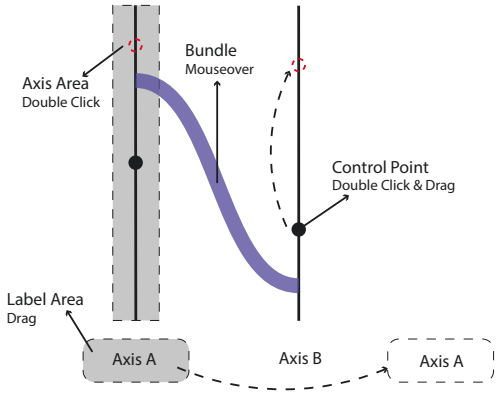


Figure 5: Interactions provided by our system for supporting HITL edge bundling. Double click on the axis area to add a control point to split a cluster. **Double click** on a control point to delete it to merge two clusters. **Drag** a control point along an axis to adjust the adjacent clusters. **Mouseover**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

2.4 Interactions for HITL Edge Bundling

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

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

3 EVALUATION

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

3.1 Experiments

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

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

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

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

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

Table 2: Comparison of the rendering time

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

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

3.2 Case Study

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

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

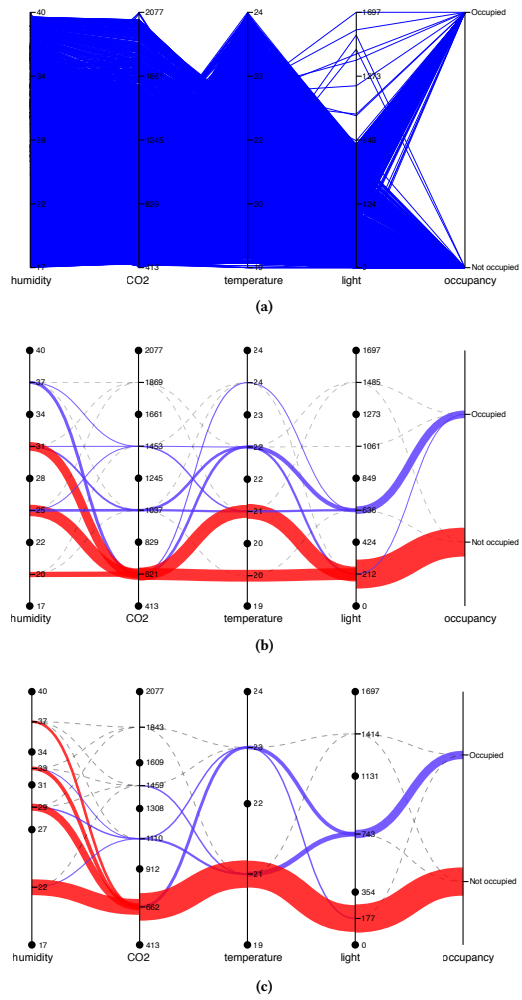


Figure 6: The visualization of the office dataset in the classic PCP and our system. (a) Visualization of the office dataset in the classic PCP. **(b)** Visualization of the office dataset in our system with 4 initial clusters for each dimension. **(c)** Visualization of the office dataset in our system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

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

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

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

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

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

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

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

3.3 Discussion

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

4 CONCLUSION AND FUTURE WORK

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

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

REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

Appendix C

Paper 3

Appendix D

Paper 4

This paper is awaiting publication and isnot included in NTNU Open

ISBN 978-82-326-6889-2 (printed ver.)
ISBN 978-82-326-5176-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology