Ph.D. Dissertation

Supporting Scalable Interaction and Analysis in Data Visualization using Colors

색상을 활용한 확장성 높은 상호작용 및 분석을 지원하는 데이터 시각화

August 2023

Dept. of Computer Science and Engineering
College of Engineering
Seoul National University

Jinwook Bok

Supporting Scalable Interaction and Analysis in Data Visualization using Colors

Advisor: Jinwook Seo

Submitting a Ph.D. Dissertation of
Electrical Engineering & Computer Science

May 2023

College of Engineering
Seoul National University

Jinwook Bok

Confirming the Ph.D. Dissertation written by
Jinwook Bok

June 2023

Chair	Sun Kim	(Seal)
Vice Chair	Jinwook Seo	(Seal)
Examiner	Youngki Lee	(Seal)
Examiner	Hanbyul Joo	(Seal)
Examiner	Jaemin Jo	(Seal)

Abstract

Jinwook Bok

Department of Electrical Engineering & Computer Science

College of Engineering | Seoul National University

Information visualization harnesses the capabilities of human vision by representing data through visual graphics, facilitating effective data exploration through visual graphics. However, with the ever-increasing size of data surpassing the limitations of human perception, providing scalability in visualizations has become a significant topic of research. Various methods and workflows have been developed to address the issue of scalability in information visualization, with combining multiple extracted information from the data in a coherent manner to enhance interaction with data. However, despite the efforts to effectively manage scalability, there are situations where dealing with the information of multiple values across multiple items becomes inevitable, particularly when resources are limited. In such cases, previous approaches may become inadequate, and users may need to resort to interacting with individual items, thereby encountering the scalability problem once again.

Reflecting the limitations, we introduce approaches in resolving the scalability issue of interacting with large sized multivariate data, utilizing colors as an important channel for resolving scalability. We leverage the critical advantage of colors in visualizing multiple values in limited space, allowing users to expand their understanding by interpreting the color patterns associated with different values. Tackling scalability issues in visualizing multiple items, we present Parallel Histogram Plot (PHP), a technique that overcomes the innate limitations of parallel coordinates plot (PCP) by attaching stacked-bar histograms with discrete color schemes to PCP. The color-coded histograms enable users to see an overview of the whole data without cluttering or scalability issues. Each rectangle in the PHP histograms is color coded according to the data ranking by a selected attribute. The color-coding scheme allows users to visually examine relationships between attributes, even between those that are displayed far apart, without repositioning or reordering axes. Addressing the complexity of multiple attributes in items, we introduce IssueML, a visualization system for monitoring and analyzing multiple issues that occur during the development of large softwares. Based on expert interviews, IssueML is equipped with specialized visualization techniques for monitoring issues and their progress over time. With the help of multiple, coordinated views, IssueML enables scalable observation and analysis of multiple issues, following the Visual Information Seeking Mantra. Finally, to support the user's interaction with multiple items, we propose TRaVis, a novel visualization approach in visualizing temporal rank data. In TRaVis each of the ranking changes are expressed as a single row of color patches, which are stacked according to order without overlapping. Such heatmap-like visualization enables the observation of trends of multiple items in a non-cluttering manner. By altering how items are stacked in the visualization, TRaVis enables the examination of temporal rank data in conjunction with the sorting criterion, which supports curious individuals in their visual information seeking process. We wrap up the dissertation by discussing the learned lessons and suggesting future research agendas based on the three researches.

Keywords: Information Visualization; Multivariate Data; User Interaction; Colors
Student Number: 2015-22900

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Information visualization leverages the graphical representation of data to facilitate user interaction and information retrieval, taking advantage of human visual perception. In order to effectively communicate information through visualizations, the original data is often transformed or modified to align with the specific visualization purpose and the chosen visual channels. This process of modifying the data plays a crucial role in determining the effectiveness of the visualization. By carefully considering the task at hand and selecting appropriate data modifications and visual channels, information visualizations can effectively express complex information, enabling users to make meaningful interpretations and gain valuable insights from the data.

However, providing scalability has become a crucial challenge in information visualization due to the continuous growth in data volume and complexity. To visualize with scalability, it is imperative to effectively represent information within the limitations of limited space and human perception. Thus, to address the challenge of interacting with large data, it is common to extract essential information from the data and reduce the number of targets to be rendered based on the user's objectives. One example is to cluster multiple items into a single cluster, effectively reducing the number of individual targets to interact with. By grouping similar items together, users can efficiently identify and analyze the shared characteristics within the data without being overwhelmed by the sheer volume of multiple items. Furthermore, as no single approach can extract all the necessary information from complex data, it is a common practice in visual analytics to express multiple interconnected facets of the data, each reflecting different yet related aspects. By combining multiple facets of the data in a rigorous manner, following established guidelines such as the top-to-bottom approach in the Visual Information Seeking Mantra [71] or the alternative bottom-up approach [47, 80], users can effectively interact with large datasets and search for information in a scalable manner.

Despite ongoing efforts to address scalability in visual analytics, challenges still persist in the realm of scalable data exploration. While reduction techniques can be applied to simplify the data, there are instances where users need to interact with each value within multiple items, leading to the reintroduction of scalability issues. For example, although clustering can make the observation task more efficient, users may still need to interact with individual items within the clusters. Furthermore, approaches in visual analytics can introduce additional complexity to the overall visualization process. While they may demonstrate effectiveness in controlled research settings, their applicability in real-world situations may vary. In contrast to controlled settings, real-world users often face ambiguous tasks that lack clear definitions or understanding. They may also encounter resource limitations, such as limited computational power or lack of domain-specific knowledge, which can hinder their ability to effectively visualize the data based on the given task. While the increase in variety and complexity of data to interact further exacerbate the need for scalable visualization solutions in such limited situations, the attention given to these critical issues has been relatively insufficient.

In situations where appropriate reduction methods are not available or limited, users often have to resort to simpler visualization techniques that focus on expressing individual values without reducing the data. These techniques generally leverage visual properties such as positions or lengths to express multiple values, based on established results of human perception. However, such results do not consider the challenges that arise when multiple items need to be displayed within limited space. The existing results of human perception, although effective in larger spaces, may not adequately address the challenges of visualizing multiple values in restricted spaces. As a result, the use of visual approaches that were previously effective can be limited, and the overlapping of visual components can hinder the interpretation of the data. The difficulty in effectively visualizing the details of multiple items has become a critical problem in visualization, especially with the continued growth in the size of interactive data. Unfortunately, the issue of expressing details of multiple values has received relatively less attention in both research and practice compared to the emphasis on visualizing a scalable overview. This is particularly true when it comes to supporting novice users who may have limited knowledge of visualization methods and rely on simpler visual techniques. Thus, addressing this limitation in information visualization is crucial for bridging the gap between the growing complexity of data and inexperienced users, by providing them with enhanced usability through visualization techniques that effectively represent multiple values.

Taking into account the limitations highlighted earlier, our dissertation focuses on the design and implementation of novel visualization techniques aimed at supporting users in effectively interacting with multiple values of multiple items in multivariate data. In our research, we recognize and leverage the advantageous characteristics of colors as a critical channel for addressing the scalability issue. Colors offer the benefit of conveying information without requiring excessive space, which is crucial in dealing with scalability challenges. Additionally, colors enable users to directly interact with individual values within the visualization, enhancing the overall understanding and facilitating direct interaction and comprehension. While colors are commonly utilized in visualizing multiple patterns, such as expressing gene sequences in bioinformatics [1], they are often underutilized in other domains due to concerns about accuracy limitations. However, in situations where the size of the data is significant and previous 'accurate' approaches are ineffective, colors can be highly effective in limited situations. To leverage the advantages of colors in the researches, we employ mappings that assigns colors to values, allowing the expression multiple values of multiple items as color patterns. By carefully positioning of these color patterns, the visualizations ensure effective understanding of the data from which users can interact with the data across various contexts.

In the dissertation, we delve into three critical aspects related to the scalability issue in information visualization: Effectively handling a large number of items, dealing with multiple and complicated attributes, and supporting user's interaction in large sized data. Firstly, focusing on providing a scalable visualization in multiple items, we developed Parallel Histogram Plots for expressing the overview of multiple items augmenting parallel coordinates plot. Secondly, dealing with multiple, complicated attributes of items, we implemented IssueML, a visualization tool for monitoring how fields in multiple issues have progressed over time. Finally, for supporting user interaction with multiple items, we designed TRaVis, a visualization technique for displaying multiple ranking changes of items from which users can seek for information in various perspectives.

Thesis Statement Utilizing colors as a critical channel to express values can overcome the scalability and complexity issues in visualizing multiple values of multiple items, supporting user's exploration and interactions with large multivariate data.

1.2 Research Questions

Over the course of the research, we have sought ways to address the thesis statement of utilizing colors to deal with scalability in expressing large sized data with the following research questions

RQ1. How can multiple items in data be visualized in a scalable manner using colors?

RQ2. How can items with complex, multiple attributes be effectively expressed with colors?

RQ3. How can visualizations support the users' interaction with multiple items utilizing colors?

1.3 Contributions

The core contributions of this dissertation are as follows:

1. Development and evaluation of **Parallel Histogram Plot**, a visualization technique for visualizing scalable overview of multiple items in multivariate data by attaching colored, stacked bar histograms in each axis of parallel coordinates plot.

2. Design and implementation of **IssueML**, an interactive visual system for monitoring and analyzing multiple issues and its related information that occur during development of a large software.

3. Implementation of **TRaVis**, a visualization technique for supporting the observation and exploration of multiple items in a temporal rank data.

1.3.1 Augmenting Parallel Coordinates Plots with Color-coded Stacked Histograms

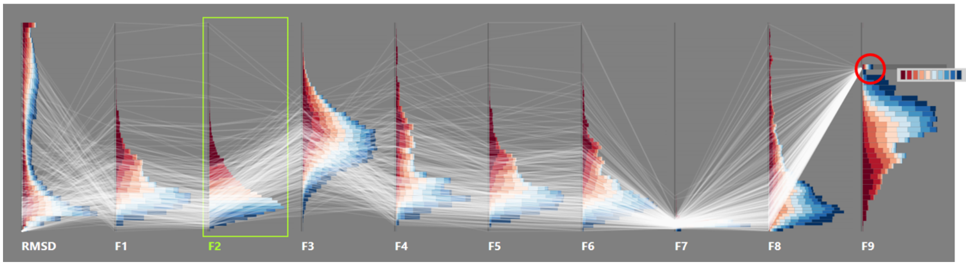


Figure 1.1: Parallel Histogram Plots (PHP) visualizing multiple attributes.

Addressing RQ1, we designed Parallel Histogram Plots (PHP), a novel visualization technique for displaying scalable overview of multiple items. Parallel coordinates plot (PCP) is a commonly used visualization technique for observing the information of multivariate data. In PCP, information of each item are encoded as a polyline, and from the patterns of polylines corresponding to multiple items, users can discover major trends and outliers in the data. However, scalability of PCP is limited due to the cluttering of polylines, in which patterns become difficult to discern as the number of items to visualize increases. To overcome the limitations, we developed Parallel Histogram Plots (PHP) that augments PCP with color-coded stacked bar charts attached to each axis of PCP. From the stacked bar charts, users can observe the overall information of multiple attributes in a scalable manner. Furthermore, the bar charts in PHP are color-coded based on a user-selected attribute. By examining the color distribution across the histograms, users can uncover relationships between the selected attribute and other attributes, even when those attributes are distant from each other. This capability overcomes the limitation of PCP, which is limited in the observation relationships between distant attributes.

1.3.2 Interactive Visualization System for Monitoring Issues in Industrial Software Development

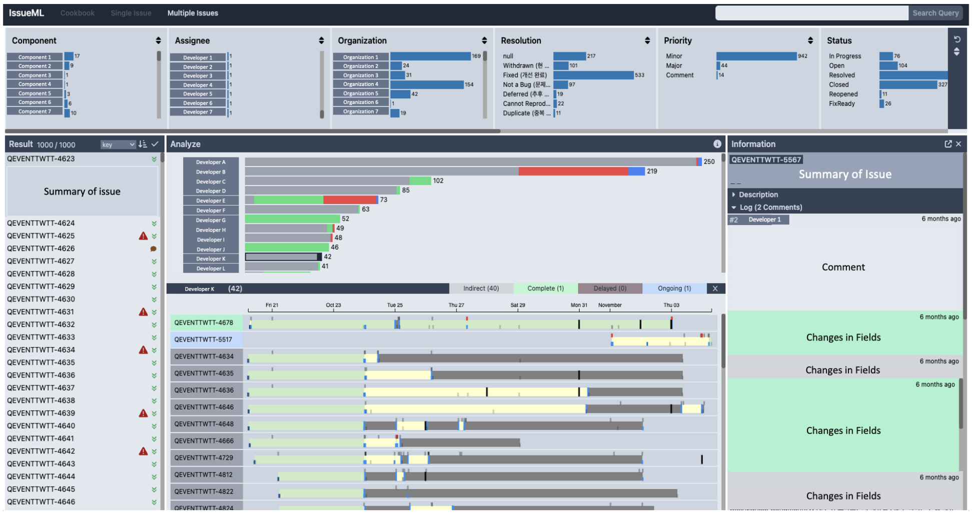


Figure 1.2: Overview of IssueML displaying multiple issues.

Focusing on RQ2, we developed IssueML, an interactive visualization system for monitoring the progress of multiple issues. Errors that occur during the development of a large software are recorded as issues. Issues contain critical information related to the symptoms and conditions of the errors, and managers are responsible for monitoring such issues to ensure timely resolution. However, the sheer volume of issues to be monitored, coupled with the fact that information within each issue evolves over time as the resolution progresses, presents a significant challenge. Due to the complexity of changes in multiple issues, managers were previously required to manually review the details of each issue to understand its progress, which was time-consuming and inefficient. To address these limitations, we developed IssueML, a visual analytic system designed for monitoring the progress of multiple issues. Based on interviews with domain experts, IssueML is equipped with various visualizations that aid the observations of multiple issues following the Visual Information Seeking Mantra [71]. Especially, the progress of each issues are expressed as patterns utilizing colors, from which users can effectively monitor changes in multiple issues and observe how critical information has evolved over time.

1.3.3 Towards Supporting the Exploration of Temporal Rank Data with Multiple Colormaps

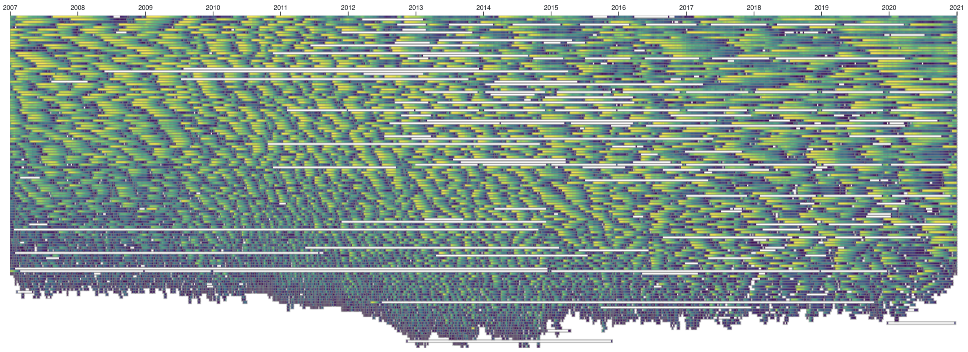


Figure 1.3: TRaVis visualizing the ranking changes of multiple items.

Related to RQ3, we developed TRaVis, a visualization technique for supporting the exploration of multiple items in temporal rank data. The concept of rank is widely used in comparing items, as it offers the convenience of summarizing the related context into a single value, i.e., rank. Although visualizations are often helpful in exploring rank data, previous methods have had limitations in depicting changes in rankings of multiple items over time. We designed TRaVis, a novel visualization approach specifically designed for displaying temporal rank data. In TRaVis each change in ranking for an item is represented by a row of color patches, which are stacked in a particular order to create a heatmap-like visualization. The heatmap-like visualization of TRaVis effectively displays trends in multiple items without overlapping them. By changing the stacking order of items, TRaVis allows the observation of temporal rank data in relation to the sorting criterion, which supports curious individuals in their visual information seeking process.

1.4 Structure of the Dissertation

The remainder of the dissertation is organized as follows. First, in Chapter 2, we introduce prior works related to utilizing colors as a visual channel, and also cover previous approaches related to the researches in the dissertation. In Chapter 3, we present Parallel Histogram Plots, in which we illustrate the design of the technique and measure its effectiveness compared to alternative visualization techniques in the correlation coefficient retrieval task. Chapter 4 describes the system overview and use cases of IssueML, including its system requirements which was established based on interviews with domain users. Following in Chapter 5, the designs and its rationale of TRaVis, along with various use cases in visualizing real world data are demonstrated. We discuss insights gathered from the researches in Chapter 6, and conclude our dissertation in Chapter 7 by summarizing the contributions and presenting potential research topics related to the dissertation.

Chapter 2

Related Work

This chapter covers previous works related to the dissertation that motivated the researches. First we discuss colors as a visual channel, and cover researches related to mapping values to colors. Second, we introduce overview in how users interact with the overall data. Finally, we discuss previous approaches related to our researches of visualizing data in a scalable manner.

2.1 Colors as a Visual Channel

When color is employed as a visual channel, data information is encoded into different properties including hue, lightness, and saturation. These color properties enable users to extract valuable values and insights from the visual representation of the data. Researches for understanding how users perceive information in colors have been researched in various tasks and perspectives, including how people distinguish different colors with names [31], or measuring its effectiveness in graphical perception according to various tasks [9, 30]. However, colors underperform in the perspective of graphical perception [9, 30] and is prone to complicated interaction effect in multiple colors that affect the perception of humans, such as contrast altering the appearance [54], or various visual illusions that occur under certain conditions. Thus, it requires careful usage when expressing information with colors.

Addressing such difficulties in utilizing colors, various researches and guidelines were proposed for overcoming and improving the usage in visualizing information with colors [72, 87]. For example, Wang et. al introduced a method for optimizing the selection of colors in multiclass scatterplots [86], and Lu et. al developed integrated approach for creating and assigning colors to visualizations [50]. Likewise, various guideline in choosing colors for effective representation of values [48, 85] have aided users in effectively observing information with colors despite the limitations.

Meanwhile, compared to handling a finite, defined set of colors, attempts in visualizing continuous values as colors mostly involve utilizing colormaps. Colormaps refers to a continuous set or array of colors, in which the range of values can be mapped to the range of colors for values to be expressed as colors. The rainbow colormap is one widely used color map in which the colors of the rainbow spectrum are continuously mapped to values for expressive representation. However, the effectiveness of rainbow colormap are known to underperform in various aspects [7, 68], and thus it is generally not recommended to use them. Alternatively, carefully designed colormaps based on heuristic rules are often utilized in visualizing quantatitive values. Popular colormaps used include the viridis colormap from mpl colormaps [79], or the predefined color scales in Colorbrewer [27], which are generated according to various criteria related to human perception and defined rules [92]. Moreover, approaches in further enhancing the versatility of the colormaps were researched, for example the Value-Suppressing Uncertainty Palettes for dealing with the problem of expressing uncertain values in the data [11].

Exploiting the advantages of colors in expressing patterns, visualizations have utilized multiple rows of colors for visualizing the information of multiple items in a scalable manner [2, 69]. In this approach, the advantage of colors in visualizing patterns in limited space (compared to line charts or horizon charts) without cluttering is leveraged [22]. The Line Graph Explorer [44] uses a focus + context technique with multiple colored line charts that changes into rows of colors when shrunk. On Lasagna plots [73], the rows and columns of the plot can be sorted by various methodologies (vertically, horizontally, etc.) to find trends in items or in a certain time range. The multiple rows of colors can also be utilized in bioinfomatics for encoding and comparing patterns in genomic information [1], multiple runners [59], or in expressing fitness activities [55]. Expanding upon such previous approaches, in TRaVis we express each of the ranking changes as a single row of colors for scalable visualization. To further support scalability in the visualization, we greedily stack each row of colors in a manner similar to Tetris blocks, from which users can observe patterns in multiple items in limited space.

2.2 Approaches in Visual Information Seeking

As a single visualization facet is limited in expressing all of the information in the data, it is important to provide multiple facets of data in visual analytics from which users can understand the data step by step. In guiding how such multiple facets should function, the Visual Information Seeking Mantra [71] is one of the most renowned guideline. The Visual Information Seeking Mantra is consisted of 4 main keywords, of Overview, Zoom, Filter, and Details on Demand, which corresponds to key steps of visualizing information of data in a top-down manner from which users can effectively understand

and interact with. Alternatively, visual interaction of starting from the detail of a single item and expanding the target based on the information is another popular approach in information seeking [47, 80]. Converse to the top-down approach in Visual Information Seeking Mantra, this "planting a seed and watching it grow" approach mostly refers to the bottom-up style in interacting with multiple items in the data.

In both the top-down approach of the Visual Information Seeking Mantra and the alternative bottom-up approaches, the main focus is providing guidelines on how should users interact with the data. Conversely, the term of information flaneurs [14] approach from a different perspective in interacting with data, of user's attitude and intention with the data. The term flaneur is commonly used to describe a person who leisurely walks around, observing their surroundings without any particular goal in mind. In the context of information seeking, information flaneur refers to a paradigm of information interaction that is driven by aesthetics and the desire for serendipitous encounters, rather than a clear motivation or task. An example that supports this concept of information seeking is the Bohemian Library [74], where multiple interconnected visualizations support the serendipitous discovery of books. When designing visualizations for information flaneurs, it is important to provide multiple facets of the data that allow users to encounter information from various perspectives. Although rank is a popular means of comparison, users had limited interaction with the temporal aspects of rank data due to the complexity of multiple ranking changes in numerous items. Dealing with this, in designing TRaVis, we visualize the ranks of items in limited space without overlapping using multiple rows of heatmaps, enabling the discovery of interesting patterns in each items. Also, by sorting the items according to various criteria, we enable the observation of temporal rank data in multiple perspectives.

2.3 Approaches in Visualizing with Scalability

In this section we discuss previous researches related to visualizing multiple items of multiple values in various approaches related to our research.

2.3.1 Parallel Coordinate Plots

This section introduces previous research upon which we built our visualization and interaction idioms in this study. We first summarize previous approaches to enhancing the performance of PCP. Then, we introduce Attribute and Influence Explorer [76, 77] which inspired our work through the way it utilizes histograms of stacked elements.

Approaches to Enhancing PCP

Many efforts have been devoted to enhancing the performance of PCP. They have focused mainly on making it more scalable by reducing visual clutter. These efforts can be grouped into four categories according to the types of techniques employed: reduction, transformation, integration, and interaction.

Reducing the number of items displayed on PCP is a popular approach. Such a reduction results in fewer polylines, thus leading to less cluttering. Reducing the number of items to show is done mostly by aggregating (or compressing) the data. Because it is subject to information loss, maintaining the characteristics of the original data as much as possible has been the main concern of this approach. Various data reduction methods have been introduced, including clustering [20, 42], binning [58], sampling [15], and image-space methods based on image processing algorithms [3]. The number of dimensions can also be reduced by using dimension hierarchy [89], by measuring the distance between attributes [36], and by contracting adjacent axes[57]. Finally, without any form of data reduction, the ordering of the dimensions can be changed for a more orderly overview [12, 49, 51, 61, 91]. Reordering attributes is critical in PCP because it is impossible to investigate relationships between nonadjacent attributes.

Transforming the components (polylines and axes) of PCP is another well-known approach. Bezier curves can replace the polylines of PCP, which are more appropriate for bundling and thus more suitable for presenting clusters of items [33, 60, 84]. Other methods of transforming the polylines include density fields [32], polygons [52], bands [46], and layers of consistency maps [56]. These substituted visual elements can serve a specific purpose better compared with the original polylines. For example, in the method of Parallel Sets [46], the lines are replaced with bands to effectively visualize categorical variables. The axes of PCP are also targets for transformation. The arrangement of the axes can go beyond the 1D linear one. They can be arranged on a 2D plane [8] or in a 3D space [10, 40] to enable users to examine relationships among multiple attributes. The axes can even be transformed into curves [64] to show the data in a polar coordinate system, or be tilted by the tension of lines in PCP [83]. When applying these approaches, we should consider the trade-off, in that we could lose the strong perceptual advantage of the original visual encoding of PCP in correlation estimation by (line-crossing) pattern recognition.

Integrating other visualizations with PCP can facilitate the visual information seeking process by revealing different facets of the data that are difficult to grasp only from the patterns of polylines. Scatterplots that display the relationship between two attributes are popular visualizations for such a purpose [65, 90]. Stacked bar charts [37] and histograms [21] attached to an axis of the PCP can show the distribution of each attribute. Other visualization idioms can also be integrated with PCP, including star glyphs [17], box plots [37, 45], spherical coordinate systems [82] and MDS plots [24]. In PHP, we integrate color-coded histograms with PCP to visualize the overview of multiple attributes in a scalable manner. Color-encoding acts as an important channel that reveals the relationship between attributes, even when they are not adjacent to each other.

There have been a few attempts to use color as an auxiliary channel for delivering additional information, such as the distribution of values of an attribute. In Value-cell bar charts [43], bars are split into multiple cells that correspond to one or more individual values. The cells are color coded by the sum of the values. From the color distribution made by each of the cells, the distribution of the values in each of the bars can be inferred. Janetzko et al. [37] utilized color-coded stacked bar charts on the axes of PCP to show clusters generated by K-means clustering. Geng et al. [21] used color in Angular Histograms for redundantly encoding the height of the tilted bars to help users perceive the height more accurately. In contrast to these approaches, our method uses the color channel for showing the linear relationships between attributes. Using colors in histograms attached to axes makes it possible for users to grasp linear relationships between (even distant) attributes through implicit connections made by perceptually matching colors. Unlike [43], which uses colors to reveal the distribution of values within a single bar chart, PHP uses colors to reveal the relationship between multiple distributions or attributes. This approach enables the analysis of data with multiple attributes. Compared with [37], in which colors are mapped onto the groups generated by a clustering algorithm, our approach is more universally applicable and provides the direct relationships between attributes. In addition, compared with [21], in which histograms must be tilted, our approach preserves the original shape of the histograms to prevent users from getting confused by the distortion. Also, the color channel is used as a pivotal channel that reveals the relationships between attributes in PHP, rather than as a redundant channel as in [21].

Interaction techniques also help users find information with PCP by facilitating the exploration process. Lack of interaction in PCP is known to discourage users from drawing information from the visualization [41]. The Angular Brushing technique enables users to filter data by the value of the angle between the line and the axis in PCP [28]. Roberts et al. designed a sketch-based brushing for high-dimensional pattern searches and a data-dependent smart brushing based on metadata [67]. When a visualization is integrated into PCP, novel interactions are often designed to make it work harmoniously with PCP. For instance, OPCP, a visualization technique that integrates a scatterplot-based visualization into PCP, has a dedicated interaction named O-brushing for facilitating pattern selection in complex data [65]. In PHP, we also designed interaction techniques that aid users in investigating small regions of a histogram that are too small for details to be seen — e.g., two-level semantic zooming that can enlarge a small selected region of the histogram while maintaining the overall layout of PHP.

2.3.2 Visualizing Temporal Rank Data

Tables are the most primitive and popular method used to visualize rankings of items. In tables, rankings of multiple items can be rendered in a scalable manner with the help of interaction techniques, such as the focus context technique utilized in Table Lens [66]. However, tables are mostly limited to visualizing a single rank chart, and thus is not effective at displaying multiple rank charts of temporal information [63]. Thus, researches such as [63, 81] has integrated line charts with tables to visualize ranking changes over time, with the help of interaction techniques related to transitioning time.

Line charts are also frequently used to visualize multiple time series of rank changes. For example, in Rank Clocks [5] line charts of rankings are rendered in radial coordinates, aimed at observing periodic patterns. Extending upon the line charts, slope charts [75] can provide additional information related to the rank data by appending visualizations in each of the axes corresponding to rank charts over time. For example, in LineUp [23] ranks by semantics of multiple, heterogeneous attributes can be compared with from the connected stacked bar charts. Perin et. al modified slope charts as gap charts [62], in which both the score and its ranking of items can be rendered in a scalable manner. However, in both line charts and slope charts, cluttering between the lines of each items is inevitable. In we prevent overlapping between items by visualizing each of the ranking changes as a row of colormap. Not only the colormaps are individually non-cluttering, the accumulated patterns can reveal information that would be hard to be discovered in multiple line charts.

As scalability is limited in line charts, various researches have been proposed to overcome the cluttering issue. Xia et. al proposed a heatmap based approach [88], in which rankings of each items are rendered as coordinates in a 2D space connected by colors. However, such colors were applied according to a single rank chart, and thus only items that appear in the chart could be displayed. RankBrusher [26] utilized histograms with glyphs representing ranking ensembles, with heatmaps in the background supporting connection between adjacent items. RankExplorer [70] designed a ThemeRiver [29] based visualization to show changes in search query data over time, with glyphs assisting the visualization of ranking changes in items. But in both methods [26, 70], the techniques are only valid at visualizing the changes in adjacent rank charts, limiting users' observation task to chronological order. Visualizing ranks in stock data [25], or bicycle riders [13] were also introduced, but the techniques were limited for specific tasks. Converse to previous researches, in we aim at visualizing temporal rank data for information flaneurs, and visualize information of ranking changes with less restrictions.

Chapter 3

Augmenting Parallel Coordinates Plots with Color-coded Stacked Histograms

This chapter introduces Parallel Histogram Plot (PHP), a novel visualization technique for displaying scalable overview of multiple attributes in parallel coordinates plot, by utilizing color-coded stacked histograms. This research mainly addresses the first research question (RQ1) of the dissertation, of visualizing the overview of multiple items in a scalable manner using colors.

3.1 Background

Parallel coordinates plot (PCP) [35] is a visualization technique that arranges multiple attributes parallel to each other in a 2D plane. Clusters of data items and relations between attributes, including correlations, can be perceived by the patterns of lines in PCP. This pattern recognition becomes harder, however, when lines overlap more with each other as the number of items and attributes increases. Furthermore, relationships between attributes are difficult, if not impossible, to infer from visual patterns in PCP when the axes are not adjacent. Many approaches have been proposed to deal with these limitations, e.g., the overplotting of lines or the ordering of axes [34]; however, there are still many challenges that researchers have to face when visualizing data with PCP because of the innate limitations of the original PCP. Sometimes, the limitations have been resolved by sacrificing the original structure of PCP, which significantly weakens its perceptual advantages.

Reflecting the limitations, we introduce Parallel Histogram Plot (PHP), a visualization technique that deals with the innate limitations of PCP while preserving its perceptual advantages and characteristics. Following the Visual Information-Seeking Mantra [71], we augment the original polylines of PCP with color-coded stacked bar histograms. Attached to each axis of the original PCP layout, the histograms provide a scalable overview by showing the distribution of data items of each attribute. Polylines of PCP are used in the later stages of the Visual Information Seeking process, when the cluttering problem is less severe after less important items have been filtered out. Colors applied to the stacked bars of histograms are determined by a user-selected attribute. Visual comparison of the color distributions on histograms for multiple attributes reveals relationships between the attributes without cluttering or overlapping of lines as in PCP. Relationships between distant attributes that are hard, if not impossible, to grasp in the original PCP can be readily perceived in PHP through the visual comparison of color distributions for the attributes. We also designed interaction idioms for PHP to help users investigate the details of histograms in a limited screen space.

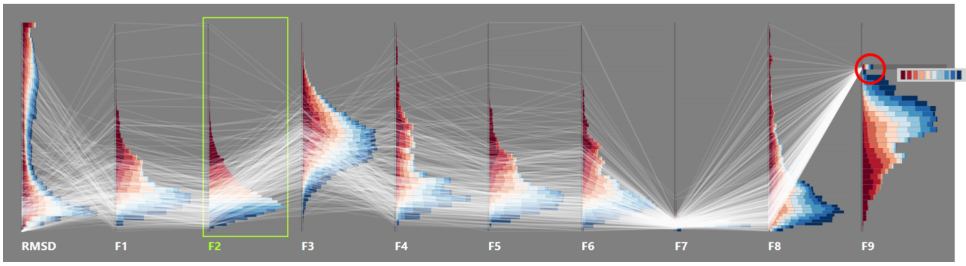


Figure 3.1: Parallel Histogram Plots (PHP) used to draw the CASP dataset [78]. The attribute F2 is selected for color coding. From the color distribution, it can be deduced that F2 is positively correlated with F1, F5, and F6 and negatively correlated with F9. In addition, the data items in the upper-right region (red circle) of the F9 histogram are selected and thus displayed as polylines. The widget on the F9 histogram helps with clicking tiny bars on the histogram.

3.2 Design of PHP

3.2.1 Design Rationale

Our approach focuses on overcoming the limitations of PCP while maintaining its original advantages. PHP is designed to deal with two critical limitations of PCP: (L1) cluttering of polylines and (L2) the difficulty in estimating relationships between nonadjacent axes.

L1 – Cluttering of polylines

The polyline encoding of PCP helps users recognize clusters/outliers and estimate correlations from visual patterns made from the line crossings. However, such encoding inherently suffers from a scalability issue, in that polylines clutter the view with too many overlapping lines. The problem becomes worse as the data size becomes larger.

L2 – Difficulty in estimating relationships between non-adjacent axes

PCP utilizes a linear layout to display multiple attributes. The linear layout is easy to understand and allows multiple attributes to be displayed in a relatively small area. However, the layout makes it challenging to interpret the relationship between attributes that are not adjacent. Thus, finding an effective order of attributes in PCP has been an important research topic.

To achieve our design goal of overcoming the two main limitations of PCP while preserving its advantages, we attach a histogram to each PCP axis. Histograms can reveal the distribution of data items on each attribute in a scalable manner, irrespective of the data size. It is also a relatively simple visualization that does not require any drastic modification of the original layout of PCP, fulfilling our objective of maintaining the innate advantages of PCP. However, histograms have the limitation that they cannot show any relationships between attributes [21]. To resolve this limitation, we adopt color as a crucial visual channel for expressing the relationships between attributes. We order the data items by a user-selected attribute and split the data items into groups according to the order while ensuring that each group has a similar number of items. We represent each group as a bar and assign a unique color to each group. We then build histograms by stacking color-coded bars, with each bar representing a group. By comparing color distributions on attributes in PHP, users can estimate the relationships not only between adjacent attributes but also between distant ones even without direct connections. The indirect connection provided by colors in PCP is free from cluttering and less influenced by the distance between the attributes. With the adoption of color encoding, the ordering of axes becomes much less important, as the relationship between attributes can be perceived by matching colors even when they are distant from each other. The next section and Figure 3.2 show how PHP is built from data.

Following the Visual Information-Seeking Mantra [71], we utilize the original polylines of PCP along with the color-coded histograms. The colored histograms are good at displaying an overview of the data and the relationships between attributes; however, they are not effective in helping users estimate the exact value of each data item. Meanwhile, polylines excel in helping users grasp the value of each data item at an attribute, but they easily suffer from cluttering when there are many of them. Thus, we combine these two components so that they complement each other. In the beginning, color-coded histograms show an overview of the data. After zooming and filtering out less important/relevant data items in the histograms, the polylines show details of a small group of data items selected from the histograms, enabling users to take advantage of the original PCP design.

3.2.2 Construction of Color-coded Histograms

Grouping data items

To construct the color-coded histograms of PHP, we first split the data into equally sized groups according to a user-selected attribute (Figure 3.2(a)). Instead of using the original data value of the selected attribute to derive groups, we use ranking as the criterion to create groups. First, data items are sorted by the user-selected attribute, and the data items are grouped according to the ranking. Unequally sized groups could occur because we make sure that data items with the same value for the selected attribute are placed together when splitting groups. This prevents data items that have the same value for the selected attribute from being inconsistently placed in different groups. Grouping by ranking mitigates the effects of outliers and skewed distributions in the color mapping, which will be applied to each group in the next step.

Applying colors to groups

Second, we apply a unique color to each group of the split data using a discrete, diverging color scheme (Figure 3.2(b)). A discrete color scheme is used, so that a color in the color scheme is assigned to a group. The color scheme is designed to distinguish between groups and to show the differences (or similarities) between them. We adopt this scheme to emphasize low- and high-ranked groups with more saturated colors because they are usually more valuable in the data analysis. In this paper, we use a ten-level blue-red diverging color scheme acquired from ColorBrewer2 [27] (low to high rankings from blue to red). We chose 10 as the number of colors to be rendered, which is close to the number of colors that a human can distinguish simultaneously [37].

Building stacked-bar histograms

Finally, using the preprocessed data (grouped by ranking and then color mapped), we draw a histogram that represents the distribution of each attribute on the corresponding PCP axis (Figure 3.2(c)). The histogram is constructed in the same way as a stacked bar chart is drawn, with each group in its unique color. When groups are stacked, the order of the color-coded elements must agree with the order of the colors in the color scheme so that elements in the same color are merged. This helps users perceive patterns from the color distribution. Users can also recognize the relationship between the selected attribute and others by perceiving the distribution patterns of colors across the histograms. The stacking of elements in PHP makes the layout similar to that of Attribute and Influence Explorer, with a histogram consisting of stacks of lightbulbs that represent individual data items. In PHP, however, a single stacked element represents a group of data, and its length

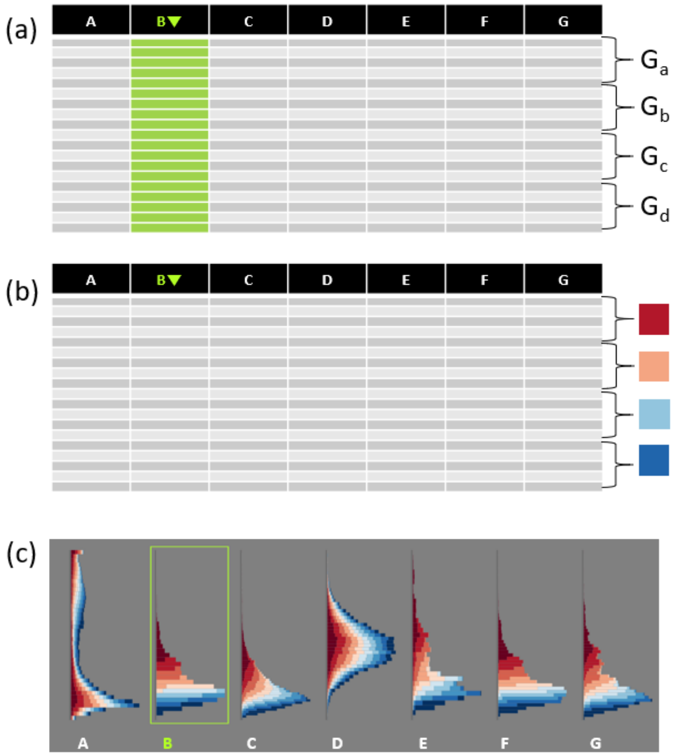


Figure 3.2: How PHP is constructed from data. (a) Data items are sorted by a user-selected attribute (attribute B), and items are grouped ($G_a - G_d$) according to the sorted order, i.e., the rank data items of the selected attribute. (b) A unique color is applied to each group, with a diverging colormap: a reddish color for higher ranks and a blueish color for lower ranks. (c) Stacked histograms are rendered in the applied color.

is proportional to the number of data items belonging to the group in the corresponding attribute.

3.3 Visual Information Seeking with PHP

Using the visual encoding idioms of PHP [1], users can recognize important features in the data, including the distributions of attributes, correlations between attributes, and outliers. We also design interaction idioms combined

[1] A demo version of PHP is available at https://bokjinwook.github.io/ParallelHistogramPlots/index.html

with the visual encoding idioms, including two-level zooming and ghost bars for more scalable and space-efficient exploration. In this section, we use 16 years of accumulated statistics data of baseball pitchers in Major League Baseball, acquired from FanGraphs [18], as the dataset for ease of explanation. The dataset contains 7,673 items representing each player's record of a year, with 17 sampled attributes reflecting the players' performance.

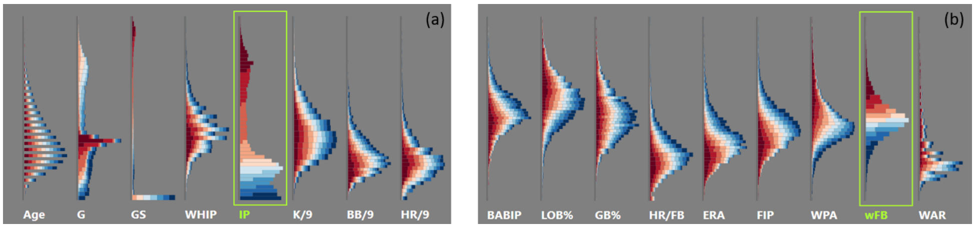


Figure 3.3: The Baseball dataset [18] rendered in PHP. The attribute selected for color coding (i.e., the pivot attribute) is bordered by a green rectangle ((a) IP and (b) wFB). Relationships between the pivot attribute and all the others can be observed from how the colors are distributed.

3.3.1 Interpreting PHP

PHP utilizes color coding derived from a single user-selected attribute, the so-called pivot attribute (Figure 3.3). In PHP, selecting the pivot attribute is a crucial step in revealing features in the data. Visually comparing color distributions on histograms can reveal relationships between the pivot attribute and all the other attributes. Users can steer their data exploration by changing the pivot attribute to seek relationships between attributes with different aspects. This methodology can be used effectively in situations in which only some of the attributes in the dataset are familiar. Users can start their exploration by first selecting a familiar attribute as the pivot attribute and then expand their knowledge from the known to the unknown attributes by analyzing their relationships.

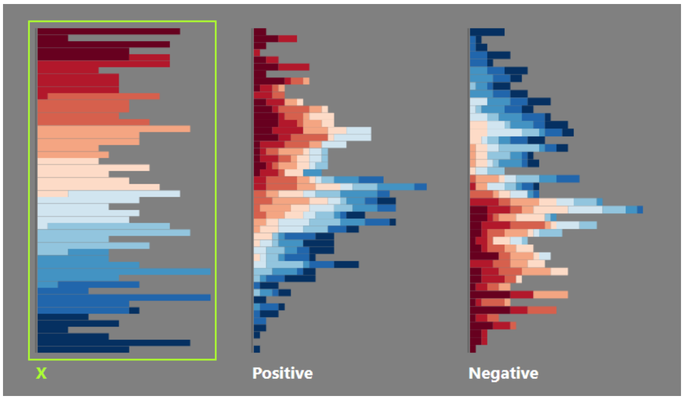


Figure 3.4: Example of positive and negative correlations displayed in PHP. The attributes, Positive and Negative have positive (+0.8) and negative (-0.8) correlations with attribute X, respectively.

Correlations between the pivot attribute and other attributes of interest can be estimated just by visually comparing the color distributions of the corresponding histograms. For example, in Figure 3.4, the attribute X is selected as the pivot attribute, with the color changing from blue to red from bottom to top. The histogram for the attribute Positive shows a color distribution very similar to that of the attribute X, implying a strong positive correlation between these two attributes. But the histogram of the attribute Negative shows a color distribution inverted from that of the attribute X, implying a strong negative correlation between these two attributes. By the same principle, in Figure 3.3(b), it is easy to recognize that WPA and LOB% are positively correlated with the pivot attribute wFB. Meanwhile, it can also be easily recognized that BABIP, HR/FB, ERA, and FIP are negatively correlated with wFB, as such visual recognition is not affected by the distance from the pivot attribute.

Users can also recognize clusters of items that have similar color patterns, or outliers that do not follow the major color patterns around them in the

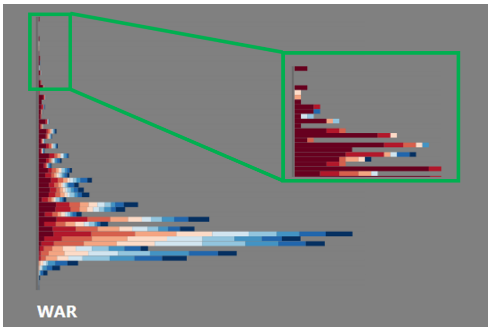


Figure 3.5: Histogram of WAR from Figure 3.3(b). The area inside the green box is magnified for visibility. In the region of the green box, it can be observed that some blue items are distant from the overall red data items.

histograms. Similar colors gathered in a small region indicate that the data items sharing similar properties are clustered in that region. In Figure 3.3(a), in which the histograms are color coded by the attribute IP, it can easily be observed that data items with high IP values are tightly gathered in the lower middle of the attribute G. Meanwhile, salient colors indicate that there exist data items outside the overall distribution of those that surround them, suggesting outliers. Figure 3.5 displays a magnified histogram of WAR from Figure 3.3(b). Some data items in the green box have salient blue colors that are different from the overall red surroundings. Compared with most of the

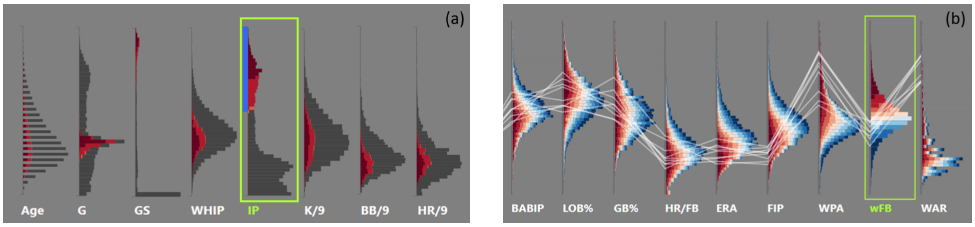


Figure 3.6: Displaying selected items in PHP. (a) Data items with high IP values are selected from Figure 3.3. The distribution of selected items can be compared with the overall distribution, revealing that the selected items are gathered in the lower-middle narrow region of G. (b) The distinct blue items among the dominant red items in the WAR histogram are selected from Figure 3.5 to be displayed by polylines. The polylines reveal detailed characteristics of the selected data items.

items nearby, these outliers have much lower rankings of the pivot attribute wFB, as is apparent from their distinct colors.

Like many other PCP-based visualizations, PHP supports common PCP-based interactions, including selecting items by the range of an attribute's value with brushing and changing the order of axes. In PHP, ordering axes is less important, as the relationship between attributes can be observed just by color even when they are far away. Nonetheless, PHP enables users to sort the attributes by correlation or similarity for a more efficient analysis of the data. PHP also supports common interactions/modifications related to histograms, such as selecting a range of data or bars of interest in the histograms or changing the number of bins. In PHP, we adopt the analytical strategy of comparing the distributions between selected and unselected data of Attribute and Influence Explorer [76, 77]. When items of interest are selected, the color-coded stacked bar histogram for the selected data is shown in the foreground, while the original histograms for all the data are shown in grayscale in the background of PHP (Figure 3.6(a)). In this way, the characteristics of the selected data can be examined in the context of the whole data.

Following the Visual Information-Seeking Mantra, we enable users to harmoniously use the original PCP in PHP, in situations in which the power of the pattern recognition in the original PCP can shine, i.e., exploring a smaller group of selected items with less clutter. When a user selects a bar for a group of items of interest by hovering over or clicking on it, the corresponding data items are shown as polylines as in the original PCP (Figure 3.6(b)). This interaction can help users make a connection between histograms and PCP lines. Users can show or hide either the histograms or the polylines to avoid potential visual interference between the lines and the bars. In Fig-

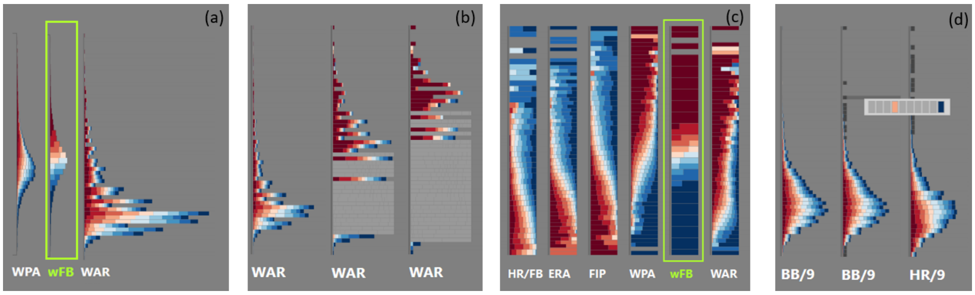


Figure 3.7: Various interactions of PHP designed to deal with small components. (a) Focus+context zooming enables users to enlarge histograms of interest (the histograms of WAR are enlarged when the axis is dragged). (b) Clamp zooming helps in observing small elements in a space-efficient manner (from left to right, the histograms of WAR are increasingly clamp zoomed). (c) When clamp zooming maxes out all bars, the histograms turn into a heatmap-like visualization that can display the color distribution in a minimal space. (d) Ghost bars in the right two histograms of BB/9 and HR/9 help identify small bars that are unseen in the leftmost normal histogram of BB/9. The UI widget of pop-up color patches helps users click on tiny bars.

ure 3.6(b), the outliers selected from Figure 3.5 are displayed as polylines. Detailed information about the selected data can be revealed from the polylines. The polylines show that the selected outliers tend to share similarities, and that they have relatively low wFB; high WPA; and low FIP, ERA, and HR/FB.

3.3.2 Tools for Zooming in on Small bars

While clicking and hovering interactions on histograms are simple and intuitive actions, issues arise when users must interact with small components in the visualization, which are hard to see and select. Each bar of a histogram in PHP consists of stacks of color-coded bars. Among the stacked bars, there can be small bars that are hard to interact with. We designed interaction techniques to overcome such problems.

Two-level zooming

One of the main issues with searching for information in histograms of stacked bars is the difficulty of noticing bins that are rendered too small owing to a skewed distribution, outliers, etc. To support observing small bins and bars, we introduce a two-level zooming interaction technique. First-level zooming, named focus+context zooming, widens the gap between axes to assign more space to an axis of interest while preserving the contexts around it in a shrunk space. In PHP, a histogram is horizontally attached to an axis, and thus occupies the space between two adjacent axes. Widening the gap between axes by dragging an axis gives more space to the histogram shown in the gap, which can reveal more details about the histogram (i.e., small bars getting bigger) (Figure 3.7(a)). As in the focus+context technique used in Table Lens [66], users can horizontally increase the size of histograms to see more details while maintaining contextual information about all the data in the visualization.

Focus+context zooming still has limitations if the distribution of a histogram is skewed too much. In such a case, a very large space is required for the histogram to see the details of tiny bars, which sacrifices the space for other histograms. Considering this problem, we complement first-level zooming with another space-efficient, second-level zooming technique called clamp zooming. Clamp zooming is a 'within-area' zooming technique. Without changing the allocated space for a histogram, it horizontally stretches each bar inside the histogram with the same magnification, being maxed out when reaching the maximum length. When the bars reach the maximum length, they are gray colored to distinguish them from other, smaller, not-maxed-out bars, which helps users focus on the smaller bars (Figure 3.7(b)). This clamp zooming helps users investigate the long tail of a skewed distribution. In Figure 3.7(b), outliers that had to be enlarged extensively in Figure 3.5 (distinct blue items in the upper region of the WAR histogram) can be seen in a relatively smaller space.

When clamp zooming maxes out all bars, the original colors of the bars are restored, and the histograms transform into a heatmap visualization (Figure 3.7(c)). This heatmap visualization is useful in extreme conditions, such as when the space allocated to a histogram is so small that the colormap from the histograms is hard to perceive. It can also be used to display a relatively high number of attributes within a limited screen size.

Ghost bars for invisibly small elements

When scaling histogram bars to fit the allocated space between axes, it is often inevitable that some bars cannot be shown because their heights become smaller than one pixel. In Angular Histograms [21], another histogram-based PCP visualization, this problem is resolved by using an additional visual encoding idiom named Attribute Curves. Attribute Curves provides a clue that some data elements exist in a bin. But this approach takes additional space along with the original visualization. We adopt a more space-efficient approach by using a visual cue within the visualization that implies the existence of small bins, named ghost bars. The ghost bar technique shows a small but noticeable gray bar for originally invisible bins, whose length is too short to be shown on the current scale (Figure 3.7(d)). From the gray bars, users can determine whether any bins are unseen because of their small size. The ghost bars are colored this way so that they can be distinguished from the normal histogram bars. Furthermore, they can be untoggled when they are not needed to prevent confusion.

Support for selecting tiny bins

Finally, we provide a UI widget to help users select small bins in a histogram bar. When users click on the empty area right next to a (small) bar in a histogram, a widget pops up and shows a color panel, in which the colors used in the bar are shown as patches (Figure 3.7(d)). In contrast to trying to directly click on the small bar in the histogram, which may be challenging, the color patches on the pop-up can be easily and more accurately clicked. Sometimes the pop-up widget can show colors that are invisible in the original small bar, which correspond to the invisibly small bars in the current scale.

3.4 Use Case

In this section, we demonstrate the efficacy and utility of PHP compared with other similar visualizations, e.g. the original PCP, AH [21], and scatterplot matrices (SPLOMs). For comparison, we used the protein tertiary structure dataset [78], which consists of 45,730 items with 10 attributes (i.e., the physicochemical properties of proteins). This dataset is part of the CASP (Critical Assessment of Techniques for Protein Structure Prediction) dataset, which contains various properties of a protein's structure.

As can be observed in Figure 3.8(a), the two main limitations of PCP previously discussed (L1 and L2) prevail in PCP. Because there are many items in the dataset, the overlapping of polylines is too severe in the original PCP, even though the lines are rendered translucent to mitigate the overlap. AH and PHP (Figures 3.8(b) and (c)) both mitigate the cluttering issue using histograms. AH utilizes a vector-based approach for each bar of the histogram, with an additional attribute of direction determined by the mean angle of the polylines in the corresponding bin. Owing to this direction attribute, the his-

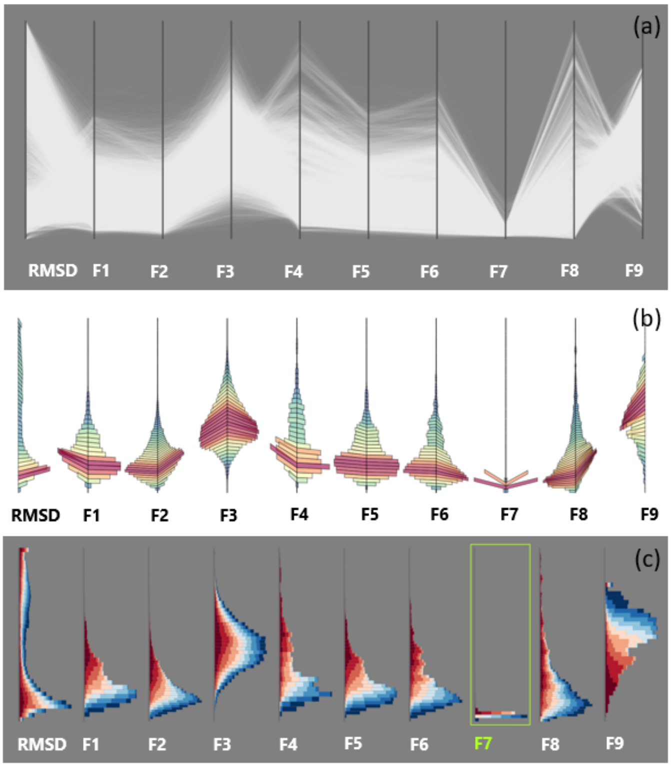


Figure 3.8: CASP dataset rendered in (a) PCP, (b) AH, and (c) PHP. F7 is set as the pivot attribute in PHP. In PHP, the relationship between F7 and other attributes can be discovered by how the colors spread in the histograms, whereas in the other visualizations such discovery is hindered by the skewed distribution of F7.

tograms are tilted, likely making it hard to derive their exact distribution [21]. To deal with this limitation, AH utilizes colors as an additional channel to show the length of each histogram's bars [21]. In contrast, PHP does not distort the distribution and utilizes color as a channel to show the relationship between the pivot and other attributes. The use of the color channel makes it possible to identify the relationship between non-adjacent attributes, in contrast to AH, in which determining the correlation depends heavily on the ordering of axes, as in other PCP-based visualizations [21]. In PHP, users can find information in a more time-efficient manner because there is no need for reordering the attributes to deduce the relationships between them.

PHP can display more attributes in a limited space than AH. PHP renders one histogram for each attribute, but AH renders two histograms for each attribute (excluding the first and last ones), taking roughly double the amount of space to render equally sized histograms. Thus, each histogram of PHP is rendered about two times larger than a histogram of AH. In larger histograms, users can observe subtle patterns or smaller bins more accurately, as well as being able to see whether the difference in histograms is tilted or not. This space efficiency also becomes an issue in SPLOMs when visualizing multiple attributes (Figure 3.8(d)). When visualizing data with n attributes, in SPLOMs $n \times n$ scatterplots are rendered, compared with n histograms in PHP. SPLOMs become highly congested with scatterplots as the number of attributes increases, and each scatterplot becomes smaller, making it harder to observe relationships between attributes. When visualizing multiple attributes, the space efficiency of a visualization is important because it is directly related to how many attributes can be displayed in a limited screen space—i.e., the scalability of a visualization by the number of attributes. Compared to other visualizations, PHP can visualize the relationship between multiple attributes in a more space-efficient manner, benefiting users who aim to find information across multiple attributes.

In the protein dataset, the attribute F7 is radically skewed toward the bottom side of the axis. This skewness affects the performance of PCP-based visualizations. In Figure 3.8(a) and 3.8(b), the nearby lines and histogram bars in PCP and AH, respectively, are drastically slanted toward the lower direction. In contrast, PHP is more resilient to this skewness issue, as the shape of an attribute's distribution is independent of other attributes, unlike

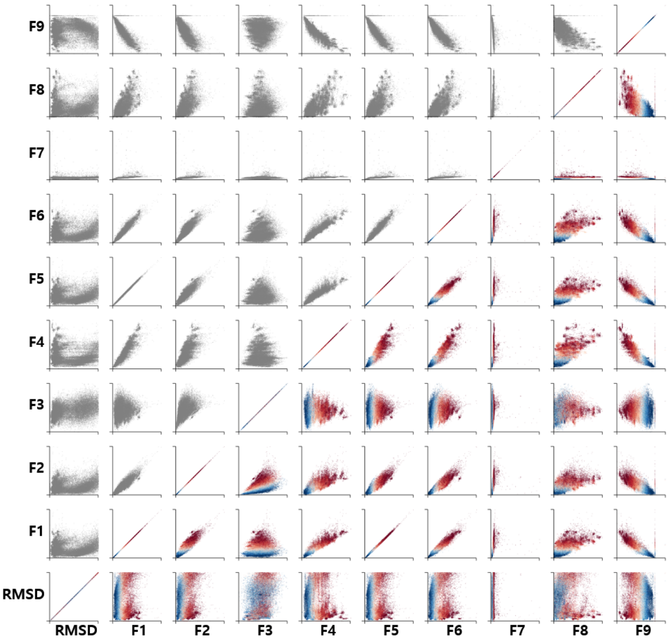


Figure 3.9: CASP dataset rendered as a scatterplot matrix (SPLOM) with the colors of PHP (F7 is the pivot attribute) applied to the scatterplots in the lower-right triangle of the matrix. The colors enable additional discovery in how the items spread out in the context of F7, e.g., the items with high values for F7 (red) gather in distinct regions (right or left regions) in the scatterplots between RMSD and other attributes.

in PCP and AH (Figure 3.8(c)). Moreover, correlations between the skewed F7 and other attributes can be observed by selecting F7 as the pivot attribute. In this case, the color encoding is determined by the ranking of F7, so the color distributions of all histograms show the relationships between F7 and the other attributes, not affected by the skewness of F7. In Figure 3.8(c), F1, F2, F4, F5, and F6 have positive correlations, F3 and F8 do not have a particularly positive or negative correlation, and F9 has a negative correlation with the skewed F7. PHP requires only selecting the skewed attribute as the pivot attribute, whereas other visualizations need further processing of the data (enlarging the visualization, filtering out outliers, logarithmic scaling, etc.) to observe more information.

Utilizing colors in PHP enables the discovery of interesting patterns. In Figure 3.1, the notable red colors in the upper region of the attribute F9 (circled in red) indicate that the data items in that region do not follow the negative correlation between F2 and F9. The same information is almost impossible to obtain from AH or PCP because the attributes F2 and F9 are not adjacent to each other. While a SPLOM can show all pairwise relationships at once, it is also hard to find patterns in SPLOMs (Figure 3.9) because each scatterplot is not rendered large enough owing to the number of attributes, and such interesting data items do not stand out as colors as in PHP. A focus+context technique or a simple interaction, such as selecting a scatterplot of interest to be shown as an enlarged inset, could be employed in SPLOM to mitigate this problem. In PHP, such a small group of interesting data items can be selected and displayed as polylines, as in the polylines of Figure 3.1. Because only a small portion (about 1% of the data) is selected, the items can be displayed without cluttering. Characteristics of the selected items can be observed from the polylines, with the selected items seeming to show a negative correlation between RMSD and F1.

The color mapping used in PHP can also be applied to other visualizations to improve the information-seeking process. One example of this is shown in Figure 3.9, in which the color mapping of PHP (F7 is set as the pivot attribute) is applied to scatterplots in the lower-right triangle of SPLOM. From how the colors spread out in individual scatterplots, additional information related to the pivot attribute can be inferred. For example, from the scatterplot between F4 and F9, it can be observed that items with high-value items of F7 are spread mostly in the upper-left region, while items with lower values of F7 are spread mostly in the lower-right region. This indicates a positive correlation between F4 and F7 and a negative correlation between

F9 and F7, which can likewise also be discovered in PHP. Also, in most of the scatterplots of RMSD and other attributes, it can be observed that items with a high value of F7 (red) are gathered in distinct regions, either the right or the left region (i.e., having high or low values of the corresponding attribute). However, the scatterplot between F3 and RMSD shows a distinct pattern, with items with a high value of F7 being gathered in the middle region of F3 and the upper and lower regions of RMSD.

3.5 User Study

We conducted a controlled user study to assess the performance of PHP in terms of correlation coefficient retrieval. The user study consisted of two within-subject tasks. In the first task, we compared the performance on correlation retrieval between two attributes. In the second task, distance between two attributes was added as a factor to measure how the PCP-based visualizations perform when retrieving the correlation between non-adjacent attributes. The ordering of the two tasks was fixed for all the participants: The first task was performed before the second task.

We selected three visualizations to be compared with PHP: PCP, scatterplot, and AH [21]. PCP was selected as a baseline condition to show the level of improvement of our design. While PHP is an improved version of PCP, its visual cues used to judge the correlation are different (color pattern in PHP vs. line crossing in PCP). We intended to measure the effect of such a difference in visual encoding. Scatterplots were selected because they are commonly used and known to be the best method for visually analyzing the relationship between two attributes. They were used as another baseline for comparison with other techniques. We chose AH among various other improvements of PCP considering that, like PHP, it uses histograms to deal with scalability. Other approaches that use histograms [37, 76, 77] were also considered but were discarded because the visual property of the histograms of these methods does not support correlation retrieval task and requires interactions to derive any correlation between attributes.

3.5.1 Design

For the experiment, we recruited 36 participants from a university's online community (25 males, 11 females; aged 21-33 [mean ± SD: 25.6 ± 2.6]). Participants were screened according to two conditions: (1) participants should be familiar with the statistical terms used throughout the experiment (e.g. Pearson correlation coefficient), and (2) participants should not be colorblind. On average, the user study lasted about 60 minutes. The participants were paid about 10 dollars for their participation. A 27-inch LG monitor (27MP48HQ) was used to display the visualizations for all conditions.

Before performing the tasks, the participants received instructions for each visualization. The instructions included how the visualization is constructed from raw data, and how to interpret the patterns in the visualization to retrieve the correlation. For all visualizations used in all tasks, the interactions were disabled; only the visual encodings were utilized to retrieve the correlation coefficient.

First task: two attributes

In the first task, users were asked to estimate the correlation coefficient (ranging from -1 to 1, with an interval of 0.1) between the two attributes displayed. In PHP, the leftmost attribute was set as the pivot attribute. We recorded the time and error rate of the responses. All responses in the experiment were self-paced, and users typed in their responses.

Two within-user factors were utilized in the experiment: (1) type of the visualization to be displayed (PCP, scatterplot, PHP and AH, with the ordering being determined by a Latin square (4 levels)) and (2) the correlation coefficient set of the data (4 levels). The set of correlation coefficients were defined to have 4 levels: ±[0.9, 0.8, 0.7, 0.6] and ±[0.5, 0.4, 0.3, 0.2, 0.1]. Each set will be referred to HP, HN, LP, and LN, representing high positive, high negative, low positive, and low negative coefficients, respectively.

In this task, we used randomly generated data from a normal distribution with a fixed size of 1,000 items with two attributes for each correlation coefficient set. A coefficient value was randomly chosen from a predetermined set of correlation coefficients. A pivot attribute was first generated with a normal distribution. Then, the other attribute was generated to follow the chosen coefficient value with the pivot attribute. The actual correlation coefficient of the generated data (pivot and other attributes) was slightly different from the chosen coefficient as noise was added during data generation; however, we ensured that this difference did not exceed 0.025. For each combination of visualization method (4 levels) and correlation coefficient set (4 levels), the coefficient estimation experiment was repeated 5 times. Thus, a total of $4 \times 4 \times 5 = 80$ responses was collected.

Prior to the main task, training sessions were given to the participants. The training session had the same conditions as the main task, but the response was not recorded, and the participants could check the answer and train themselves. A training session consisted of 12 responses, and users could request more training sessions if needed. On average, users performed around 2 to 3 training sessions per visualization.

Second task: multiple attributes

In the second task, 4 attributes were displayed in one of the three visualizations (PCP, AH, and PHP). The users were asked to estimate the correlation coefficient (ranging from -1 to 1, with an interval of 0.1) between the leftmost attribute and one of the other selected attributes. In this task, we did not include scatterplots for comparison because their methodology of displaying multiple attributes (scatterplot matrices) greatly differs from other PCP-based visualizations (PCP, AH, and PHP). We recorded the time and error rate of the responses. All responses in the experiment were self-paced, and users typed in their responses.

Three within-user factors were utilized in the experiment: (1) the type of visualization (PCP, AH, and PHP) (3 levels, with the ordering determined by a Latin square), (2) the correlation set of the target attribute (4 levels [HN, LN, LP, and HP], the same as in the first task), and (3) the position of the target attribute (3 positions excluding the leftmost; the leftmost attribute will be referred to as the pivot attribute, and each position of the target attribute will be referred to as the first, second, and third positions from the left).

In this task, we used randomly generated data from a normal distribution with a fixed size of 1,000 items with 4 attributes as in the first task. A pivot attribute was first generated with a normal distribution. Then, the other three attributes were generated according to the pivot value. When the attribute was not the target of correlation retrieval, it was generated to have a random correlation (between -1 and 1) with the pivot attribute. When the attribute was the target of retrieval, the data was generated in the same way as the target attribute in the first task. For each combination of the target data (3 levels) and correlation coefficient set (4 levels), the coefficient estimation experiment was repeated 3 times. Thus, a total of $3 \times 4 \times 3 = 36$ responses were collected per visualization, and thus a total of 36 × 3 = 108 responses was collected in the task.

Prior to the main task, training sessions were given to the participants. The training sessions had the same conditions as the main task, but the response was not recorded and participants could check the answer and train themselves. A training session consisted of 12 responses, and users could request more training sessions if needed. On average users performed around 1 to 2 training sessions per visualization.

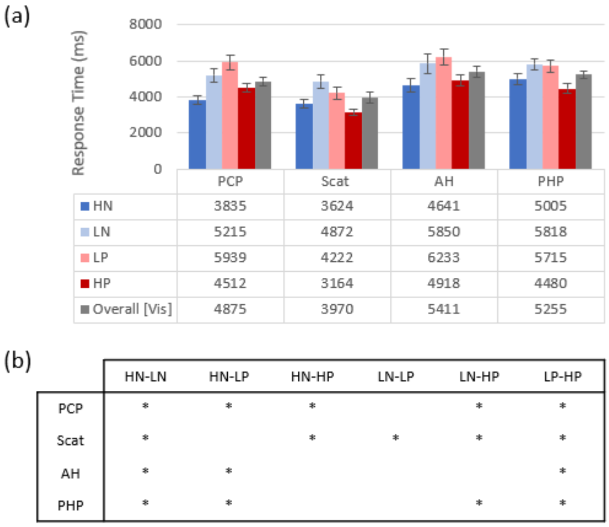
3.5.2 Results

In both tasks, we recorded the task completion time (i.e., the time between the appearance of a visualization and the user's answer in milliseconds) and the error rate of each user's answers (i.e., the absolute difference between the user's response and the chosen coefficient).

First task: two attributes

The task completion time and error rate were analyzed using a 4 × 4 (4 visualization methods × 4 correlation coefficient sets) repeated measures ANOVA. Bonferroni's pairwise comparison was used for all post hoc tests.

Task completion time: Figure 3.10(a) shows the task completion time of all correlation coefficient sets for each visualization method. There was a significant main effect by visualization type ($F_{3,35}$ = 13.207, $p < .001$). Post hoc tests revealed that the task completion time of scatterplots (mean ± SD: 3,970 ± 265 ms) was significantly lower than the task completion times of all other conditions (PCP: 4,875 ± 271; AH: 5,411 ± 354; PHP: 5,255 ± 266). We also found a significant main effect by correlation coefficient set ($F_{3,35}$ = 35.310, $p < .001$), with post hoc tests showing that the participants responded to



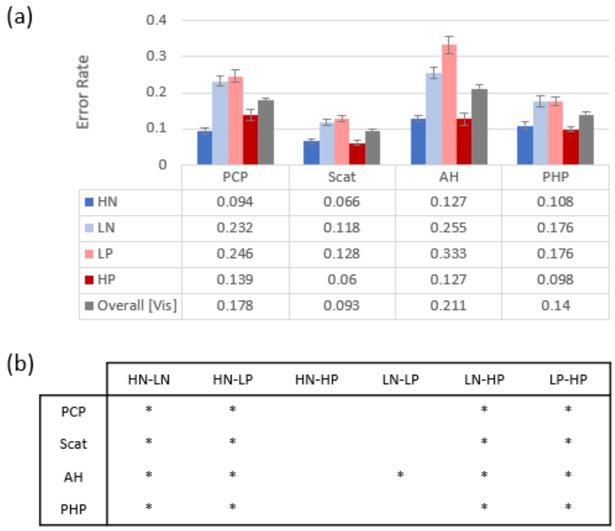
	PCP	Scat	AH	PHP
■ HN	3835	3624	4641	5005
■ LN	5215	4872	5850	5818
■ LP	5939	4222	6233	5715
■ HP	4512	3164	4918	4480
■ Overall [Vis]	4875	3970	5411	5255

	HN-LN	HN-LP	HN-HP	LN-LP	LN-HP	LP-HP
PCP	*	*	*		*	*
Scat	*		*	*	*	*
AH	*	*				*
PHP	*	*			*	*

Figure 3.10: Results regarding task completion time in the first task. (a) Results by visualization method and correlation coefficient set. Error bars indicate the standard deviation of the measured mean. (b) Significance of the difference between correlation coefficient sets for each visualization. An asterisk (*) in the table indicates that the pairwise difference is significant ($p < .05$).

the HN (4,276 ± 229) and HP (4,268 ± 200) conditions significantly faster than to the LP (5,439 ± 323) and LN (5,527 ± 300) conditions. This indicates that the participants took less time to respond to more strong patterns with positive/negative correlations.

There was also an interaction effect between visualization type and correlation coefficient set ($F_{9,35}$ = 3.312, p = .001). For further analysis, we performed a one-way repeated measures ANOVA (4 correlation coefficient sets) for each visualization method. The result of the pairwise comparison of the four correlation sets are shown in Figure 3.10(b). Each visualization showed a slightly different trend. In PCP, HN outperformed all other conditions, mostly because the crossing patterns were most distinct in that condition.



	PCP	Scat	AH	PHP
■ HN	0.094	0.066	0.127	0.108
■ LN	0.232	0.118	0.255	0.176
■ LP	0.246	0.128	0.333	0.176
■ HP	0.139	0.06	0.127	0.098
■ Overall [Vis]	0.178	0.093	0.211	0.14

	HN-LN	HN-LP	HN-HP	LN-LP	LN-HP	LP-HP
PCP	*	*			*	*
Scat	*	*			*	*
AH	*	*		*	*	*
PHP	*	*			*	*

Figure 3.11: Results regarding error rate in the first task. (a) Result by visualization method and correlation coefficient set. Error bars indicate the standard deviation of the measured mean. (b) Significance of the difference between correlation coefficient sets for each visualization. An asterisk (*) in the table indicates that the pairwise difference is significant ($p < .05$).

On the other hand, because there are no crossing patterns in PHP, such a trend did not appear for PHP.

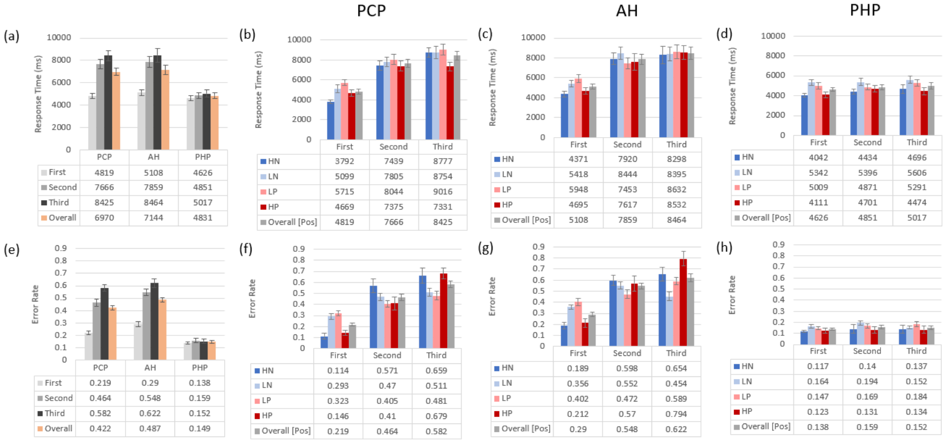
Error rate: Figure 3.11(a) shows the error rate of all correlation coefficient sets for each visualization method. There was a main effect by visualization type with regard to the accuracy of the responses ($F_{3,35}$ = 46.618, $p < .001$). From post hoc tests, it was found that the error rate of scatterplots (mean ± SD: 0.093 ± 0.005) was significantly lower than the error rates of all other conditions (PCP: 0.178 ± 0.007; AH: 0.211 ± 0.011; PHP: 0.140 ± 0.007). In addition, the error rate of PHP was significantly lower than the error rates of PCP and AH. There was also a significant main effect by correlation coefficient set ($F_{3,35}$ = 100.049, $p < .001$). Post hoc tests indicated that the error rates in the HN (0.099 ± 0.004) and HP (0.106 ± 0.006) conditions were significantly lower than those of the LN (0.195 ± 0.007) and LP (0.221 ± 0.009) conditions. Furthermore, LN showed a significantly lower error rate than LP.

An interaction effect between visualization type and correlation coefficient set was observed ($F_{9,35}$ = 7.385, $p < .001$). For further analysis, we performed a one-way repeated measures ANOVA (4 correlation coefficient sets) for each visualization method. The result of the pairwise comparison of the four correlation sets is shown in Figure 3.11(b). Only in AH did LP significantly underperform LN, whereas the other visualizations did not show such notable differences.

Second task: multiple attributes

The task completion time and error rate were analyzed using a 3 × 4 × 3 (3 visualization methods × 4 correlation coefficient sets × 3 positions of target attribute) repeated measures ANOVA. Bonferroni's pairwise comparison was used for all post hoc tests.

Task completion time: There was a significant main effect by visualization type ($F_{2,35}$ = 23.702, $p < .001$), with post hoc tests showing that the task completion time of PHP (mean ± SD: 4,831 ± 244 ms) was significantly lower than the task completion times of the other two methods (PCP: 6,970 ± 331; AH: 7,144 ± 419) (Figure 3.12(a)). We also observed a main effect by correlation coefficient set ($F_{3,35}$ = 11.656, $p < .001$). The response was significantly faster in the highly correlated conditions (HN: 5,974 ± 267; HP: 5,945 ± 293) than in the other two conditions (LN: 6,675 ± 276; LP: 6,664 ± 287). Position of target attribute also showed a significant main effect ($F_{2,35}$ = 64.835, $p <$ .001). The pairwise differences in task completion time between any two positions were all significant, while the response time increased as the distance



(a)

	PCP	AH	PHP
First	4819	5108	4626
Second	7666	7859	4851
Third	8425	8464	5017
Overall	6970	7144	4831

(b)

	First	Second	Third
HN	3792	7439	8777
LN	5099	7805	8754
LP	5715	8044	9016
HP	4669	7375	7331
Overall [Pos]	4819	7666	8425

(c)

	First	Second	Third
HN	4371	7920	8298
LN	5418	8444	8395
LP	5948	7453	8632
HP	4695	7617	8532
Overall [Pos]	5108	7859	8464

(d)

	First	Second	Third
HN	4042	4434	4696
LN	5342	5396	5606
LP	5009	4871	5291
HP	4111	4701	4474
Overall [Pos]	4626	4851	5017

(e)

	PCP	AH	PHP
First	0.219	0.29	0.138
Second	0.464	0.548	0.159
Third	0.582	0.622	0.152
Overall	0.422	0.487	0.149

(f)

	First	Second	Third
HN	0.114	0.571	0.659
LN	0.293	0.47	0.511
LP	0.323	0.405	0.481
HP	0.146	0.41	0.679
Overall [Pos]	0.219	0.464	0.582

(g)

	First	Second	Third
HN	0.189	0.598	0.654
LN	0.356	0.552	0.454
LP	0.402	0.472	0.589
HP	0.212	0.57	0.794
Overall [Pos]	0.29	0.548	0.622

(h)

	First	Second	Third
HN	0.117	0.14	0.137
LN	0.164	0.194	0.152
LP	0.147	0.169	0.184
HP	0.123	0.131	0.134
Overall [Pos]	0.138	0.159	0.152

Figure 3.12: Performance evaluation results of the second task. Error bars indicate the standard deviation of the measured mean. (a) Response time of the visualization by position of the target attribute. (b)-(d) Response time of each visualization by position of the target attribute and correlation coefficient set ((b) PCP, (c) AH, and (d) PHP). (e) Error rate of the visualization by position of the target attribute. (f)-(h) Error rate of each visualization by position of the target attribute and correlation coefficient set ((f) PCP, (g) AH, and (h) PHP).

between the pivot and the target attribute became bigger (first: 4,851 ± 178; second: 6,792 ± 294; third: 7,302 ± 372.

Interaction effects were also observed. Visualization type and position of target attribute showed a significant interaction effect ($F_{4,35}$ = 20.798, $p <$ .001), as did correlation set and position ($F_{6,35}$ = 2.995, p = .008). For further analysis of the interaction effects, we performed a 4 × 3 (4 correlation coefficient sets × 3 positions of target attribute) repeated measures ANOVA for each visualization. As shown in Figure 3.12(b)-(d), in PCP and AH, correlation coefficient set and position of target attribute both showed a significant main effect in addition to the interaction effect between them. Meanwhile, in PHP, only correlation coefficient set showed a significant main effect, whereas task completion time was not affected by position of target attribute.

Error rate: There was a significant main effect by visualization type ($F_{2,35}$ = 144.112, $p < .001$). Post hoc analysis revealed that the error rate of PHP (mean ± SD: 0.149 ± 0.011) was significantly lower than the error rates of the other two visualizations (PCP: 0.422 ± 0.018; AH: 0.487 ± 0.018) while PCP significantly outperformed AH (Figure 3.12(e)). Position of target attribute also had a significant main effect ($F_{2,35}$ = 122.976, $p < .001$). According to the post hoc analysis, all position pairs showed a significant difference, while the error rate increased as the distance between the pivot and target attributes increased (first: 0.215 ± 0.010; second: 0.390 ± 0.014; third: 0.452 ± 0.016). No significant main effect by correlation coefficient set was observed ($F_{3,35}$ = .029, p = .993).

Multiple interaction effects were also observed. Interaction effects between visualization type and position of target attribute ($F_{4,35}$ = 24.503, $p <$ .001), between correlation coefficient set and position of target attribute ($F_{6,35}$ = 15.351, $p < .001$), and between all of the three within variables ($F_{12,35}$ = 4.827, $p < .001$). For analysis of the interaction effects, we performed a 4 × 3 (4 correlation coefficient sets × 3 positions of target attribute) repeated measures ANOVA for each visualization. As shown in Figure 3.12(f)-3.12(h), in PCP and AH, we observed a significant main effect by position of target attribute and the interaction effect between it and correlation coefficient set. By contrast, in PHP, only correlation coefficient set showed a significant main effect, implying that position of the target attribute did not play a significant role in the performance regarding accuracy.

3.6 Discussion

The first task shows that in terms of the accuracy of the responses, PHP outperforms PCP and AH, but PHP is outperformed by scatterplots in the correlation coefficient estimation task. We suspect that the performance difference comes mainly from the innate difference in the effectiveness of the visual encodings, i.e., crossing patterns in PCP and AH, color in PHP, and position in scatterplot. Other factors could have affected the performance. While training could offset the effect, the well-known scatterplot might have advantages over the other unfamiliar visualizations. Fatigue from the first task may have negatively affected the performance in the second task, in addition to the second task being relatively more complicated than the first task. There was mostly no tradeoff between response time and error rate (faster performance does not increase the error rate). One exception to this was a faster response time in scatterplots under positive correlation conditions compared with negative conditions. In scatterplots, the response time of HP was faster than that of HN, and LP was faster than LN. But there was no significant difference in the performance between the two pairs. Although we have no empirical evidence, we suspect that the difference in response time is caused by participants' being more familiar with scatterplots with positive correlations. We think a more thorough analysis of this issue can be a potentially interesting future topic.

Results of the second task show that the positioning of attributes in PHP does not influence the performance of the correlation retrieval task, unlike other conditions in which the performance severely decreases when the target and pivot attributes are not adjacent. The empirical results imply that PHP mitigates one of the two main innate limitations of PCP we previously stressed—i.e., the difficulty in estimating relationships between non-adjacent axes. AH, which that also utilizes histograms to deal with scalability did not outperform PCP and performed worse than PHP. Crossing and cluttering of bars remain in AH, even though histograms are used to deal with the scalability issue of PCP, implying that AH does not fully overcome the first limitation of PCP we mentioned —i.e., the cluttering of polylines caused by multiple crossings. Compared with AH, PHP utilizes a totally different visual channel, i.e., color, to deal with the cluttering problem, and thus it is free from the cluttering by crossing line patterns. We expect that when the number of items further increases, AH will perform better than PCP because of the effectiveness of histograms in dealing with scalability.

When analyzing multidimensional data, it is a common approach to start by inspecting each attribute individually (1D) and then continue by examining the relationships between two or more attributes in order to obtain insights in higher dimensions [**seo2005rank**]. PHP supports this data exploration process. Each histogram in PHP shows the distribution of one dimension, which is hard to see in PCP or SPLOMs. In PHP, users can select a pivot attribute and observe all the data from the perspective of that attribute using the attribute's colormap. After studying the 1D histograms, users can explore the relationships between two or more attributes using the color mapping applied to all other histograms. The implicit connection via colormapping reveals relationships between the pivot attribute and other attributes. Users can move on to select another attribute as a pivot, group and reorder similar attributes for higher dimensional analysis, or zoom in further to inspect a small group of items of interest in an attribute.

Throughout the paper, we fixed various parameters that could affect the performance of the visualization—e.g., the set of colors of the color scheme, the number of colors used in the color scheme, and the number of bins of the histograms. Measuring the effects of changing these parameters could be an interesting future research direction. Throughout the paper and user study, we used a blue-red color scheme for PHP. Studying how a different color scheme might affect task performance in correlation estimation could also be interesting. In addition, the number of distinct colors was fixed at a relatively small value (10) throughout the paper. The number of discriminable hues mapped onto small, separated regions is known to be moderate, i.e., fewer than 10. While using relatively few colors can still help users grasp the overall trend in the data, it could potentially oversimplify the information in the data, hindering the discovery of more diverse and precise patterns of colors in the visualization. However, such a detailed exploration is possible with the original PCP visual encoding, i.e., polyline representation. Investigating the effect of the number of colors in terms of perceiving a data distribution is an appealing future research topic. Increasing the number of colors could reveal different structures in the data, but it could become harder to discern different colors, and individual bars might become too small to interact with.

The number of bins affects the shape of a histogram, which is related to how the colormap is rendered. We expect that changing the number of bins should not greatly influence users' task performance in estimating correlation, as they examine the overall color distribution. However, since the shape of the colormap changes, it could influence some tasks, such as finding outliers or a group of similar items. Since categorical attributes do not carry any ordering information, our rank-based approach cannot be directly applied to categorical attributes. It would be interesting to study how to harmoniously combine the ranking channel and the identity channel in using color mappings for multidimensional data analysis. Also, while we proposed various approaches to dealing with skewed histograms, such as using colors based on ranking or utilizing two-level zooming interactions, they all require some level of user input. Combining the proposed approaches with other analytic methods (e.g., log transformation) that deals with the skewedness of a distribution would be an interesting direction. Finally, PHP can be integrated with other related visualizations similarly to how PCP has been integrated with other visualizations (e.g., scatterplots).

3.7 Summary

We introduced PHP, a novel visualization technique designed to overcome the innate limitations of PCP. PHP utilizes color-coded, stacked-bar histograms to show the relationships between attributes without the issue of cluttering and regardless of the distance between the attributes. With PHP, users can discover interesting items using colored stacked-bar histograms: Similar colors gathered in a small region suggest clusters of data items that follow a certain trend, and salient colors from the overall color distribution suggest outliers. In addition, PHP provides interactions to help users investigate the details of histograms in a limited screen space: two-level zooming (i.e., focus+context zooming and clamp zooming), ghost bars, and a UI widget of the color panel. Following the Visual Information-Seeking Mantra, polylines are used to display the details of focused data, while color-coded histograms provide the overview. We demonstrated how PHP can be used on a real-world dataset in a use case. We also tested the performance of PHP in correlation coefficient estimation tasks. The results showed that PHP correlation estimates were consistent regardless of the distance between attributes.

Chapter 4

Interactive Visualization System for Monitoring Issues in Industrial Software Development

This chapter introduces IssueML, a visualization system for monitoring the related information in multiple issues that occur during development of large projects in industrial environments. This research mainly addresses the second research question (RQ2) of the dissertation, of visualizing multiple, complicated attributes in data items a scalable manner using colors.

4.1 Motivation

In industrial fields where large projects are collaboratively developed, it is natural for many errors to occur during the process. To efficiently manage errors, symptoms and conditions during the occurrence are organized as issues. Since issues correspond to each errors, project managers utilize them as milestones to be resolved to ensure the progression of the development process. Hence, one of the primary responsibilities of the managers is to monitor and manage ongoing issues and their related developers responsible for resolution. However, monitoring issues is not a simple task. Not only are the number of errors significant, but errors also occur in complicated and unexpected patterns that are proportional to the size of the project. Furthermore, information in issues is frequently updated over time due to newly revealed information during the resolution process, making monitoring even more challenging.

We introduce IssueML, a visualization system that facilitates the monitoring of multiple issues generated from large projects. In designing IssueML, we conducted interviews with industrial insiders to identify domain problems related to managing multiple issues. From the interviews, we define two critical tasks in monitoring issues: managing the related developers and visualizing the progress over time. Based on these tasks, IssueML consists of multiple specialized views that support the visualization of multiple issues in a space-efficient and intuitive manner, with respect to their related developers and progress over time. These views are designed by following the Visual Information Seeking Mantra [71], enabling users to steer their data exploration, starting from an overview of overall issues and progressing to the details of a selected subset of issues.

4.2 Background

4.2.1 Interview

We conducted interviews with insiders from an electronics company to gain a deeper understanding of how industry professionals handle multiple issues, and the related problems they have when dealing with the issues. The interviews involved four managers, responsible for managing their subordinate developers, and five developers, responsible for actually resolving the assigned issues. The interviews were conducted in a hybrid (either remote or in person) fashion, depending on the individual conditions. We used a semi-structured interview format, in which we observed their working routines in monitoring or resolving issues, and then asked open-ended questions based on the observation. Topics were mainly focused on questioning about previous limitations in dealing with issues, and the hopes they had functioned they hoped to further understand the difficulties in dealing with issues. On average, the interview lasted about 1 hour. Based on the interviews, we summarize the related background domain problem in dealing with issues, and identified three primary tasks in monitoring multiple issues.

4.2.2 Domain Situation

The electronics company manufactures software and hardware for smart TVs, with the collaboration of multiple departments. To prevent errors in the final product, there is a dedicated test department in which various functions related to the product are tested. When error occurs during the test, the symptoms and related attributes are recorded as issues, and such issues are allocated to developers to be resolved. As the errors are detrimental in the release of the product, issues should be resolved as soon as possible. Thus, one of the critical job of managers is to manage and reduce the issues allocated to their subordinate developers, by consistently monitoring the related issues. An overview of how issues are generated and resolved is provided in Figure 4.1.

However, since the cause of the error is unknown at the point when the error is discovered, and testers are not directly involved in the development of the product, issues are often wrongly allocated to developers. In this case, de-

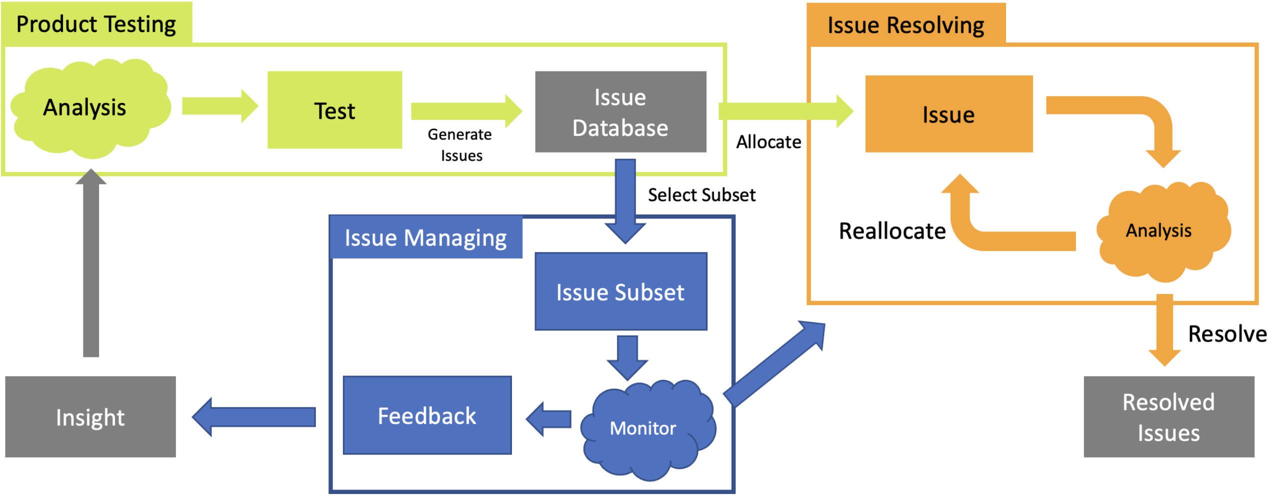


Figure 4.1: Diagram depicting the lifecycle of issues. Issues are generated according to planned product tests, and allocated to developers for resolution. Managers monitor subset of the issues related to them, and provide feedbacks to future tests and aid the resolution task. In this research, we mainly focus on providing the monitoring aspect of providing information of multiple issues in visualization.

velopers should inspect upon the related information and update the earned information in the issues, and pass on to other developers who are potentially related to the issue. As a result, issues contain history of multiple trial and errors in its fields and comments, which is pivotal in understanding and resolving the data. Followingly, observing related information in issues is a critical stage in resolving (or passing on) the issue. But as the information in multiple fields are updated over time, it is time-consuming and difficult to interact with each of the issues. Such problem is more severe in the perspective of managers who need to monitor multiple issues allocated to multiple developers. Moreover, it was hard to interact with such changes in the first place; even though the changes in fields were logged, they were not directly provided to the users. Based on such background, we identified three main tasks, mainly focusing on the of task of managers who need to monitor multiple issues related to their subordinate developers.

4.2.3 Task

Task 1: Identifying developers' status

One of the primary roles of managers is to review and monitor developers' involvement in the assigned issues. Although managers could refer to the values of selected fields (e.g., the *assignee* field), interviews revealed that solely relying on the values may be misleading. Values of fields may change over time during the resolution process; for example, the assignee field frequently changes over time as the cause of the issue is updated. To address this issue, IssueML provides visual cues that enable managers to monitor developers' status across state changes.

Task 2: Identifying updates in issues

Monitoring issues requires closely observing how the issue progresses over time. In the past, managers had to manually read through most comments left by developers to track changes while understanding the context of the issues, which was inconvenient and required a significant amount of time. Alternatively, they could rely on the internal issue management system [39], which only supported limited tasks for monitoring issues. Log data contained all of the update histories but was often very large and unstructured, making it difficult for users to analyze, and thus was not utilized in the monitoring task. To address these challenges, we designed an effective visualization system that efficiently summarizes the progress of issues over time, following the Visual Information Seeking Mantra. This approach allows managers to quickly and effectively monitor issues, starting from the overview of the whole issues to the details of the focused issues.

Task 3: Supporting Discovery of Similar Issues

In analyzing the cause of the issue, we discovered that developers and managers often refer to previous issues in resolving the current issue from the interviews. By referring to the information of previous issues with similar symptoms, users can earn hints on how to resolve the issue or ask help to the developer who previously resolved the issue, speeding up the resolution process. However, it was hard and cumbersome to find the similar issues, as users had to manually search for the issues by matching keywords, hoping that the similar issues appear in the search results which they had to individually inspect upon. To address this limitation, we implemented a NLP based approach that automatically searches for similar issues, overcoming the limitations in direct string matching. Users can further control the results by applying filters according to multiple fields in the issue. Also, to reduce the burden of users in inspecting upon each of the similar candidates, in IssueML we provide visual cues for quickly comparing between the original and the candidate issue.

4.3 Issue Data

In this section, we outline the process of utilizing and modifying the original issue data to meet the requirements of our tasks and the visualization design of IssueML. The issue data we utilized is based on the Atlassian JIRA [39] software of generating and managing issue data which is currently used as the issue management tool in the electronics company we collaborated with.

Firstly, among multiple fields, we extracted critical fields that managers and developers commonly referred to when dealing with the issues based on the interviews, as in Table 4.1. In most cases, managers first searched issues according the the *assignee* field to retrieve issues of interest. Then in the

Field (key)	Explanation	Type
description	Symptoms and conditions of the test in the error	string
summary	Summary of description field	string
assignee	ID of currently assigned developer	string (id)
resolution	Resolution of the issue. default is null	categorical
priority	Priority of the issue (major, minor, ...)	categorical
status	Current status of the issue (Open, Closed, FixReady...)	categorical
duedate	Deadline of the issue to be resolved	date
comment	Comments left by developer/managers. Object contains the time, author, and the details of the comment	object

Table 4.1: Major fields in the issues utilized for IssueML.

Field (key)	Explanation	Type
author	ID of person who changed the values	string
time	Time of the changes occurred	date
items	Information of the changed fields. Object contains each of the updated keys and its value before/after the update	object

Table 4.2: Structure of each record consisting the overall log data

subset of issues, they checked if the issues are being resolved according to the assigned due date, referring to the *resolution*, *status* and *duedate* field. Finally, after referring to the *summary* field, if further inspection of the details were required, users referred to the *description* and *comment* field.

There were two main limitations in the previous interaction process. Firstly, while the changes in fields aids users in finding the cause of the issue, such was hard to be utilized due to limits in accessing and interacting the large log files. To overcome the difficulties related to interacting with the large and complicated log data, in IssueML we aimed at providing an effective overview of the log data. Log of the issue data are consisted of multiple records in which changes in multiple fields in a single instance are recorded. (The structure of a single record is described in Table 4.2.) However, not all of the changes in the fields were considered important. Thus, in IssueML we distinguish records that involve changes in 'important' fields (*assignee*, *resolution* and *status*) from other records to assist users to interact with changes in important fields more effectively.

Secondly, since it was previous complicated to track the history of previous changes in the fields, managers could not track issues that their subordinate developers were previously assigned, but are not currently assigned. As a result, managers could not fully track the developer's contribution in

Status	Condition	Explanation
indirect	!(*currently assigned*) && (*previously assigned*)	Issues that were previously assigned to the developer, but not currently assigned
complete	(*currently assigned*) && *resolution* != *null*	Issues that are resolved in time
ongoing	(*currently assigned*) && *resolution* == *null* && *duedate* > *current*	Issues that are not resolved, but did not pass the due date
delayed	(*currently assigned*) && *resolution* == *null* && *duedate* < *current*	Issues that passed the deadline
unrelated	!(*currently assigned*) && !(*previously assigned*)	Issues that are not related to the developer

Table 4.3: Classification of issues according to a developer

resolving issues. In IssueML, to overcome potential misunderstandings in the contribution of developers, we extract the history of changes in the assignee in each of the logs. Then, according to a single developer, we categorize the status of the selected issues with regards to its completion status and the developer's involvement in the issues. The categorization not only reflects previous involvements in the issues, but also simplifies the previous management of multiple developers according to dedicated categories, supporting the effective observation of the performance of developers related to Task 1. An overview of the categories according to a single developer are depicted in Table 4.3.

4.4 IssueML

Based on the identified tasks from interviews, we designed a visualization system called IssueML (Figure 4.2). Previously, users mostly utilized queries to retrieve and interact with multiple issues. The process required repeated modification of the queries to change the selecting, and was ineffective at

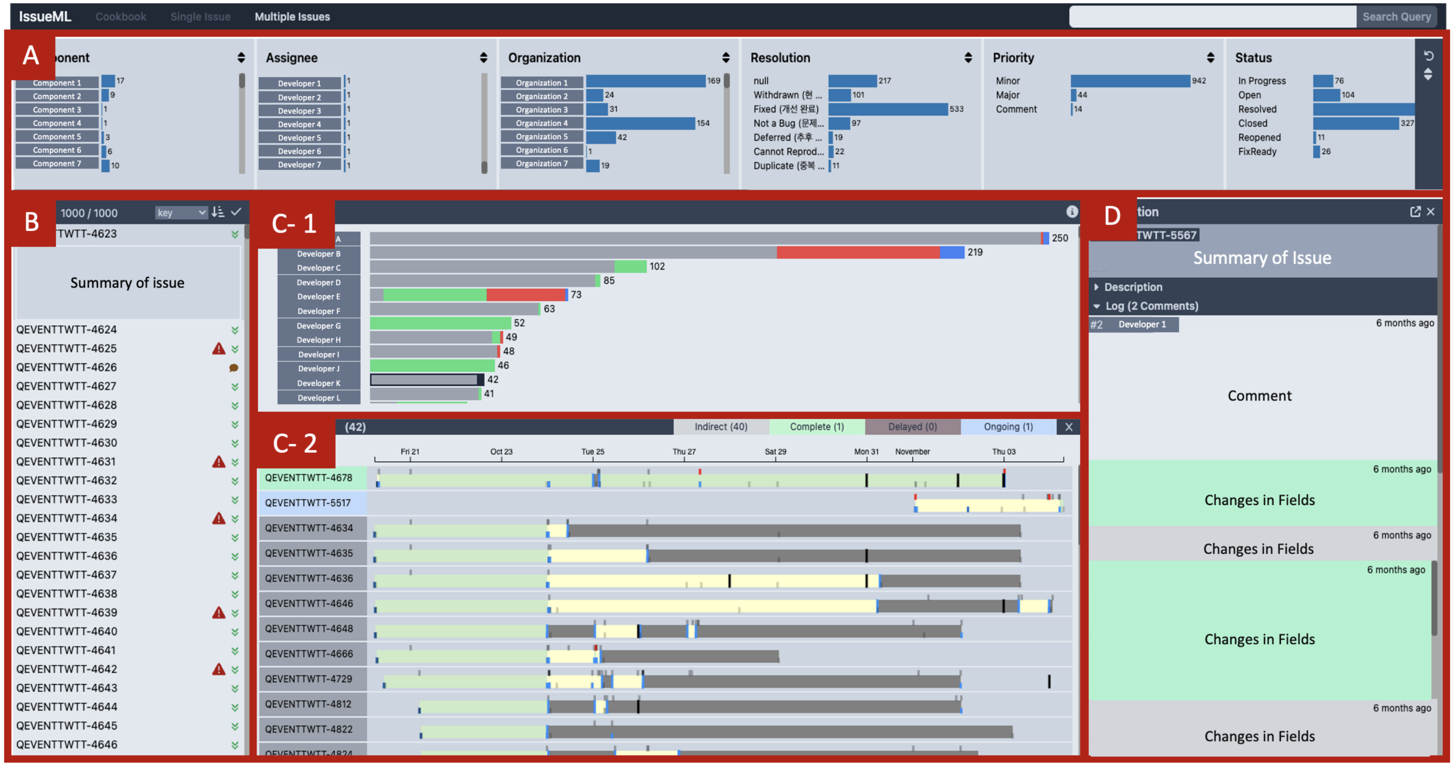


Figure 4.2: IssueML visualizing a real world data. Some fields are censored due to privacy. (a) From the attribute view, users can observe the distribution of each attribute, and select ranges or values of interest to filter the issues. (b) Each of the selected issues is displayed in the issue list view. (c-1) Status of each developers can be easily monitored from the stacked bar chart of analysis view. (c-2) Upon selecting one developer of interest, progress of each of the corresponding issues is visualized. (d) Detail of one selected issue can be observed in the information view.

dealing with fields involving continuous attributes as each of the information had to be manually entered. Reflecting the limitations, IssueMLis consisted of four coordinated views dedicated to effectively monitoring multiple issues in a scalable manner In line with the Visual Information Seeking Mantra, users can explore the data starting from the overview of multiple issues to the details of a single issue, fulfilling the aforementioned critical tasks.

4.4.1 Attribute View

The **attribute view** (Figure 4.2(a)) can be the starting point for the monitoring task of multiple issues. It presents the distribution of each attribute in the issues as either a bar chart (for categorical attributes) or a histogram (for numerical attributes). The attribute view provides an overview of the issue data by offering distributions of each attribute, in which users can select items of interest by applying filters. Previously, in applying filters user had to manually enter the values, which was ineffective and less intuitive. In IssueML users can interactively filter items of interest according to the attribute values by selecting values or ranges in the distributions. When a filter is applied, users can compare the distribution of the overall data to the distribution of the selected issues that are highlighted, as in Figure 4.3. Based on the in-

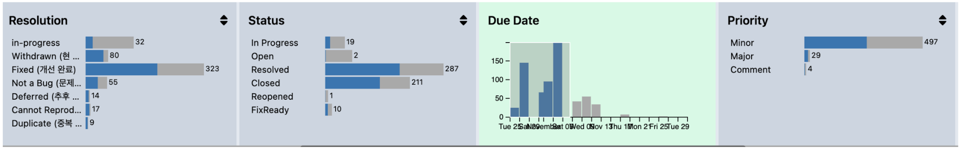


Figure 4.3: Example of the filtering interaction applied in the attribute view. Upon selecting a certain range/categories in the attributes, the distribution of the selected issues can be compared with the overall distribution in the background.

formation, users can easily the observe how each of the filters influence the selection of items, and further elaborate the selection of issues.

4.4.2 Issue List View

In the **issue list view** (Figure 4.2(b)), the selected issues from the attribute view are visualized as a list. Users can swiftly obtain vital information about each issue, such as its priority and whether it has surpassed its due date, from the corresponding icons in each row of items. For further details, users can refer to the summary of each issue by hovering over the list item. Based on such information, users may either click on one of the items to display details of the issue in the information view (Figure 4.2(d)), or manually select an issue of interest and compare its distributions to the overall distribution, as in the same way as selecting items in the attribute view. From the issue list view, users can elaborate the selection of items previously sampled from the attribute view, which can be further analyzed in the analysis view.

4.4.3 Analysis View

After issues of interest are selected from the attribute and the issue list view, their patterns can be analyzed in the **analysis view** (Figure 4.2(c)). In the analysis view, users can observe the progress of each developer and further inspect the issues related to a selected developer with visualizations, fulfilling the main tasks identified from the interviews. The analysis view consists of two components, the developer component (Figure 4.2(c-1)) and the issue log component (Figure 4.2(c-2)).

Developer Component

In the *developer component* (Figure 4.2(c-1)), the distribution of the selected issue, categorized by the involved developers is displayed as a stacked bar chart. To address the user task of managing developers (Task 1), we utilized the classification of issues defined in Table 4.3 for efficient observation of previous and current contributions in the issues as well as its completion status. For each developer involved in the selected issues, we render a stacked bar chart for each of the defined categories outside of *unrelated* issues. Colors are applied to each of the issue groups, enabling users to directly distinguish each of the groups in the bar charts. The colors were deliberately selected to help users quickly distinguish one group from each other. *Gray* indicates indirect issues that were previously assigned to the developer, but not currently assigned, and colored bars correspond to the issues that are currently assigned to the developer (*green*: completed issues, *blue*: ongoing issues that are not overdue, *red*: overdue issues). From the bar charts users can effectively observe the status of each developer and select a developer of interest to be further analyzed in the issue log component.

Issue Log Component

After a developer is selected, corresponding issues are visualized in the *issue log component*. In the component, progress of each issues over time is displayed as a single row of timeline, with visual cues that indicate major events. Figure 4.4 depicts how important progresses in an issue are visualized in the issue log component. From the patterns of colors and ticks in each of the issues, users can intuitively observe how each of the issues have progressed over time. To assist users in effectively perceiving information without feeling overwhelmed, we made a deliberate choice to limit the number of colors

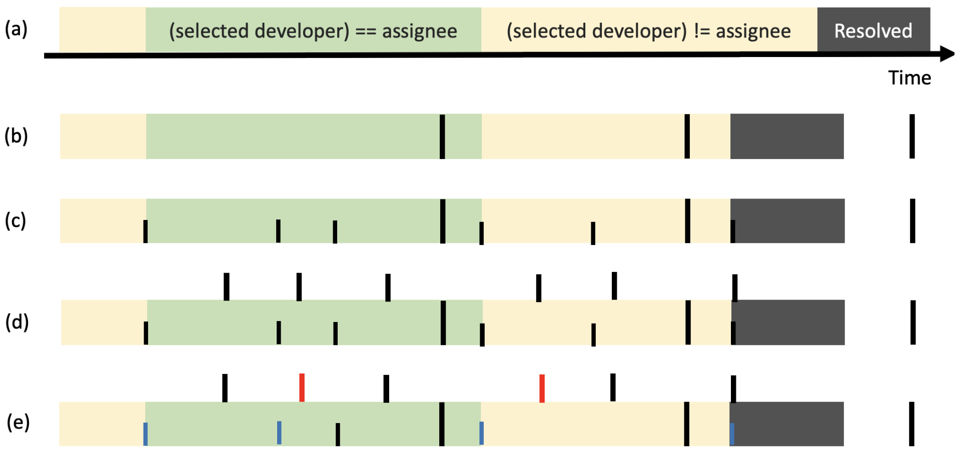
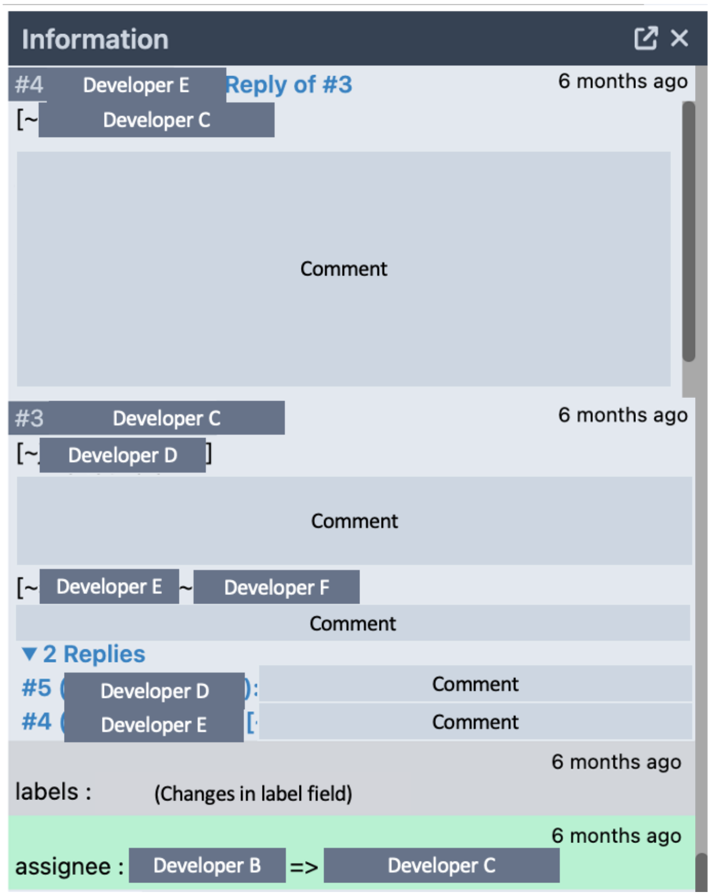

(a)
(selected developer) == assignee
(selected developer) != assignee
Resolved
Time
(b)
(c)
(d)
(e)

Figure 4.4: Step-to-step diagram of visualizing a single issue's information over time in the issue log component. (a) The overall visualization is a timeline, in which colors indicate changes in the assignee, or changes in the issue's resolution status. (b) Long ticks are mapped to due dates. Previous due dates are also identically rendered in the visualization for managers to observe. (c) The lower short ticks represent each records in the log file. (d) The higher short ticks correspond to comments related to the selected issue. (e) Colors refer to special types of events. Red ticks in comments are comments from the selected developer, and blue ticks correspond to records containing changes in important fields previously categorized.

used in the patterns of the visualizations. This approach helps to mitigate color fatigue and ensures that users can focus on the relevant information without being distracted by excessive color variations. With this visualization technique, the previous task of identifying the changes in issues (Task 2), which was not accomplishable unless users read the comments or related logs thoroughly, can be fulfilled by readily finding information from the visual patterns. In addition, based on the visualizations, it is able to interact with multiple issues and find related information by visually comparing the patterns, such as finding groups of similar issues or detecting outliers with outstanding patterns. Furthermore, if an item worth further inspection is dis-



#4 Developer E Reply of #3 6 months ago
[~ Developer C

Comment

#3 Developer C 6 months ago
[~ Developer D]

Comment

[~ Developer E ~ Developer F

Comment

▼ 2 Replies

#5 (Developer D): Comment
#4 (Developer E [Comment

6 months ago
labels : (Changes in label field)

6 months ago
assignee : Developer B => Developer C

Figure 4.5: Example of the threading approach in the comments from an issue in the information view. In comment 3, Developer C mentions Developer D and E, and in comment 4, Developer D replies to Developer C. With threading, users can directly jump to such co-mentioned issues by clicking on the blue text corresponding to each comment.

covered from the patterns, users can hover over the ticks to retrieve its details or interact with further details in the information view.

4.4.4 Information View

In the **information view**, detailed information of one selected issue can be observed. Details of the selected issue over time such as the raw value of each of the comments and histories can be examined in inverse chronological order, prioritizing recent updates. For effective observation of various events, different background colors are applied according to the types (emerald for important records, gray for other records, slate for comments) of the expressed information. Furthermore, to aid the observation of comments, we provided direct links for threaded comments (*i.e.*, pairs of comments where a developer(A) is mentioned by another developer(B) in one comment, and the relationship is reversed in the other comment), in which users can directly jump to related comment without reading all the other comments. With such expressions, users can check upon the details of the discovered information from previous views in the information view. Based on the information, users can either continue the observation of multiple issues, or further inspect upon the details of the selected issue in the issue analysis view.

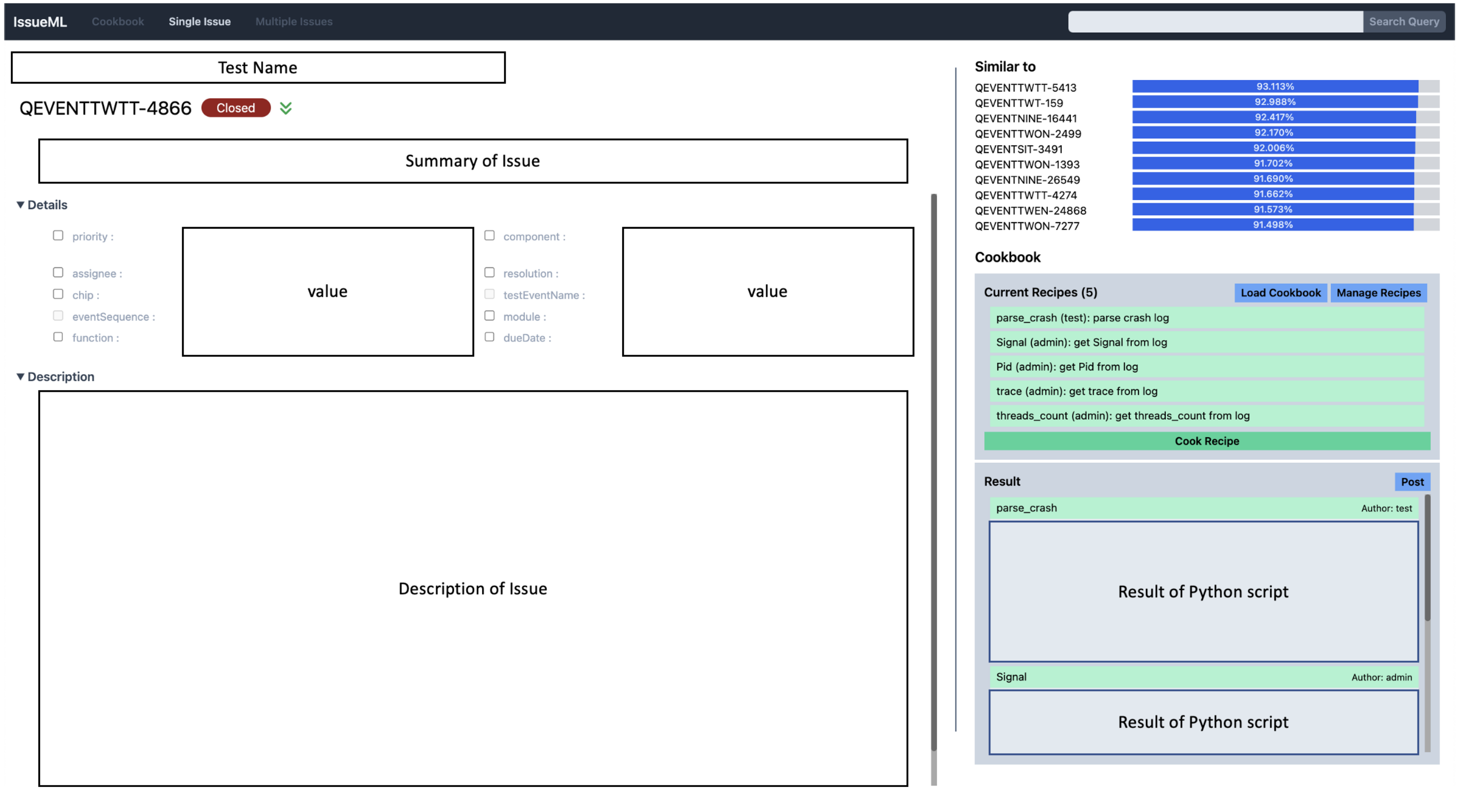
4.4.5 Issue Analysis View

Issue analysis view (Figure 4.6) is a separate view in IssueML dedicated for inspecting and analyzing a single selected issue. In issue analysis view, most of the details of the selected issue, such as the values in each of the fields can be observed in a single screen. Based on the information, the issue analysis view is designed to support the user task (Task 3) of supporting the discovery of similar issues. On the right upper side of the issue analysis view, a list of similar issues and its degree of similarity is provided. This similarity is based on the FastText [6] encoding, where we encode each of the summaries in the issues, and calculate the cosine similarity between the selected and all of the other issues. Among the issues, we display the top 10 similar issues in the analysis view for user to interact with. The provided suggestions based on

NLP techniques enable users to overcome the previous limitation in direct string matching.

Furthermore, in the issue analysis view, users can adjust the result of the similar issues and observe the details of them. By checking on each of fields in the issue of reference, only issues that share the same values in the checked fields appear in the bar charts of similar issues. In addition, to support users in verifying whether the similar issues according to the NLP technique are actually similar to the original issue, users can further compare between the two issues with the help of visualization techniques as in Figure 4.7 Upon selecting one of the candidates, users can compare between the original issue and the similar issue with the help of highlights. Highlights are rendered when text or properties are shared in both of the issues, enabling the effective observation of shared properties.

Finally, to support the analysis of attached log files in issues that are complex and large, the issue analysis view is integrated with a dedicated system for managing scripts for processing logs. In the system, Python scripts for processing logs can be managed and uploaded by developers. Developers can also choose the scripts to utilize for their logs, and run the scripts in the server of IssueML. Then, the results of the scripts are displayed as in the bottom right part of Figure 4.6. The script system enables users to efficiently run scripts without dealing with complicated structures or routines in dealing with the log data. In addition, scripts from multiple developers can be shared via the system from which they can share their know-hows in dealing with issues.



Test Name

QEVENTTWTT-4866 Closed

Summary of Issue

▾ Details

□ priority :

□ assignee :
□ chip :
□ eventSequence :
□ function :

value

□ component :

□ resolution :
□ testEventName :
□ module :
□ dueDate :

value

▾ Description

Description of Issue

Similar to

QEVENTTWTT-5413	93.113%
QEVENTTWT-159	92.988%
QEVENTNINE-16441	92.417%
QEVENTTWON-2499	92.170%
QEVENTSIT-3491	92.006%
QEVENTTWON-1393	91.702%
QEVENTNINE-26549	91.690%
QEVENTTWTT-4274	91.662%
QEVENTTWEN-24868	91.573%
QEVENTTWON-7277	91.498%

Cookbook

Current Recipes (5)

parse_crash (test): parse crash log

Signal (admin): get Signal from log

Pid (admin): get Pid from log

trace (admin): get trace from log

threads_count (admin): get threads_count from log

Result

parse_crash Author: test

Result of Python script

Signal Author: admin

Result of Python script

Figure 4.6: Overview of the issue analysis view. Actual values in the fields are screened out for privacy reasons. In issue analysis view, users can observe the detailed information of the single issue, and interact with similar issues (upper right). Users can also process the log files provided in the issues with the uploaded scripts (lower right).

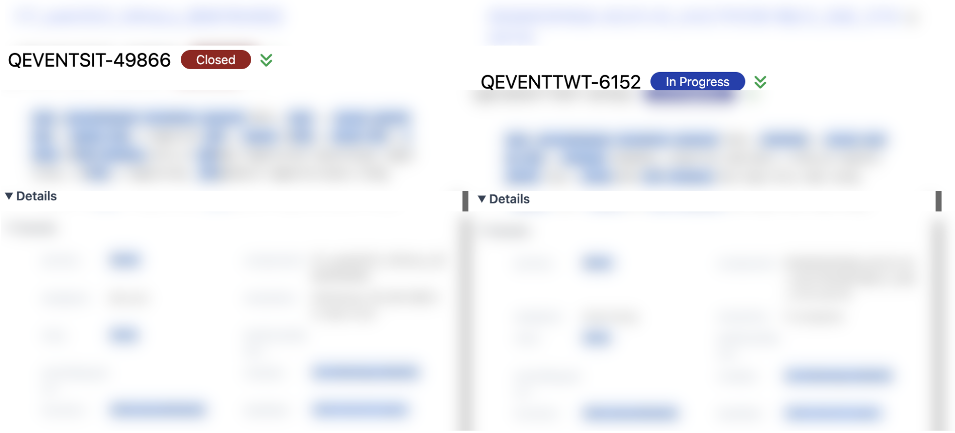


Figure 4.7: When a candidate from the list of similar issue is selected, the selected issue in displayed side to side, and the tokens that are shared in the issues are highlighted for users to compare. (Content is blurred for privacy reasons.)

4.5 Use Case

From IssueML, users can monitor the progress of issues starting from the overview of multiple issues to the details of a singular issue, following the Visual Information Seeking Mantra [71]. Most notably, the visualization technique of displaying the events of a single issue with colors in the issue log component enables the observation of progresses in multiple issues in a single screen. In this section, we introduce various examples of discovering interesting information in multiple issues of real world data. Data used in the section are actual issues generated during the development of smart TVs in an electronics company, containing a total of 187775 issues generated within 3 years (2020.06 - 2022.11).

Figure 4.8 shows an example of IssueML visualizing 1000 of the most recent issues from the overall data. In interacting with multiple issues, it is common to select a subset of interest from the overall data, and begin the monitoring task. In IssueML, the visual filters support users in narrowing down the retrieved issues, as in the example where issues with major priority are selected in Figure 4.8. Among the bar chart of developers in developer component, users can select one developer of interest, such as Developer A with many incomplete issues. Upon selecting, users can observe how the selected issues have progressed over time from each of the colored patterns. In the patterns, it can be discovered that during the time when the selected developer is the assignee of the issue, less activities (corresponding to ticks) occur. In addition, the red ticks corresponding to comments also only appear when the developer is not the assignee of the issue. Such might suggest that the selected developer is closer to the manager of developers, whose main role is to appropriately assign the issues rather than resolving them.

Reflected in the example, as users tend to resolve issues according to their roles and routine, progresses in their assigned issues often resemble each other. Thus, in the patterns of multiple issues, users can find potential issues of interest by searching for issues with outstanding patterns. Figure 4.9 shows the issue log component of a different subset of issues selected from the previously retrieved 1000 items. In the Figure, it can be observed that while most progresses are similar to each other, issues in red boxes tend to have distant green regions. Such issues with different patterns can be the target of inspection, as the different pattern might suggest unexpected events that caused changes in the routine of developers. As a result, the expression of progresses as visual patterns not only enables the observation of individual issues, but also aids users in understanding how the developers deal with the issues, and further supports the discovery of distinct patterns which users can prioritize in their observation.

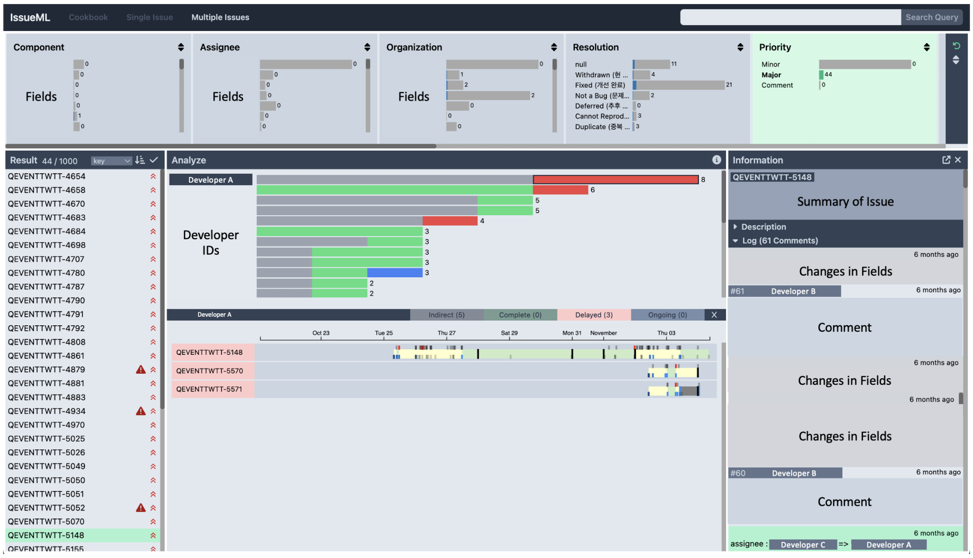


Figure 4.8: Example of interacting with multiple issues in IssueML. In the Figure, a total of 1000 issues generated during the development of smart TVs are displayed. Users can filter issues according to various characteristics from the attribute view (in the Figure, issues with major priorities are selected) and select an developer of interest in the developer component of IssueML (in the Figure, Developer A is selected). Upon selecting one developer, users can observe how multiple issues progress based on the color patterns. Patterns reveal that the selected developer did not interfere a lot in the issues when the issue was assigned to him/her. (Values in some fields are masked for privacy reasons)

4.6 Discussion

The approach of visually encoding the information of multiple issues in IssueML provides a novel perspective in the issue monitoring task. Previously, managers responsible for the monitoring did not, and could not have a clear task with multiple issues, as dealing with information in each of the singular issues itself was challenging and overwhelming. However, with the visual approach of IssueML, such complicated process can be overcame. Following the Visual Information Seeking Mantra, managers can effectively control the target issues to be inspected on. Then, with the visual patterns expressing the records in an issue, managers can quickly figure out on which records to

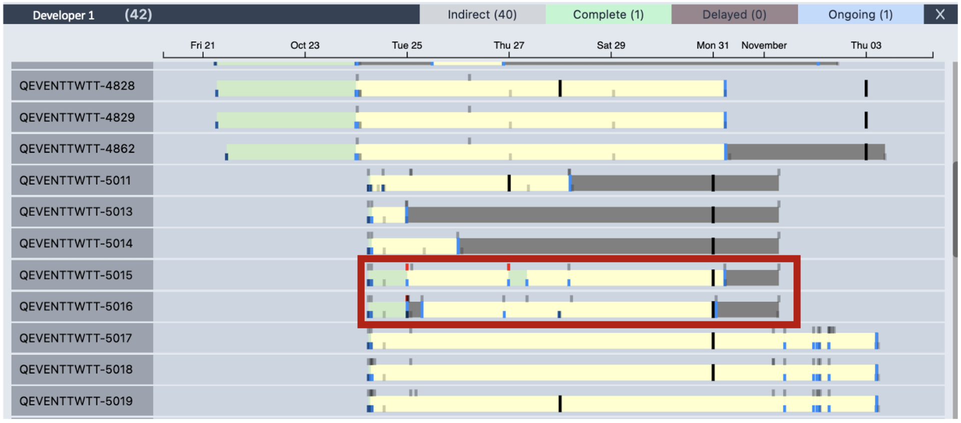


Figure 4.9: Another example of interacting with multiple issues in IssueML in the issue log component. Among the patterns of multiple issues, users can discover distinct patterns (red boxed issues) indicating atypical progresses during the resolution, from which they can begin the inspection on.

focus instead of reading the entire details. Such approach enables effective interaction with progresses of multiple issues that was previously unavailable, as reflected in the use case. We believe that our approach in visualizing issues can shed light on new possible tasks that were unavailable due to the sheering complexity in observing the information multiple issues. To further discover related tasks, we are currently deploying IssueML in actual work environments, and plan to collect feedback from various users. Based on iterative feedback with them, we hope to further extend the functions and tasks related to interacting with multiple issues in IssueML.

One of the future directions is to further increase the scalability in visualizing multiple issues. Currently, an issue is expressed as a row of colors, and users can observe the changes in multiple issues from the multiple rows. In this approach, the number of issues users can interact with is bound by the number of rows a single screen can accommodate. However, in the use case, we discovered that many issues that are normally resolved follow a similar pattern, which we think can be visualized as a row indicating groups of similar trends rather than multiple rows that damage scalability. By reducing the total number of rows with grouping, users will be able to interact with even more items effectively, and discover interesting patterns more easily with the increased capacity. Likewise, we hope to discover various criteria in which the issues can be grouped by, and find appropriate expressions that can envelope the information in multiple issues.

Furthermore, we hope to develop new visual methodologies for expressing how the multiple issues are related to each other, for a better understanding of the overview. Currently, the information of multiple issues are categorized by the related developer or assignee. While this approach fulfills the user task of monitoring each of the developers, it is limited in observing the details of the issue which users still have to individually inspect upon. We believe that providing the overview in how the issues are related to each other will further help managers in effectively understanding the current situation related to errors. Based on the gained information, managers may infer to the causes of the errors without inspecting on each of the issues, and apply appropriate measures accordingly. However, since the row-based visualization for each of the issues may be limited in expressing the relationship between issues, we plan to design visualizations specialized for expressing the relationship, such as a scatterplot.

Finally, we wish to extend the system to enable users interact with more complicated and detailed information in the issue data, further supporting the analysis and resolution task. For example, in the current system, we mostly relied on the summary of the issue for finding similar issues. However, the summary and description field are mostly generated during the test, and thus cannot reflect updates that occur during the resolution process. Thus, we plan to further integrate comments (which can reflect recent updates) and log data (which contains most information related to the error) into the components of IssueML to additionally support users in interacting with the information of multiple issues. Since such data is generally more large and complex, we also plan to implement NLP-based, dedicated algorithms for effectively extracting information in such fields, enabling effective interaction of users.

4.7 Summary

We introduced IssueML, an interactive visualization system for managing multiple issues during the development of large scaled software. Based on interviews with domain users, IssueML consists of multiple views that enable effective monitoring of multiple issues, starting from the overview of multiple issues to the details of selected issues following the Visual Information Seeking Mantra. The patterns enable the observation of progresses in multiple issues that were limited, from which users can efficiently find issues of interest to further inspect upon. When one issue of interest is selected, users can further examine the details of the issue by referring to similar issues and utilizing scripts for processing the log files in the selected issue. In the future, we plan to deploy IssueML in actual industrial environments, and collect feedback from users of different background. Based on the feedback, we hope to further improve the functionalities of IssueML in monitoring and resolving multiple issues.

Chapter 5

Towards Supporting the Exploration of Temporal Rank Data with Multiple Colormaps

This chapter presents TRaVis, a novel visualization approach for visualizing and interacting with each of the multiple ranking changes of items in a temporal rank data using colormaps, supporting the data exploration task of users. The research mainly addresses the third research question (RQ3) of the dissertation, of interacting with the information of multiple items in a scalable manner using colors.

5.1 Background

Ranks provide a simple, intuitive criterion for comparing items. With ranks, the task of comparing multiple items can be reduced to comparing each of the relative positions, instead of raw values that may be large in size or complicated to comprehend. Thanks to its simplicity, ranks are popularly utilized in peoples' everyday, non-analytic interactions with data, such as referring

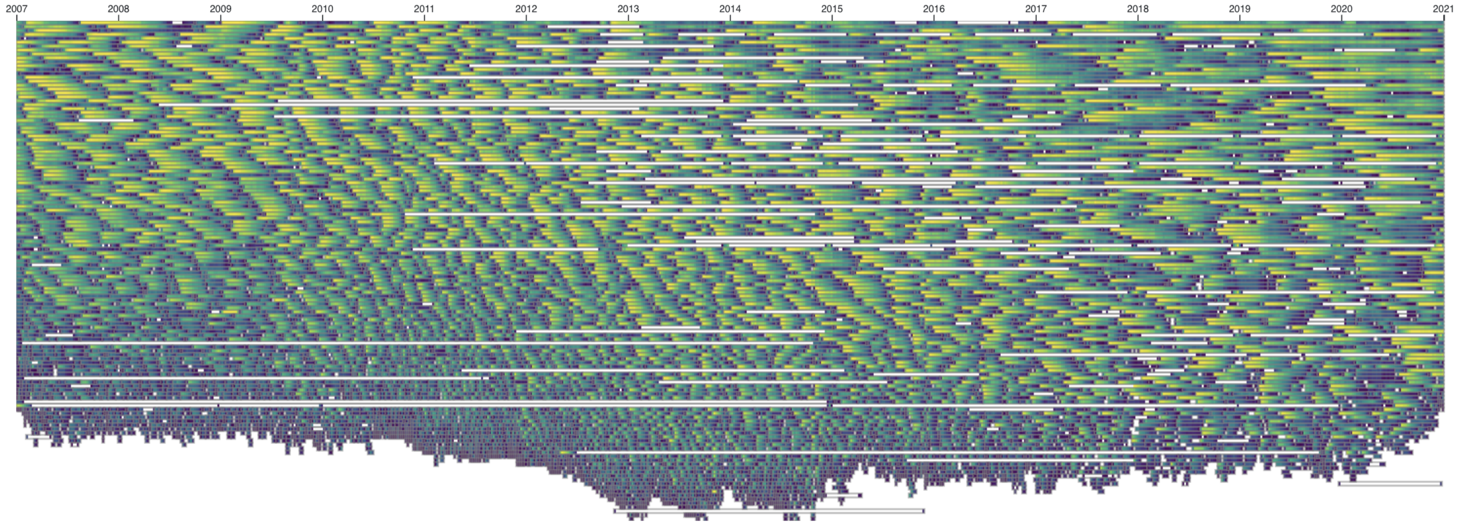


Figure 5.1: The music rank chart [**melon**] visualized as TRaVis. In TRaVis, ranking changes of each items are displayed as rows of colormaps, enabling the visualization of multiple items in limited space without cluttering. The stacked color patterns reveal interesting patterns in the data, such as the overall downward trend in rankings, or the increased length in the colormaps over time which indicate the recent trend of items lasting longer in the charts.

to ranks of popular musics, movies, or sport teams. Furthermore, users often expand this comparison task to multiple values for more information, such as observing the ranking changes over time. However, despite the simplicity and intuitiveness of ranks, interacting with the temporal aspects of ranks is a challenging task. Compared to a single rank chart with fixed items, items in a temporal rank data may appear, disappear or even reappear over time. Thus, not only the number of items to interact is increased, but in each of the items, missing values also need to be dealt with.

Due to the increased complexity, visualizations that are frequently utilized when interacting with rank data also becomes ineffective. For example, tables are inherently inefficient at visualizing ranking changes over time [63], and line charts cannot express the ranking changes of multiple items in a limited space due to cluttering. While previous researches related to visualizing temporal rank data were purposed, they did not fully address the existing issues; for example, the visualization technique being limited to visualizing items from a single rank chart [88], or only being able to display ranking changes in adjacent time frames [70]. Especially, most researches did not consider the casual, non-analytic aspect in interacting with rank data, where users may not have a certain task or objective but hope to encounter unexpected, interesting information from the data in various perspectives [14].

Reflecting upon the limitations, we introduce a novel approach, TRaVis, in visualizing multiple items in a temporal rank data. In TRaVis we encode each items' ranking changes over time as a single row of colormap, with the colors correspond to ranks. Then each of the colormaps are stacked in a 2D space while preserving the temporal information of rankings. Such visual encoding in TRaVis prevents overlapping between items, which helps users to effectively explore for interesting patterns. Moreover, characteristics and trends of the overall data that were previously hard to observe can be revealed from the accumulated patterns. With user interaction of changing the sorting method, users can further seek for information by multiple perspectives in TRaVis. Such effectiveness of TRaVis in aiding users' exploration of temporal rank data is demonstrated in multiple use case involving real world data.

5.2 Design Rationale

Ranks can reduce the difficulty of interacting with multiple items by summarizing the related context of raw values into a single value. Conversely, since ranks are the reduced value of most details, from an analytical perspective it is generally more effective to refer to the raw, detailed values rather than the simplified version of ranks. Thus, rank data is often used in a casual and lightweight fashion in seeking for interesting information, without a specific analytical task. Based on such background, we aim at fulfilling the needs of information flaneurs [14] who do not have a clear analytical objective.

When designing a visualization for information flaneurs, it is important to provide multiple facets of data from which users can interact with. However, most previous approaches in visualizing temporal rank data only provide limited facets of the data to overcome complexity. For example, tables mostly express a single chart from a single time frame, and line charts visualize finite numbers of items due to prevent cluttering. Similarly, researches designed for temporal rank data also restricted the visualization to reduce complexity; for instance, only visualizing the adjacent ranking changes [26, 70], or bounding the items that can be visualized to a certain condition [88].

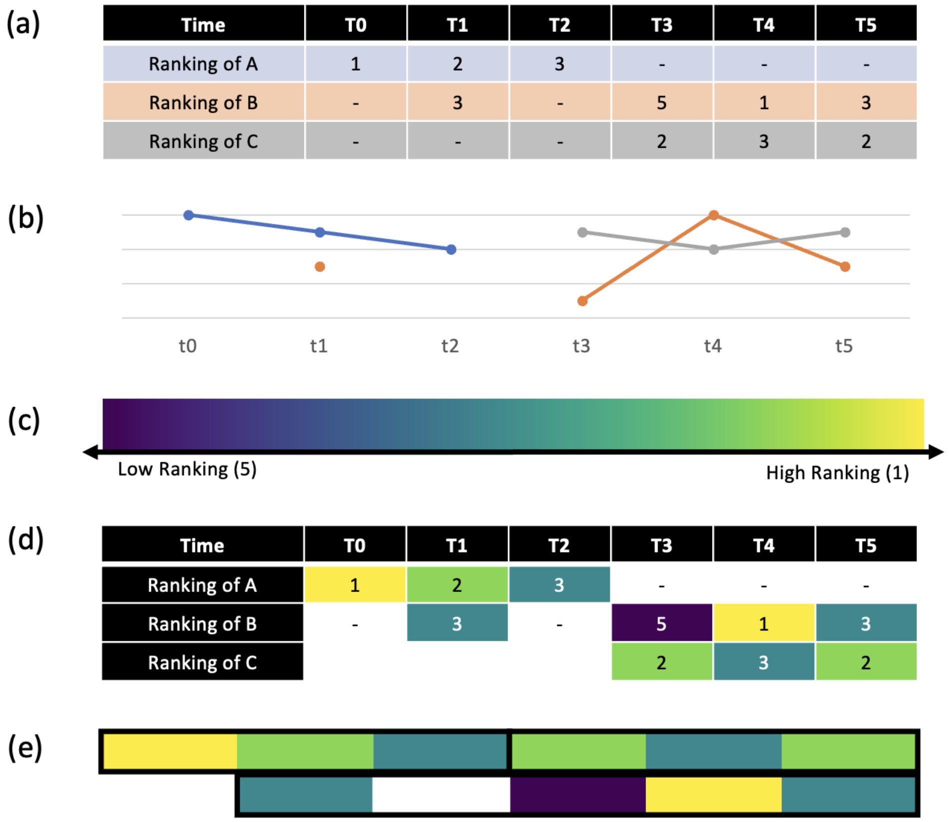
While such reduction is effective when appropriately applied for a particular task, from the perspective of information flaneurs who do not have a clear analytical motivation, the exploration process is hindered by the limited visualization. Thus, in TRaVis we aim for explorability of the data by providing an overview in which users can not only observe the items without restrictions, but also can control the visualization according to their needs.

To maximize explorability in the data and support the user's interaction with each of the items, items are individually expressed instead of grouping in TRaVis as grouping may limit or complicate the interaction. With the visualization technique of stacking multiple colormaps, users can observe each of the items in a non-cluttering manner. Moreover, in TRaVis each of the patterns of colormaps accumulate to generate a stacked pattern, from which characteristics of the overall data can be observed. TRaVis is also equipped with the user interaction of observing the data according to user selected perspectives. Even though flaneurs do not have a clear objective, it would be difficult to interact with many items if no direction at all were provided. Thus, in TRaVis users can steer their observation process by changing how the items are ordered in the heatmap and observe the data in different perspectives. Upon changing the ordering, the stacked patterns also change, from which users can observe the characteristics of the data in a different manner.

5.3 TRaVis

5.3.1 Rendering of TRaVis

The most common visual mapping in visualizing time series is to map each of the values into a 2D position, with X coordinate corresponding to time, and Y coordinate corresponding to ranks as in Figure 5.2(b). However, when



(a)

Time	T0	T1	T2	T3	T4	T5
Ranking of A	1	2	3	-	-	-
Ranking of B	-	3	-	5	1	3
Ranking of C	-	-	-	2	3	2

(d)

Time	T0	T1	T2	T3	T4	T5
Ranking of A	1	2	3	-	-	-
Ranking of B	-	3	-	5	1	3
Ranking of C				2	3	2

Figure 5.2: How TRaVis is rendered from temporal rank data. (a) Ranks of 2 items A and B. Dash (-) indicates no records at that time. (b) Data of (a) visualized as line charts. In line charts, problems such as disjointed records (T1 of B) and overlapping between lines can be potential issues as number of items increases. (c) The viridis [79] color scale utilized to map rankings to colors. (d) Data of (a), in which each of the ranks are colored according to the color scale of (c). (e) In TRaVis, each of the colormaps (as in (d)) of each items are greedily stacked while preserving temporal information.

there are many items to visualize in a limited space, juxtaposition of each items is inevitable. Such causes cluttering between items, which damages the exploration task of the data as each items are become to discern. Moreover, line charts are ineffective at dealing with disconnections caused by missing values (T1 of B in Figure 5.2(b)), as providing identity in disjointed line segments is limited.

To overcome the limitations of line charts, we render each item as a single row, preventing the crossings patterns that causes overlapping between items. This is similar to visualizing each items as rows from a table, in which position of each cells corresponds to time in the rank charts as visualized in Figure 5.2(a). Furthermore, we apply colors correspond to ranking values to each cells instead of texts to indicate the values even when they are small in size. We map rank values to a continuous color scale, as in Figure 5.2(c), and apply the according colors in the cells as in Figure 5.2(d). For cells that do not have a value, we leave them empty with no colors.

Throughout the research, we made use of the Viridis color scale [79] to represent ranks in our visualizations. By employing the Viridis color scale, in TRaVis we mapped rank values to different hues among various factors in colors. While there were alternative color mapping options available, such as varying the brightness of a single color, we determined that visualizing hue as the mapped value offered improved differentiation between undefined values and defined values. Given the significance of handling undefined ranks in our context, we chose to map hues as values to effectively address this requirement.

Finally, with each of the rows of colormaps, we stack them in a 2D space to render TRaVis, as in Figure 5.2(e). Items are first sorted by a criterion (in the case of Figure 5.2, in alphabetical order). Then each colormaps of items are greedily stacked from top to bottom according to the sorted order without overlapping, while preserving the temporal information with horizontal positioning. When stacking the items, items are positioned at the possibly highest position,as in item C of Figure 5.2(e). Even though the order of C is behind B, C is rendered on top of B because there is space left to render C above B, without overlapping with A. Furthermore, each colormaps are bordered to distinguish between different items.

In TRaVis, patterns of stacked items resemble a heatmap, from which characteristics of the data in multiple items can be discovered. As the heatmap is consisted of patterns of each items, similar trends in the heatmap reflect characteristics in multiple items, while outstanding patterns correspond to outliers that do not follow the trend. Moreover, users can discover and interact with ranking changes in interesting items by referring to its nearby cells in the same row in a non cluttering manner. Since each colormap preserves the information of missing values as empty spaces, users can also observe the patterns related to disjointed rankings, which was difficult to be accomplished in line charts. Thus, TRaVis can support the data exploration task of flaneurs by providing the overview of temporal information from which the pattern of each items can be directly observed, without limiting or complicating the interaction.

5.3.2 Controlling Visual Components

We designed a variety of interactions that modifies various visual components in TRaVis to further support the visual exploration task. Figure 5.3 shows an overview of the provided interactions. Firstly, users can choose to remove borders to observe the patterns more clearly, at the cost of making the distinction between individual items harder (Figure **??**(a)). To compensate this, users can also decide to reduce the width of color tile of the starting point and ending point from the other ranks to distinguish and distance different items, as in Figure 5.3(b). The blank spaces can also be dealt with, by filling in other items in the blank spaces as in Figure 5.3(c). While blank

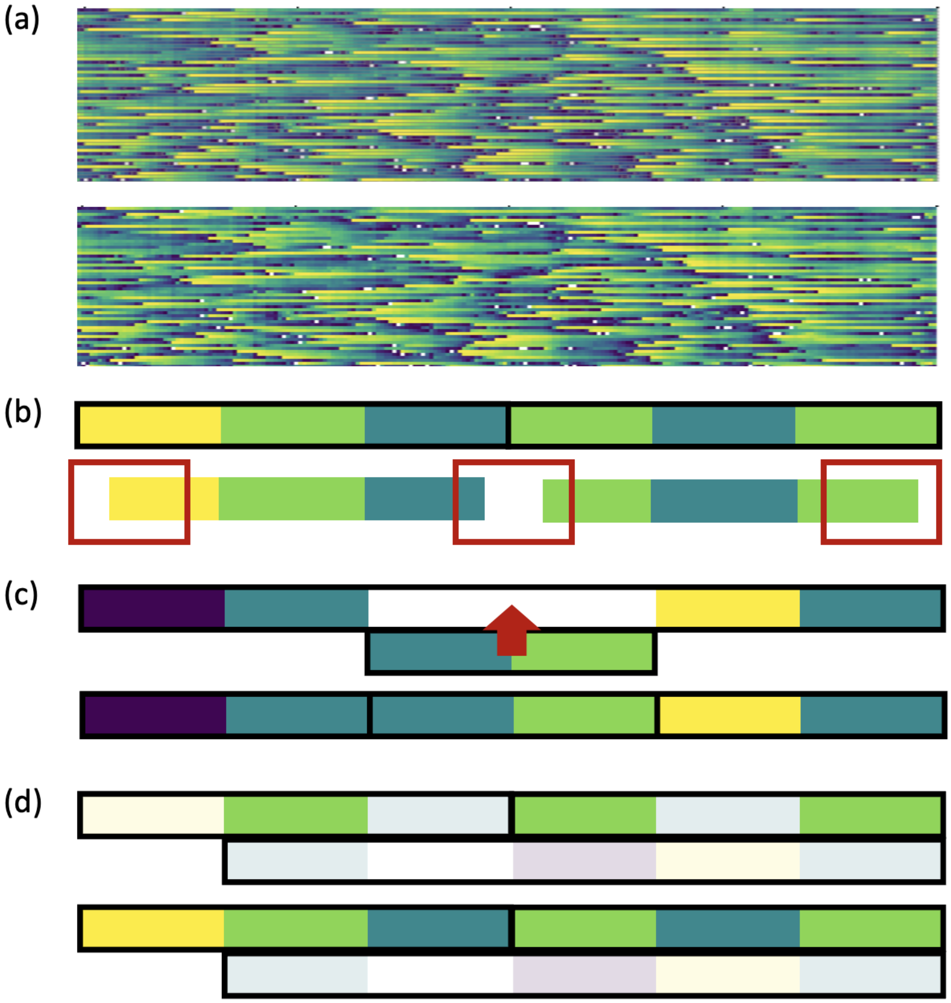


Figure 5.3: Various interactions in TRaVis for supporting the observation of multiple items. (a) The border of each items can be toggled (top: with borders, bottom: without borders). Ranking changes can be more clearly observed when borders are removed, but each of the items become less distinguishable. (b) Size of the starting point and ending point can be reduced, enabling distinction of items without borders. (c) Items can be placed in between empty spaces of other items for more efficient space usage. (d) Users can partially (top), or fully (bottom) filter items depending on the task. In the example, items are partially/fully selected according to the light green value.

spaces provide information related to reentered items, such spaces stand out compared to the colored patterns and may act as distractions in observing the overall trend. Thus users can position items in between the ranks of other items, reducing the size of the blank patterns. While this approach makes it difficult to precisely observe each of the individual items, it helps to observe the overall trends as the blanks are reduced. Figure 5.4 shows the interaction of reducing blanks applied in the original visualization of Figure 5.1. It can be discovered that compared to the original visualization, the visualization with interactions applied can better represent the pattern changes over time while being more space-efficient.

To further aid the exploration, users can alter their perspective in observing the data with interactions. Since the pattern of the heatmap is dependant in how the items are sorted, changing the sorting criterion of items is a crucial interaction, from which users can steer their observation process. Changing the sorting method changes the stacked pattern, and users can observe the data from a different perspective. While the sorting method in TRaVis is not completely exact due to the greedy nature of positioning, items that are highly prioritized according to the sorting criterion tend to be rendered on top of the heatmap. Based on the information, users can roughly expect how items are positioned and start the observation based on the information. But since such pseudo-sorting is not completely accurate, users can also discover unexpected items during the observation task in a serendipitous manner. Thus, changing the positioning of items plays a critical role in supporting explorability for information flaneurs.

Finally, based on the discovered information in multiple colormaps by different sorting methods, users can filter items by various attributes and increase the efficiency of the information exploration process. When filters are applied, records that are filtered out are dimmed and sorted last, allowing users to focus on the selected items. In addition, since filtering also changes

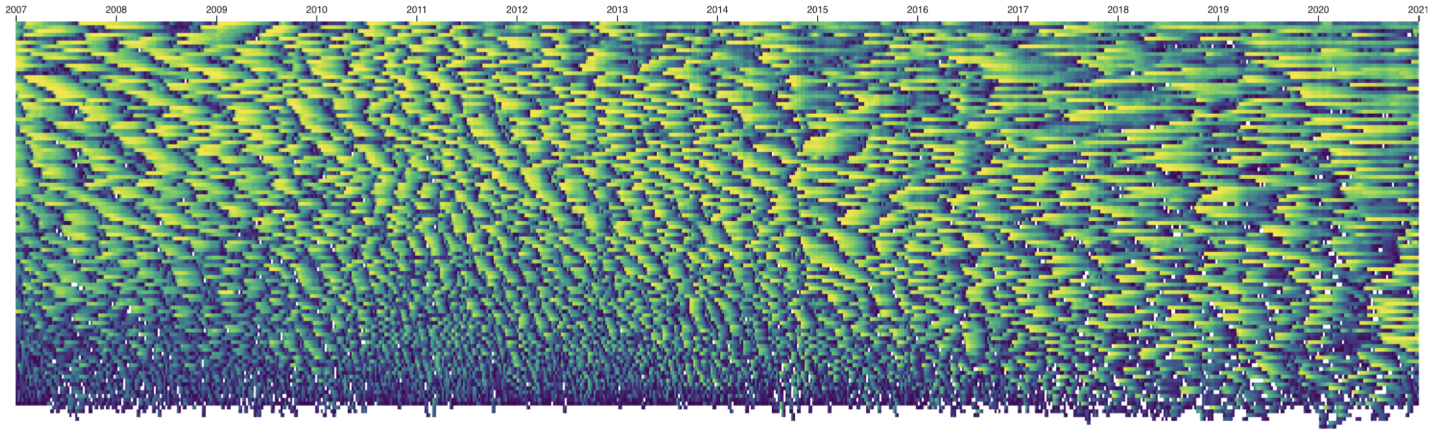


Figure 5.4: The music rank chart [**melon**] visualized as TRaVis, with border removed and blanks filled in with other items. Even though the information of each of the items is harder to be observed, the overall pattern of multiple items can be better displayed.

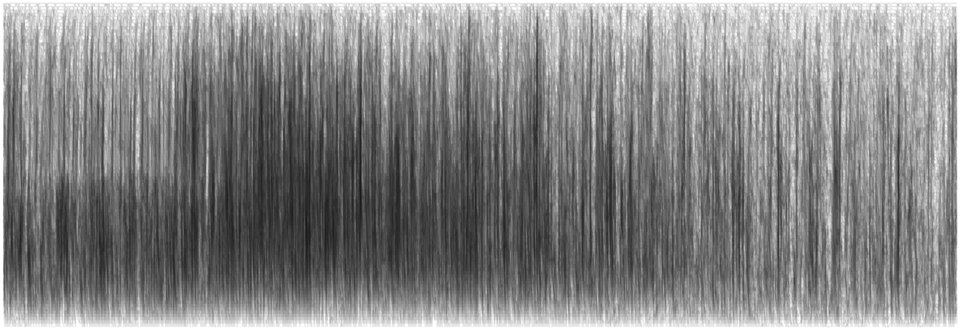


Figure 5.5: The music rank data visualized as line chart. Outside of the information of density generated from the cluttering of lines, it is hard for users to further interact with the data.

how items are sorted, such interaction can also can aid users in discovering interesting items. In addition to applying filtering item wise, users can partially filter items according to certain conditions. Similar to filtering the overall item, when partial filters are applied each of the instances that are filtered out are dimmed out.

5.4 Use Case

The visual encoding of TRaVis enables users to observe the overview of multiple items from a temporal rank data in various perspectives. Based on the overview, users can explore around the items and find interesting information from which users can further expand their examination, fulfilling the objective of information flaneurs. We demonstrate two use cases with real world dataset in which the visual technique of TRaVis can benefit the information seeking process in a temporal rank data.

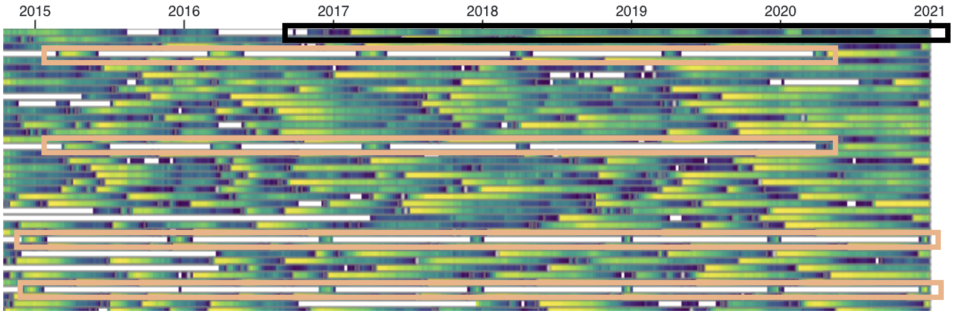


Figure 5.6: The upper right portion of TRaVis from Figure 5.1. Users can readily discover irregular patterns (black box) or items with repeated patterns (orange box) from the color patterns, without additional interactions of reducing the number of displayed items.

5.4.1 Music Rank Data

The music chart data consists of 14 years (2007-2020, 731 weeks) of weekly top 100 popular songs, acquired from a Korean streaming platform [**melon**]. Reflecting the trendy nature of popular songs, new songs frequently appear in the data, which eventually replaces old songs. As a result, the number of items to display is followingly increased, and difficulty in displaying multiple items arises. We demonstrate how TRaVis can visualize information despite the many items in the data.

Figure 5.1 shows the music data visualized as TRaVis, with items sorted by the number of times the items appeared in the rank chart. Even though a total of 9519 items are displayed, TRaVis is capable of visualizing all of the items in a limited space without cluttering. Compared to the line chart visualizing the same data in Figure 5.5 in which multiple lines clutter severely, the information of each of the items can be better observed in TRaVis. Information related to the patterns in the ranks can be discovered from the stacked colormaps. Overall, it can be discovered that most ranking changes are downward (from yellow of high rankings to blue of low rankings), due to new songs entering the rank data. Moreover, changes in listening patterns over time can be reflected from changes in the color patterns. Length of the colormaps tend to be longer as time progresses, meaning that recent songs tend to remain longer in the charts compared to previous songs. Such is attributed to the result of increased competition in streaming platforms, in which fans competitively stream musics for higher rankings [38].

From the visualization of TRaVis, users can search for and discover interesting items based on how they are sorted. In this particular case, where the items are sorted by their longevity, it is natural for users to pay closer attention to the items that have remained for a longer period. By examining the details of each item, particularly those with a longer lifespan, users can uncover interesting patterns and insights from the stacked colormaps. Figure 5.6 shows the upper right part of the overall visualization from Figure 5.1. From the patterns, users can discover irregular patterns (black box in Figure 5.6) where the downward trend is abruptly reversed multiple times, or find items that periodically reappear in spring or winter seasons (orange boxes in Figure 5.6) every year. As each of the ranking changes of items are displayed in a non-cluttering manner, interesting patterns in rankings can be readily discovered without multiple steps of interactions such as filtering.

Based on the discovered patterns, users can expand the exploration in different perspectives by reordering. In Figure 5.7(a), items are sorted by the number of times songs reentered the rank charts, focusing on the blank patterns of reentries in Figure 5.6. Items with reentries tend to appear more frequently in recent times, reflecting the trend of songs climbing up the rank chart due to refocused interest [16]. To further observe the patterns related to reentries, users can change the ordering in alternative criteria. From Figure 5.7(b) in which items are ordered by the sex of the artist, it can be revealed

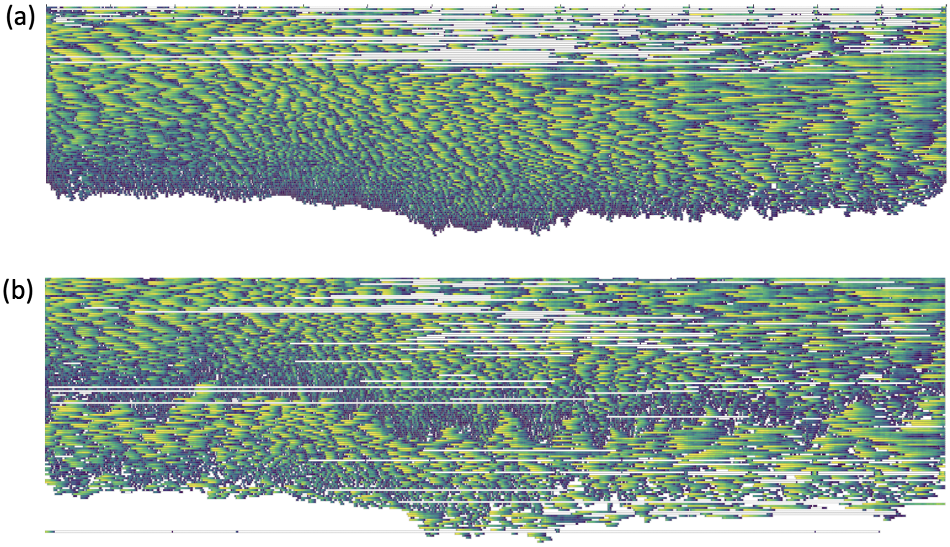


Figure 5.7: Different sorting and filtering applied in the music rank data [16]. Interactions in TRaVis can help discovery information in multiple perspectives. (a) Items ordered by the number of times songs reappear in the ranks. (b) Items ordered by the sex of the artist (male, female, others from the top).

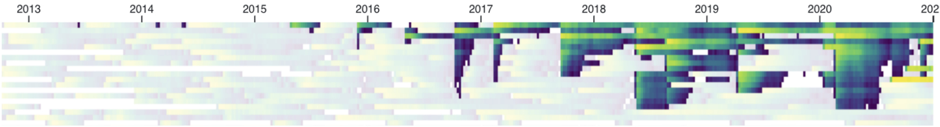


Figure 5.8: Items filtered by the same artist of the black boxed song from Figure 5.7. It can be observed that the rise in rankings correspond to periodic appearances of new songs.

that songs in the upper region corresponding to male artists tend to reenter more that the lower region corresponding to female artists.

Furthermore, users can filter items to focus on the subset of data that interests them. Figure 5.8 shows the subset of data visualized by the artist of the black boxed item in Figure 5.6, to further investigate why the patterns have occurred. It can be observed that the spikes correspond to the simultaneous appearance of items, implying that the rise in the ranks is highly related to

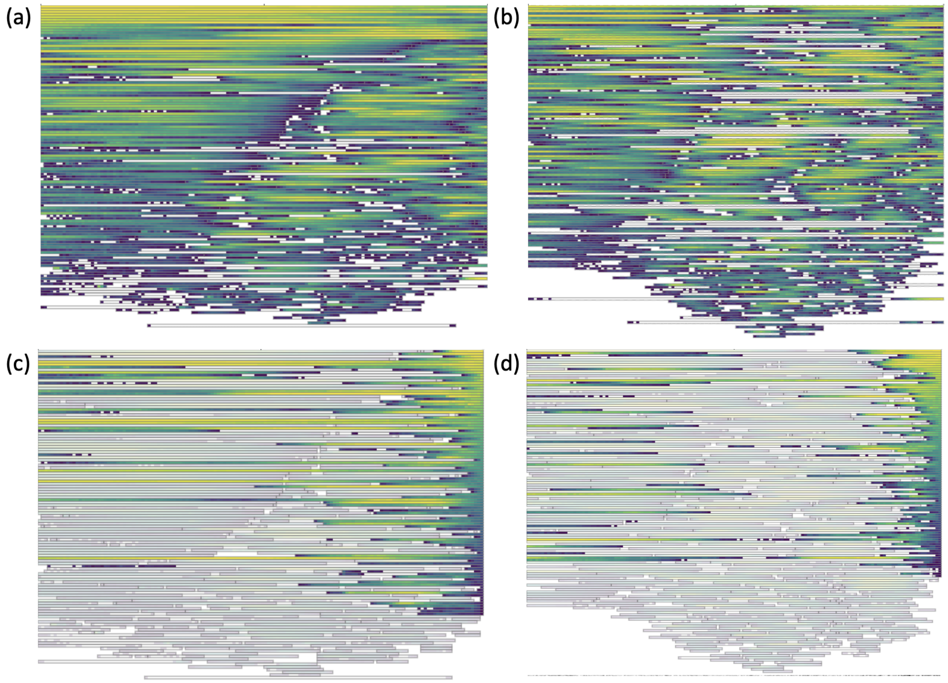


Figure 5.9: The Baby Name Data [4] rendered as TRaVis. Color distribution reveals different characteristics of the two dataset. (a-b) Names sorted by frequency of appearance ((a): male, (b): female). (c-d) Names sorted and filtered by latest year's rankings ((c): male, (d): female). Colors of items that are filtered out are lightened.

the newly appeared items. As in the examples, in TRaVis users can readily discover information from the overview, and extend observation based on the information, supporting the exploration task of information flaneurs.

5.4.2 Ranks of Popular Baby Names

The baby name data [4] is consisted of yearly rankings (1880-2022, top 100) of popular babies names (two datasets of male and female) in the United States. Previously, it was hard to visualize and observe the overall data unless users had clear targets (e.g. names, rank of a single year, etc.) to focus on, due to the data containing only the name and its rankings. We demonstrate how

TRaVis can aid the observation of the overall data. Figure 5.9(a) and Figure 5.9(b) shows each of the datasets visualized as TRaVis, in which items are sorted by the number of times they appear in the charts. From the color patterns, difference in the patterns of ranking changes can be found, in which the names in the female data tend to be more diverse and variate more. Especially, while some names in the male data that have never left the ranks can be observed, such pattern does not appear in the female's data.

By combining filtering and sorting, users can observe items with TRaVis similar to previous interactions with a single rank chart, with added advantages. In Figure 5.9(c) and Figure 5.9(d), the same data is sorted and filtered according to the rankings of the most recent (2022) year. The advantage of encoding each items in a single row enables users to not only observe the current ranks, but also how the items have progressed over time. The patterns show that some of the names in the current popular names are previously popular names that regained popularity. However, it can be also discovered that such patterns in reoccurrence differ by sex. For example, in the female data, the middle-right region (corresponding to the 70s-80s) is noticeably less colored (in other words, filtered out) from the overall heatmap. However, such phenomenon is less prevalent in the male data. As shown in the examples, TRaVis enables the visual comparison of characteristics in rank data that was previously difficult to accomplish.

5.4.3 National Soccer Team Rankings

The soccer rank data [19] contains monthly rankings (top 100 of 1992.12.31 - 2022.02.10, 275 months) of national soccer teams. Compared to the previous examples where the number of unique items is large due to items that newly appear, new items corresponding to new nations rarely appear in the soccer

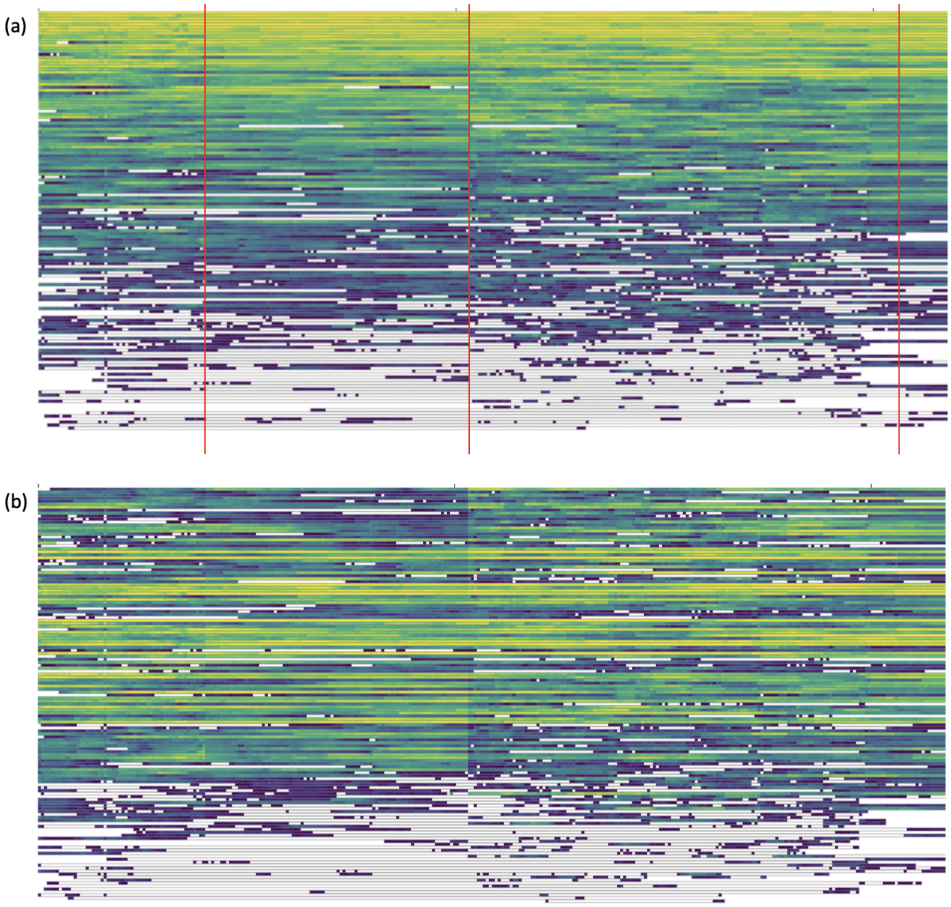


Figure 5.10: The soccer ranking data [19] rendered as TRaVis. Changing the order of items in TRaVis is a critical interaction which allows the observation of items in novel perspectives. (a) Items are ordered by the average ranks. Lines correspond to the moments when changes in the ranking criterion were applied. (b) Items are ordered by the gained or lost ranks due to the changes in the ranking criterion in 2006 (middle line). Users can effectively observe how the changes have influenced the ranks whilst conserving the information of ranking changes over time.

data. Thus, scalability is not a critical problem in displaying the soccer rank data, as the data is consisted of 166 unique items. Nonetheless, with TRaVis users can effectively observe and interact with the information of multiple items. We demonstrate how the sorting interaction can aid the observation of items, focusing on the changes in ranking criteria.

The ranking criterion in the rank data had three major changes in 1999, 2006 and 2018 respectively. Figure 5.10(a) shows the soccer rank data visualized as TRaVis, in which items are sorted by the average of the accumulated ranks. The lines indicate when the criterion changes occurred. With the color changes, users can observe how the changes affected the ranks of nations. From the Figure, users can discover that the criterion change in 2006 (corresponding to the middle line) relatively had the biggest influence in the ranks of the nations. Moreover, they can discover that the ranking changes are stale after the changes in 2019, due to the changes in the scoring method and the influence of COVID-19.

To further observe the influence of the changes in the criterion, users can order items according to the as in Figure 5.10(b). Previously, to interact with the same information, users had to manually find items in the visualization of multiple items, or observe the information from a separated, independent visualization. In TRaVis users can integrate this process into a single visualization by sorting the items according to the gained and lost ranks. With the visual information, users can effectively find items according to the sorting criterion, with the context of ranking changes in multiple items provided. Reflected to the example, users can freely define the required sorting method according to their needs, and effectively observe how the patterns of multiple items are related to the sorting criterion.

5.5 Discussion

In TRaVis, each of the ranking changes of items are visualized in a non-cluttering manner in limited space, thanks to its unique method of positioning items. Due to the positioning, users can search and discover interesting patterns in the data from the information of each of the items. The usage of colors as a visual channel for encoding rankings serves a critical role in enabling such scalable visualization. While colors are considered as a less accurate channel for visualizing numerical values [9], with colors the ranking values of each items can be rendered in minimal space without direct overlapping between the values. The low resolution of colors is also advantageous in visualizing the overview of multiple items, as subtle changes in ranks that may act as disturbances in observing the overview are less noticed due to its low accuracy. From the patterns of colors generated by stacks of colormaps, the overview of the temporal rank data that was previously difficult to express can be effectively visualized, from which users can explore for interesting information in various perspectives with the help of interactions.

Sorting the items is a pivotal interaction in TRaVis, which enables the interaction of multiple items by a preferred perspective. Compared to previous approaches in visualizing temporal rank data where the position of each items are fixed to encode ranking values, in our approach the vertical position of items can be controlled by users. Such opens up numerous new possible interactions with the rank data that were limited in other visual approaches. Most notably, using TRaVis users can sort the items according to a certain criterion and observe how the temporal trend is altered according to the criterion. Previously, such interaction was mostly only available in multiple juxtaposed visual components, as there was no effective way to express both the temporal trends in relation to the criterion of interest. In TRaVis such limitation can be overcome by sorting the items according to the criterion, in which the relationship between the temporal trend and the sorting criterion can be hinted from how the temporal trends are positioned in the visualization.

We also believe a further inspection on the positioning of items would benefit the users' interaction with TRaVis. Currently, the sorting in TRaVis is most limited to item-wise sorting according to a single attribute. We hope to extend the sorting to more various criteria, further supporting the observation of data in multiple perspectives. For example, users may manually position items of interest according to their preference, similar to stacking blocks. Based on the user input, the sorting order of other items may adaptive change, for instance, in which items that are similar to the interacted items are positioned nearby. Likewise, the items can be sorted according to similarity in the ranking changes, aiding users to find information related to the patterns in changes of rankings over time. In addition to the provided examples, we would like to extend and discover novel methodologies in positioning items for supporting various user needs.

Throughout the paper, we mostly utilized the viridis [79] color scale for visualizing the ranking values to control the potential effects that may occur when the color scale is changed. Nonetheless, we believe that the color scale is a major factor in the visualization that can be modified and controlled. In addition to altering the continuous color scale to another color scale (such as the inferno or plasma color scale [79]), the color scale can be altered into a non-continuous color scale to emphasize certain values in the ranks. The opacity of colors may also be mapped to the data, highlighting certain items or regions similar to the filtering interaction provided. Colors may even extend upon visualizing ranking values, and be utilized at expressing alternative information such as the value of a certain attribute at the certain time. Combining such approaches, users can observe how the characteristics of the multiple items in temporal rank data change over time, as in the example provided in Figure 5.11. While the information related to changes in the ranking is mostly lost, from the color distribution resembling an area chart users can observe how each of the trends related to genres have progressed over time. However, the approach in TRaVis has additional advantages of visualizing the information related to each of the items, which is hard to be addressed in area charts.

Nonetheless, there are also limitations in the approach of visualizing multiple items as TRaVis. Due to the overwhelming usage of colors to encode rankings, the visualization might feel overwhelming to users depending on the user task or the size of the data. In addition, to our experience, careful usage of visual channels were required as most of the other additional approaches that utilized colors were ineffective. Due to such limitation in utilizing visual channels TRaVis, currently only a single attribute (in our paper, mostly rankings) can be visualized. Thus, while the sorting interaction can aid the observation of data according to the sorting criteria, users have to manually check on the value of the criteria for further inspection. Moreover, while colors are effective at displaying the overview, the limitation of color as visual channel in terms of accuracy still remains in interacting with each of the items. Designing additional interactions or visualization techniques for overcoming the aforementioned limitations is one of the main goals we hope to accomplish. Finally, we aim to further reveal how people experience and benefit from the visual encoding of TRaVis in interacting with real world data, and discover more interesting use cases and usages.

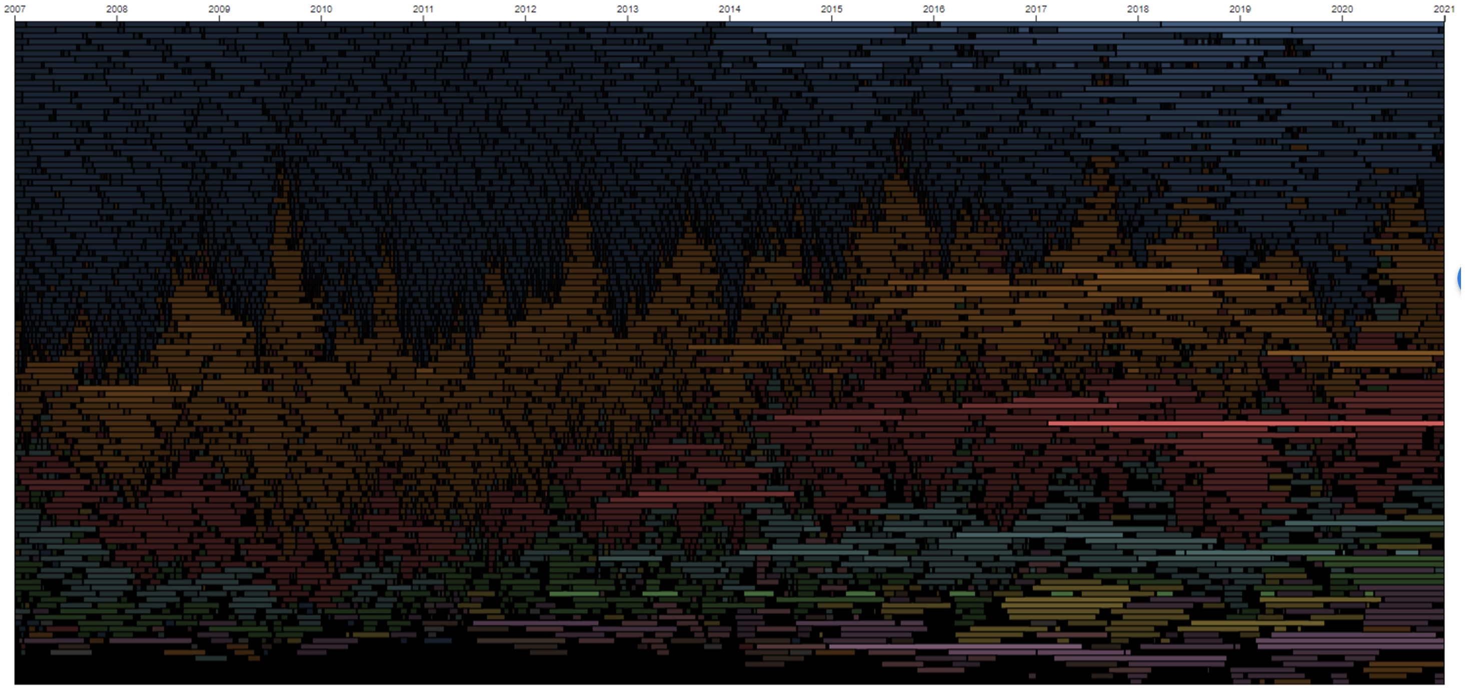


Figure 5.11: By combining and altering the color scheme, the temporal rank data can be visualized in different perspectives in additional to the ranking values. Items in the music rank data [53] are ordered and colored by the genre of the songs, and the opacity is proportional to the number of times appeared in the charts. Users can observe the changes in genre similar to heatmap or area chart, and also can discover salient items that stayed on the rank charts for a long time.
The genre-color pair in the visualization are as follows : Ballad - Blue, Dance - Orange, Red - Hiphop, Teal - R&B, Pop - Yellow, Rock - Purple

5.6 Summary

We introduced TRaVis, a novel approach in visualizing temporal rank data. In TRaVis, we display each ranking change of items as a single row of colormap and stack each the rows without overlapping, enabling the observation of multiple items and its overview in a scalable manner. Users can interact with the data by reordering the colormaps, and steer the observation task by inspecting upon the data from different perspectives. Reflecting the non-analytical context in rank data, TRaVis is designed to fulfill the needs of the information flaneurs, and supports users to explore around the data and enjoy serendipitous discoveries. We demonstrated multiple use cases with real world data in which the visual encoding of TRaVis is effectively utilized in exploring temporal rank data. For future work, we hope to expand the usage of TRaVis to other tasks and scenarios related to interacting with temporal rank data, by designing various visual modifications and interactions based on user feedback.

Chapter 6

Discussion

In this chapter, we discuss about the lessons we have learned from the researches, and the current limitations in the dissertation.

6.1 Lessons Learned

6.1.1 Significance of Utilizing Colors in Expressing Multiple Values

Throughout the researches, we commonly utilized colors to express information of multivariate data in a scalable manner. We discuss critical insights acquired from the researches.

Overcoming the Critical Cluttering Issue with Colors

Colors play a critical role in the introduced researches as they serve as the primary visual channel for representing multiple values across multiple items in visualizations. This utilization of colors offers a notable advantage by effectively expressing each value within a confined space. Unlike other visual channels that require more space to convey values (for example, mapping values to the length or size of a visual component), colors can be represented using minimal pixels, enabling the visualization of multiple values in a limited area. This approach proves particularly valuable in addressing the challenge of scalability, where the display of multiple values is necessary despite space constraints.

Moreover, incorporating colors in the expression of values offers a significant advantage in terms of flexibility in positioning compared to positional encodings. Unlike positional encodings, which are commonly used to convey information but are constrained by specific positional scales and prone to overlapping issues between items based on the characteristics of the data, colors provide a relative freedom in placement. This increased flexibility allows for greater adaptability and modification to meet the specific requirements of the given task in the data. Thus, although the use of colors may slightly compromise the accuracy of each individual value, it effectively prevents the more significant problem of cluttering. By strategically positioning and mapping colors to the data in visualizations, researchers can effectively overcome the limitations of positional encodings and create visual representations of the data based on the task.

In the dissertation, we make effective use of the advantages offered by the relatively freed positional channel to convey information when visualizing multiple values without cluttering. This was accomplished by strategically mapping colors to values, and positioning them to effectively represent the values being visualized. For instance, in Parallel Histogram Plots, colors are mapped according to a single attribute and applied on top of histograms, enabling scalable expression of information from which the relationship between attributes can be inferred from the patterns. The information conveyed by colors can effectively represent relationships regardless of their positioning even when the attributes are distant from each other, in contrast to the positional encoding of PCP. Similarly, in TRaVis, the use of colors facilitates the expression of numerous ranking changes within a limited space without introducing clutter. By utilizing colors that effectively convey information in a limited space, the issue of cluttering can be effectively addressed, enabling the representation of a vast amount of information in multiple values. We believe that this relatively position-freed property of colors can play a crucial role in tackling the prevalent scalability challenges in visualization, with the combination of appropriate color scales and its positioning.

Handling Multiple Values through Color Patterns

The advantage of utilizing colors to represent values without cluttering extends beyond the recognition of individual values. By employing colors to visualize multiple values, users can identify patterns that emerge from the combination of the colors. Patterns from multiple values enable users to observe and comprehend multiple values simultaneously, enhancing the efficiency in interacting with the data. Such advantage in utilizing colors is reflected in the proposed researches. In IssueML, the use of colors and marks to indicate important events and changes in a single issue allows users to observe not only the progression of individual issues but also compare them based on visual patterns. This aids a more scalable and effective observation of multiple issues, from which users can identify trends and outliers across multiple issues. Similarly, in TRaVis, the ranking changes in multiple items can be recognized as color patterns, and users can observe how such patterns have progressed over time. Reflected in the examples, the utilization of colors in visualizations enables users to gain valuable insights in patterns generated from multiple values, that would have been challenging to discover without visual inspection.

Such patterns derived from colors can serve as an critical step in the observation of data, allowing users to extract meaningful information from multiple values. This visual analysis with color patterns is akin to clustering multiple attributes into a single, multidimensional representation in a visual manner. Patterns of colors can provide an intuitive and accessible way to explore multiple values, allowing users to effectively uncover insights and trends in higher dimensions, even in situations where algorithmic clustering is limited. Therefore, colors provide a significant advantage as a visual channel for scalable observation and analysis in multiple values, allowing users to extend their interaction with the data beyond individual values and encompass multiple values effectively. This important advantage of color patterns can further aid users in overcoming scalability issues in interacting with multiple values in data, by enabling the expansion of the perspective of interaction to multiple values.

Further Interacting with Data by Changing Color Patterns

While color patterns facilitate scalable interaction with multiple values, from the researches we also learned the importance of incorporating functions that support observation during the visualization process. Although colors have the advantage of displaying numerous values within limited space, this advantage can be compromised without appropriate interaction techniques that assist users in discovering information from the patterns. Therefore, it is crucial to not only visualize multiple values effectively but also provide specialized interactive features that specifically address the manipulation and exploration of multiple color patterns.

Taking account the aforementioned precaution, our research aimed at providing interactions that assist users in effectively navigating multiple values of items. In Parallel Coordinates Plot, users can modify the pivot attribute, which changes how histograms are colored and enables the observation of relationships based on the selected attribute. In IssueML, users can apply filters based on various attributes, allowing them to control which patterns corresponding to issues they want to observe. Finally, in TRaVis, users can manipulate the positioning of items, facilitating the discovery of different patterns with the help of other interactions like filtering. In the researches conducted, we explored the potential of interacting with multiple values through the adjustment of the color scale or its positioning in the visualization. By making these adjustments, it is able to uncover information from different perspectives, as the altered color patterns can provide new insights in the patterns and relationships. As in the examples, in expressing multiple values as colors, it is important to enable users to customize and manipulate the color representation according to their specific needs and preferences, facilitating a deeper understanding of the multiple values being visualized.

6.1.2 Bridging the Gaps in Information Visualization

Our research primarily concentrated on visualizing and facilitating interaction with multiple values in a scalable manner within limited space using colors. Although this approach may appear counterintuitive, as it emphasizes individual values rather than grouping them, its purpose was to address and highlight two crucial gaps in information visualization.

Between the Overview and Detail

The increase in the volume of data requiring interactive analysis and exploration has posed a significant challenge in terms of scalability in data visualization. However, while efforts have been made to address this challenge at the high-level overview of items, there has been relatively less emphasis on effectively representing and navigating the granular details that have also grown in tandem with the overall data size. Despite its critical role in data interaction, previous approaches have been limited in dealing with displaying the details of multiple values, and such visual practice that break down when many items are displayed was utilized in inertia.

While reducing the amount of target data to visualize is indeed a common approach to address scalability challenges, we acknowledged that this solution may lead to a never-ending cycle. When interacting with data, it is inevitable for users to interact with data involving large amounts of details by chance, depending on the characteristics of the data. This raises the question of whether the process of reducing and retrieving data should be continuously repeated. As these challenges cannot be fully controlled or avoided, our aim in the researches was to fundamentally address these limitations by visually augmenting them. In order to visually resolve the issue, we leveraged colors as a crucial means of bridging the gap between detailed information and the overview. By utilizing colors that enable the visualization of multiple values, a scalable intermediate step that maximizes the versatility offered by colors can be provided. This approach allows for the advantages of representing each value in a non-cluttering manner and expanding the observation of multiple values through patterns.

Ultimately, our approach of presenting information within a limited space was aimed at facilitating the integration of techniques within the context of visual analytics, bridging the gap between data overviews and detailed analysis. While observing the details is one crucial step in visual analysis, it is also important to consider how it works in harmony with the overall visual representation. Colors can play a valuable role in efficiently visualizing data details within limited space, optimizing the utilization of available visual real estate, and enabling seamless integration with other visual components in the field of visual analytics. This philosophy is evident in our proposed researches. In Parallel Histogram Plots, the original PCP can be enhanced without requiring additional space, and in IssueML, the space-efficient color patterns enable the visual analysis of multiple issues following the Visual Information Seeking Mantra. We firmly believe that utilizing color-based visual representations in expressing data can promote a holistic and comprehensive understanding of the data in a scalable manner, even in the details.

Between Novices and Data

Despite the increased opportunity in users' access to more extensive and detailed data, interaction with the data was limited due to the limitations in visualization approaches. On one hand, analytical methods necessitate background knowledge related to visualizing and processing data, which users may lack. On the other hand, simple visualization techniques struggle to effectively display the data, particularly given its increased complexity and size. As a result, a dilemma arised when users are confronted with the task or desire to explore information within large data. While there is a need to simplify the information for users, limitations in understanding and interacting with reduction techniques hindered the usage of such approaches.

Considering this situation, our research focused primarily on bridging the gaps between novice users and large data by utilizing colors as representations of values. Leveraging the advantage of expressing multiple values through colors, our studies allowed users to easily recognize and interpret intuitive patterns, enabling simultaneous interaction with multiple values. Through direct reference to the values embedded within color patterns, users can understand and analyze the data without relying on reduction techniques or complex calculations. Additionally, we showed that interactively controlling the color patterns can greatly assist users in effectively exploring and observing the data, facilitating a more comprehensive understanding of the data. Therefore, by utilizing colors, users can intuitively understand and interact with multiple values in multiple items, without undergoing complicated steps or dealing with complex calculations.

Furthermore, our aim was to facilitate users in their curious observation of the data by individually representing multiple items in a limited space, thereby promoting explorability [14] within the visualizations. To enhance explorability, it is crucial to offer various perspectives and interactive options, particularly considering that novice users often lack analytical motivation and prefer encountering data in a casual manner. The use of colors enables the display of multiple items in a limited space, allowing users to easily discover interesting information and initiate their data observation. Colors also serve as a direct representation of values, making it convenient for users to refer to specific data values. This aspect is particularly evident in TRaVis, where users can effortlessly uncover captivating ranking patterns among the multiple items' patterns. Although this approach may be less analytically oriented, it still provides suggestions and offers an interactive and highly flexible experience for users to explore the information interactively. We assert that by effectively utilizing intuitive and simple visual interactions, such as strategically incorporating colors in our dissertation, we can enhance the data literacy of users, ultimately enabling a wider audience to benefit from interactions with diverse and complicated data.

6.2 Limitations

In this section, we discuss the current limitations of the researches in the dissertation. While the proposed visualizations in the dissertation focuses on providing scalability, such does not mean that they are not infinitely scalable. In Parallel Histogram Plots, even though colored histograms provides a scalable overview of multiple items, the scalability issue reappears when the number of selected subset displayed as polylines from the histogram is too large. In IssueML and TRaVis, while visualization with colors can express information of multiple items in limited space, the number of items that can be displayed is still bound by the size of screen. Nonetheless, it is important to note that as the primary objective in the researches is to provide scalability in observing multiple values, additional approaches for interacting with the overview of data is recommended when they are limited in scalability.

The visualizations are also currently limited in the types of data they can express. Parallel Histogram Plots are designed for visualizing numerical attributes, and require alternative approaches in visualizing other types of attributes such as categorical attributes. IssueML is specialized for interacting with issues, and TRaVis is only capable of visualizing temporal rank data, due to its stacking of multiple items difficult to be applied in continuous time series. As each of the approaches are specialized for a certain type of data, modification of the visual encoding or data should be required in expanding the capability of the visualizations.

Finally, in the presentation of our research, we opted to fix the colors utilized in the visualizations to eliminate the potential influence of color changes. In Parallel Histogram Plots, we utilized a bivariate color scale to better express extremum values that users are generally interested in. In IssueML, the number of colors were limited to a countable number to prevent the visualization to be over-complicated, and in TRaVis the viridis [79] was used to distinguish undefined ranks from the defined. Although each color scheme in our visualizations was chosen based on a specific rationale and internal experiments, we acknowledge that we did not thoroughly explore and compare alternative color scheme options. While we believe that changing the colors may not significantly impact or hinder the interaction, we also recognize that properly applying alternative color choices can have a significant positive effect on the visualizations we provide. A variety of factors can be considered when exploring alternative color schemes, including the color palette and various parameters such as intensity, brightness, and saturation. Taking these factors into account can greatly enhance the ability to observe and interpret information in visualizations. For example, we demonstrated that by changing the colors in TRaVis, not only the ranks of multiple items are observable, but the information of other attributes over time can also be visualized. Exploring different ways to scale and manipulate colors, and providing related interactions in the visualization holds as a potential future work in our approaches.

Chapter 7

Conclusion

Concluding the dissertation, this chapter first summarizes the contributions made by the studies and systems presented in the dissertation. Then, future research agendas and opportunities are discussed.

7.1 Summary of Contributions

Throughout the dissertation, our primary goal was to address the thesis statement of, "Utilizing colors as a critical channel to express values can overcome the scalability and complexity issues in visualizing multiple values of multiple items, supporting user's exploration and interactions with large multivariate data.", by answering to the following questions.

RQ1. ***How can multiple items in data be visualized in a scalable manner using colors?*** To address RQ1, we developed Parallel Histogram Plots (PHP), a visualization methodology for dealing with the innate limitations of parallel coordinates plot (PCP) by attaching colored, stacked-bar histograms on each axis of PCP. Colors in the histograms of PHP are applied according to a discrete color schemes corresponding to the ranks a single attribute, from which users can observe the relationship between attributes which was previously limited in PCPs. With the combination of histograms and colors, PHP can display a scalable information of multiple items and its relationship in the selected attribute and all the other attributes without cluttering. Moreover, such relationship can be observed even if the attributes are positioned distant from each other. Through the research, we provided demonstrations in which the technology is effectively utilized in visualizing multiple items, and performed a user study on how PCP performs in helping users estimate the correlation between attributes. The results showed that the performance of PHP was consistent in the estimation of correlations between two attributes regardless of the distance between them.

RQ2. ***How can items with complex, multiple attributes be effectively expressed with colors?*** Dealing with RQ2, we implemented IssueML, an visual analytics tool for monitoring multiple issues that occur during development of a large software. One of the manager's job in the development of large softwares is to ensure that errors in the software are resolved in time. To do so, they keep track on issues that are open, and manage the subordinate developers responsible for the issues. However, due to the complexity of information and its changes over time, managers had difficulties in dealing with multiple issues. Reflecting on the limitations, we developed IssueML for monitoring multiple issues based on interviews with domain experts. In IssueML, users can interact with from the overview of multiple issues from the details of a singular issue, following the Visual Information Seeking Mantra. IssueML is equipped with visualizations utilizing colors that reveal how multiple fields in each of the issue have progressed over time, enables the observation of multiple, complicated fields in multiple issues.

RQ3. ***How can visualizations support the users' interaction with multiple items utilizing colors?*** Reflecting RQ3, we designed TRaVis, a visualization technique of displaying multiple items in temporal rank data. Previously, users had limitations in interacting with multiple rank items over time, due to the complexity in ranking changes. While dedicated approaches for reducing the complexity were introduced, they were objected towards a certain analytical task, which limited the users' interaction with the data. In TRaVis, we display the multiple ranking changes in items as color patches, in which users can observe information in items without restrictions. Such visualization enables users to freely interact around the data, from which they can steer the observation of multiple items in various perspectives by changing how the items are positioned.

7.2 Future Research Agendas

We discuss future research agendas discovered from our researches.

7.2.1 Expanding the Interaction with Multiple Values using Colors

In our dissertation, our primary focus was to address the scalability issue of representing multiple values in multivariate data using colors. Based on the insights gained from the researches, we aim to expand our approach to encompass different types of data in various scenarios. One specific type of data that particularly interests us is network data, such as trees or graphs. In visualizing such networks, the relationships between items are commonly displayed as interconnected lines. However, when there are numerous nodes or links to display, cluttering issues can arise, similar to the challenges in PCP emphasized in our research of Parallel Histogram Plots. We strongly believe that by effectively positioning the nodes and utilizing colors to express information in a non-cluttering manner, while considering the users' objectives when interacting with the data, we can successfully resolve these cluttering issues in network data in a similar manner to the proposed researches.

Furthermore, we hope to leverage the benefits of colors in analytical and practical tasks involving simultaneous handling of multiple values. In our research, we focused on assisting novice users by enabling interaction with visualizations through individual value representations using colors. However, this approach becomes impractical when dealing with a large number of values, such as in machine learning scenarios where users need to observe thousands or millions of attributes simultaneously. We believe that by combining data reduction techniques, such as clustering, with space-efficient visual representations using color patterns can help overcome these limitations. When visualizing large sized data, even after applying data reduction techniques, the results may still be substantial in size. In such cases, the space-efficient color encodings can complement the visualizations effectively. Additionally, the space-efficient advantage of colors allows for seamless integration with other visual components, leading to more comprehensive and insightful visualizations in the context of visual analysis. By harnessing the inherent benefits of color encoding, we anticipate further expanding our approaches to effectively visualize larger-sized data in resolving real-world problems.

7.2.2 Assessing the Effectiveness and Scalability in Color Patterns

In our approaches, we showed that colors can be utilized as a critical channel in resolving the scalability issue in various aspects. However, in the researches, we fixed various factors in which could be further influence the effectiveness of the visualization. Measuring the effect of in changing how data is expressed is one future work. For example, in our research approaches we limited the number of colors utilized in the visualization to a certain extent. While this is based on the notion that too many colors are detrimental in recognizing the patterns, we currently do not have a clear understanding on to which extent colors can be utilized in visualizing multiple patterns. We think that a deeper understanding in such can help in exploiting colors in expressing scalable information.

Moreover, we recognize the importance of designing methods to assess the effectiveness of color encoding in visualizations. Since our research primarily focuses on higher-level tasks, it is essential to acknowledge that the effectiveness of colors can vary for low-level tasks or performance measures. For example, in our research on Parallel Histogram Plots (PHP), we measured the effectiveness of color patterns in a correlation coefficient retrieval task, which is a relatively low level task that may not fully capture the broader effectiveness of the color encoding. To ensure a comprehensive evaluation, a more detailed and thorough assessment is necessary. Our future goal is to provide comprehensive guidelines for assessing the effectiveness of color encoding in different visualization contexts. These guidelines will be based on evaluating our techniques and considering various factors, taking into account both low-level and high-level tasks.

7.2.3 Further Supporting Novices with Visualizations

One critical motivation of the dissertation was to support users in limited situations in which they may not have a clear understanding or an analytic motivation in their interaction with the data, converse to most researches which is based on rigorous definition of the supported tasks. Previous visualizations for scalability mostly required modification of data according to a certain algorithm, which requires to be well defined for the visualization to be effective. While this approach is useful in analysis tasks, such approach is limited at addressing the users' casual interaction with data. We stress that moving towards supporting such ill defined tasks in everyday users is a critical future research topic. Similar to such approaches, we hope to further research towards supporting exploration of data with scalability and high versatility in supporting users of a wider audience.

Meanwhile, although our research has primarily focused on providing visualizations for users to interact with large-sized data, we acknowledge that novices often encounter more fundamental challenges when interacting with data. These challenges typically involve difficulties in understanding visual information and extracting critical insights from visualizations, which is another crucial issue that should not be left out when supporting novice users. One promising example in supporting novices in these tasks is leveraging AI to generate visualizations. Through this approach, users can provide relatively ambiguous prompts, and AI algorithms can generate corresponding visualizations in response. We argue that taking such approaches into consideration for supporting novice users is crucial, as it plays a vital role in bridging the gap between their understanding and the visual representation of data.

7.3 Final Remarks

In the dissertation, we presented researches which utilize colors as a critical channel for visualizing multiple values in multivariate data in a scalable manner. In Parallel Coordinates Plot, we focused on dealing with scalability in the number of items, and in IssueML, multiple complicated attributes were dealt with patterns of colors. Furthermore, we discussed about how the visual encoding can support users interaction with multiple items in TRaVis. While the large size and complexity in data is generally considered as a nuisance in the field of information visualization, we believe they also present new opportunities for supporting a broader audience of users in facilitating diverse perspectives in understanding and interacting with the data, with the help of visualizations.

Bibliography

[1] D. Albers, C. Dewey, and M. Gleicher. Sequence surveyor: leveraging overview for scalable genomic alignment visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2392–2401, December 2011.

[2] Danielle Albers, Michael Correll, and Michael Gleicher. Task-driven evaluation of aggregation in time series visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 551–560, 2014.

[3] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *IEEE Symposium on Information Visualization*, pages 81–88, October 2004.

[4] Social Security Administration. Popular baby names, February 2023.

[5] Michael Batty. Rank clocks. *Nature*, 444(7119):592, 2006.

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[7] David Borland and Russell M Taylor Ii. Rainbow color map (still) considered harmful. *IEEE computer graphics and applications*, 27(2):14–17, 2007.

[8] J. H. T. Claessen and J. J. van Wijk. Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2310–2316, December 2011.

[9] William S Cleveland and Robert McGill. Graphical perception: theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.

[10] Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. Imaxes: immersive axes as embodied affordances for interactive multivariate data visualisation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 71–83, Qu&#233;bec City, QC, Canada. ACM, 2017.

[11] Michael Correll, Dominik Moritz, and Jeffrey Heer. Value-suppressing uncertainty palettes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2018.

[12] A. Dasgupta and R. Kosara. Pargnostics: screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1017–1026, November 2010.

[13] J. Díaz, M. Fort, and P. Vázquez. Tourvis: narrative visualization of multistage bicycle races. *Computer Graphics Forum*, 40(3):531–542, 2021.

[14] Marian Dörk, Sheelagh Carpendale, and Carey Williamson. The information flaneur: a fresh look at information seeking. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1215–1224, 2011.

[15] G. Ellis and A. Dix. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):717–724, September 2006.

[16] esmeel. 6 k-pop songs that entered the charts long after their release. *Soompi*, May 2021. `https://www.soompi.com/article/1465703wpp/6-k-pop-songs-that-entered-the-charts-long-after-their-release`.

[17] E. Fanea, S. Carpendale, and T. Isenberg. An interactive 3d integration of parallel coordinates and star glyphs. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* Pages 149–156, October 2005.

[18] Fangraphs baseball, July 2018.

[19] FIFA. Fifa Men's Ranking, April 2023.

[20] Ying-Huey Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings Visualization '99 (Cat. No.99CB37067)*, pages 43–508, October 1999.

[21] Z. Geng, Z. Peng, R. S.Laramee, J. C. Roberts, and R. Walker. Angular histograms: frequency-based visualizations for large, high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2572–2580, December 2011.

[22] Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, and Anastasia Bezerianos. Comparing similarity perception in time series visualizations. *IEEE transactions on visualization and computer graphics*, 25(1):523–533, 2018.

[23] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2277–2286, December 2013.

[24] H. Guo, H. Xiao, and X. Yuan. Scalable multivariate volume visualization and analysis based on dimension projection and parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1397–1410, September 2012.

[25] Huijie Guo, Meijun Liu, Bowen Yang, Ye Sun, Huamin Qu, and Lei Shi. Rankfirst: visual analysis for factor investment by ranking stock timeseries. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[26] Dongming Han, Jiacheng Pan, Fangzhou Guo, Xiaonan Luo, Yingcai Wu, Wenting Zheng, and Wei Chen. Rankbrushers: interactive analysis of temporal ranking ensembles. *Journal of Visualization*, 22(6):1241–1255, 2019.

[27] Mark Harrower and Cynthia A Brewer. Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[28] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.* Pages 127–130, October 2002.

[29] Susan Havre, Beth Hetzler, and Lucy Nowell. Themeriver: visualizing theme changes over time. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, pages 115–123. IEEE, 2000.

[30] Jeffrey Heer and Michael Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 203–212, 2010.

[31] Jeffrey Heer and Maureen Stone. Color naming models for color selection, image editing and palette design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1007–1016, 2012.

[32] J. Heinrich and D. Weiskopf. Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1531–1538, November 2009.

[33] Julian Heinrich, Yuan Luo, Arthur E Kirkpatrick, Hao Zhang, and Daniel Weiskopf. Evaluation of a bundling technique for parallel coordinates. *arXiv preprint arXiv:1109.6073*, 2011.

[34] Julian Heinrich and Daniel Weiskopf. State of the art of parallel coordinates. In *Eurographics (STARs)*, pages 95–116, 2013.

[35] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates for visualizing multi-dimensional geometry. In *Computer Graphics 1987*, pages 25–44. Springer, 1987.

[36] Takayuki Itoh, Ashnil Kumar, Karsten Klein, and Jinman Kim. High-dimensional data visualization by interactive construction of low-

dimensional parallel coordinate plots. *Journal of Visual Languages Computing*, 43:1–13, 2017.

[37] Halldór Janetzko, Manuel Stein, Dominik Sacha, and Tobias Schreck. Enhancing parallel coordinates: statistical visualizations for analyzing soccer data. *Electronic Imaging*, 2016(1):1–8, 2016.

[38] A. Jeong. K-pop: stream like you breathe. *Korea Expose*, November 2017. `https://www.koreaexpose.com/k-pop-stream-breathe/`.

[39] Jira Software. `https://www.atlassian.com/software/jira`, February 2023.

[40] J. Johansson, M. Cooper, and M. Jern. 3-dimensional display for clustered multi-relational parallel coordinates. In *Ninth International Conference on Information Visualisation (IV'05)*, pages 188–193, July 2005.

[41] J. Johansson and C. Forsell. Evaluation of parallel coordinates: overview, categorization and guidelines for future research. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):579–588, January 2016.

[42] Jimmy Johansson, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing structure in visualizations of dense 2d and 3d parallel coordinates. *Information Visualization*, 5(2):125–136, June 2006.

[43] D. A. Keim, M. C. Hao, U. Dayal, and M. Lyons. Value-cell bar charts for visualizing large transaction data sets. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):822–833, July 2007.

[44] Robert Kincaid and Heidi Lam. Line graph explorer: scalable display of line graphs using focus+context. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '06, pages 404–411, Venezia, Italy. ACM, 2006.

[45] H. Kobayashi, T. Furukawa, and K. Misue. Parallel box: visually comparable representation for multivariate data analysis. In *2014 18th International Conference on Information Visualisation*, pages 183–188, July 2014.

[46] R. Kosara, F. Bendix, and H. Hauser. Parallel sets: interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, July 2006.

[47] Bongshin Lee, Cynthia Sims Parr, Catherine Plaisant, Benjamin B Bederson, Vladislav Daniel Veksler, Wayne D Gray, and Christopher Kotfila. Treeplus: interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414–1426, 2006.

[48] Sungkil Lee, Mike Sips, and Hans-Peter Seidel. Perceptually driven visibility optimization for categorical data visualization. *IEEE Transactions on visualization and computer graphics*, 19(10):1746–1757, 2012.

[49] T. Van Long. A new metric on parallel coordinates and its application for high-dimensional data visualization. In *2015 International Conference on Advanced Technologies for Communications (ATC)*, pages 297–301, October 2015.

[50] Kecheng Lu, Mi Feng, Xin Chen, Michael Sedlmair, Oliver Deussen, Dani Lischinski, Zhanglin Cheng, and Yunhai Wang. Palettailor: discriminable colorization for categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):475–484, 2020.

[51] L. F. Lu, M. L. Huang, and T. Huang. A new axes re-ordering method in parallel coordinates visualization. In *2012 11th International Conference on Machine Learning and Applications*, volume 2, pages 252–257, December 2012.

[52] Kevin T McDonnell and Klaus Mueller. Illustrative parallel coordinates. In *Computer Graphics Forum*, volume 27 of number 3, pages 1031–1038. Wiley Online Library, 2008.

[53] Kakao Entertainment. Melon, February 2023.

[54] Sebastian Mittelstädt, Andreas Stoffel, and Daniel A Keim. Methods for compensating contrast effects in information visualization. In *Computer Graphics Forum*, volume 33 of number 3, pages 231–240. Wiley Online Library, 2014.

[55] Daniele Nadalutti and Luca Chittaro. Visual analysis of users' performance data in fitness activities. *Computers & Graphics*, 31(3):429–439, 2007.

[56] H. Nguyen and P. Rosen. Dspcp: a data scalable approach for identifying relationships in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 24(3):1301–1315, March 2018.

[57] K. Nohno, H. Wu, K. Watanabe, S. Takahashi, and I. Fujishiro. Spectral-based contractible parallel coordinates. In *2014 18th International Conference on Information Visualisation*, pages 7–12, July 2014.

[58] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, September 2006.

[59] G. Oliveira, J. Comba, R. Torchelsen, M. Padilha, and C. Silva. Visualizing running races through the multivariate time-series of multiple runners. In *2013 XXVI Conference on Graphics, Patterns and Images*, pages 99–106, August 2013.

[60] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauf. An edge-bundling layout for interactive parallel coordinates. In *2014 IEEE Pacific Visualization Symposium*, pages 57–64, March 2014.

[61] Wei Peng, M. O. Ward, and E. A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *IEEE Symposium on Information Visualization*, pages 89–96, October 2004.

[62] C. Perin, J. Boy, and F. Vernier. Using gap charts to visualize the temporal evolution of ranks and scores. *IEEE Computer Graphics and Applications*, 36(5):38–49, September 2016.

[63] Charles Perin, Romain Vuillemot, and Jean-Daniel Fekete. A table!: improving temporal navigation in soccer ranking tables. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 887–896, Toronto, Ontario, Canada. ACM, 2014.

[64] H. Qu, H. Qu, H. Qu, H. Qu, W. Chan, W. Chan, W. Chan, W. Chan, A. Xu, A. Xu, A. Xu, A. Xu, K. Chung, K. Chung, K. Chung, K. Chung, K. Lau, K. Lau, K. Lau, K. Lau, P. Guo, P. Guo, P. Guo, and P. Guo. Visual analysis of the air pollution problem in hong kong. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1408–1415, November 2007.

[65] R. G. Raidou, M. Eisemann, M. Breeuwer, E. Eisemann, and A. Vilanova. Orientation-enhanced parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):589–598, January 2016.

[66] Ramana Rao and Stuart K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 318–322, Boston, Massachusetts, USA. ACM, 1994.

[67] 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*, 25(3):1575–1590, March 2019.

[68] Bernice E Rogowitz and Lloyd A Treinish. Data visualization: the end of the rainbow. *IEEE spectrum*, 35(12):52–59, 1998.

[69] Takafumi Saito, Hiroko Nakamura Miyamura, Mitsuyoshi Yamamoto, Hiroki Saito, Yuka Hoshiya, and Takumi Kaseda. Two-tone pseudo coloring: compact visualization for one-dimensional data. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* Pages 173–180. IEEE, 2005.

[70] C. Shi, W. Cui, S. Liu, P. Xu, W. Chen, and H. Qu. Rankexplorer: visualization of ranking changes in large time series data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2669–2678, December 2012.

[71] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, September 1996.

[72] Maureen Stone. *A field guide to digital color*. CRC Press, 2016.

[73] Bruce J Swihart, Brian Caffo, Bryan D James, Matthew Strand, Brian S Schwartz, and Naresh M Punjabi. Lasagna plots: a saucy alternative to spaghetti plots. *Epidemiology (Cambridge, Mass.)*, 21(5):621, 2010.

[74] Alice Thudt, Uta Hinrichs, and Sheelagh Carpendale. The bohemian bookshelf: supporting serendipitous book discoveries through information visualization. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1461–1470, 2012.

[75] Edward R Tufte. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.

[76] Lisa Tweedie, Bob Spence, Huw Dawkes, and Hua Su. The influence explorer. In *Conference Companion on Human Factors in Computing Systems*, CHI '95, pages 129–130, Denver, Colorado, USA. ACM, 1995.

[77] Lisa Tweedie, Bob Spence, David Williams, and Ravinder Bhogal. The attribute explorer. In *Conference Companion on Human Factors in Computing Systems*, CHI '94, pages 435–436, Boston, Massachusetts, USA. ACM, 1994.

[78] UCI Machine Learning Repository, July 2018.

[79] Eric Firing Nathaniel J. Smith Stefan van der Walt. MPL colormaps, February 2023.

[80] Vizster: visualizing online social networks. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* Pages 32–39. IEEE, 2005.

[81] Romain Vuillemot and Charles Perin. Investigating the direct manipulation of ranking tables for time navigation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2703–2706, Seoul, Republic of Korea. ACM, 2015.

[82] James Walker, Zhao Geng, Mark Jones, and Robert S Laramee. Visualization of large, time-dependent, abstract data with integrated spherical and parallel coordinates. *Eurographics Association, Vienna, Austria*:43–47, 2012.

[83] R. Walker, P. A. Legg, S. Pop, Z. Geng, R. S. Laramee, and J. C. Roberts. Force-directed parallel coordinates. In *2013 17th International Conference on Information Visualisation*, pages 36–44, July 2013.

[84] J. Wang, X. Liu, H. Shen, and G. Lin. Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):81–90, January 2017.

[85] Lujin Wang, Joachim Giesen, Kevin T McDonnell, Peter Zolliker, and Klaus Mueller. Color design for illustrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1739–1754, 2008.

[86] Yunhai Wang, Xin Chen, Tong Ge, Chen Bao, Michael Sedlmair, Chi-Wing Fu, Oliver Deussen, and Baoquan Chen. Optimizing color assignment for perception of class separability in multiclass scatterplots. *IEEE transactions on visualization and computer graphics*, 25(1):820–829, 2018.

[87] Colin Ware. *Information visualization: perception for design*. Morgan Kaufmann, 2019.

[88] J. Xia, Y. Hou, Y. V. Chen, Z. C. Qian, D. S. Ebert, and W. Chen. Visualizing rank time series of wikipedia top-viewed pages. *IEEE Computer Graphics and Applications*, 37(2):42–53, March 2017.

[89] Jing Yang, Wei Peng, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714)*, pages 105–112, October 2003.

[90] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1001–1008, November 2009.

[91] Z. Zhang, K. T. McDonnell, and K. Mueller. A network-based interface for the exploration of high-dimensional data spaces. In *2012 IEEE Pacific Visualization Symposium*, pages 17–24, February 2012.

[92] Liang Zhou and Charles D Hansen. A survey of colormaps in visualization. *IEEE transactions on visualization and computer graphics*, 22(8):2051–2069, 2015.

국문 초록

정보시각화에서는 인간의 시각적 인지 능력의 활용을 극대화하기 위해 데이터를 그래픽의 형태로 표현하여 시용자가 그래픽을 통해 데이터를 탐색할 수 있도록 돕는다. 그러나, 증가하는 데이터의 크기와 복잡도로 인하여 시각화에 확장성 (scalability)을 제공하는 문제는 정보시각화의 주요 연구 주제로 발전하였다. 이를 해결하기 위하여 데이터의 정보를 다양한 측면에서 추출하고 그들을 효과적으로 조합하여 전체적인 데이터와 상호작용 할 수 있도록 돕는 다양한 방법론들이 제시되었다. 그러나 이러한 노력에도 불구하고, 사용자가 필수적으로 여러 개별 데이터의 여러 값들과 직접 직면하는 상황은 불가피할게 발생하며, 이는 특히 사용자가 활용할 수 있는 자원이 부족할 경우 더욱 빈번하게 발생한다. 이러한 경우 사용자는 다시 개별적인 아이템과의 상호작용으로 인한 확장성 문제에 직면할 수 밖에 없게 된다.

이러한 한계점들에서 착안하여, 본 논문에서는 색상을 주요 시각화 방법론으로 활용하여 큰 사이즈의 다변량 데이터 (multivariate data) 에서의 확장성 문제를 해결하는 방법들을 제시한다. 연구에서는 한정된 공간 안에 많은 값들을 표현하고 사용자가 여러 값의 패턴을 통해 데이터를 이해하고 상호작용할 수 있도록 돕는 색상의 장점을 활용한다. 많은 아이템을 시각화하는데 있어서의 확장성을 위해서, parallel coordinates plot (PCP)에서 발생하는 확장성과 관련된 한계점들을 이산적인 색상 체계 (discrete color scheme)가 부여된 히스토그램으로 극복하는 Parallal Histogram Plot (PHP)를 고안하였다. 색상이 적용된 히스토그램을 통해 사용자는 겹침 문제나 확장성 문제 없이 전체 데이터의 개요를 확인 할 수 있다. PHP에서 각 히스토그램의 사각형은 선택한 속성의 순위를 기준으로 색상이 부여된다. 이러한 색상 부여 방식을 통해 사용자는 축의 재배치나 재정렬 없이도 멀리

떨어진 속성들간의 관계 또한 관찰 할 수 있다. 데이터의 복잡하고 많은 속성을 시각화하는데 있어서의 확장성을 위해, 본 연구에서는 거대 소프트웨어 개발 중 발생하는 여러 이슈들을 분석하고 모니터링 하는것을 지원하는 시각적 시스템인 IssueML을 개발하였다. 전문가와의 인터뷰를 기반으로, IssueML은 시간 경과에 따른 여러 이슈와 그들의 진행 상황을 모니터링하기 위한 각종 시각화 기법으로 구성되어 있다. Visual Information Seeking Mantra를 따르는 여러 협응하는 시각화들의 활용을 통해 사용자는 확장성 높은 방법으로 여러 이슈의 경과를 모니터링 및 분석 할 수 있다. 끝으로, 사용자의 여러 아이템과의 상호작용을 지원하기 위해, 시간적 순위 데이터 시각화의 새로운 접근 방식인 TRaVis를 제안하였다. TRaVis에서는 각각의 순위 변화를 색상이 부여 된 단일 행 (column) 으로 표현하며, 이들은 공간상에서 서로 겹치지 않게 쌓은 방식으로 표현된다. 이러한 히트맵과 유사한 방식의 시각화는 여러 아이템의 순위 변화를 서로 겹치지 않게 관찰 할 수 있도록 한다. 시각화에서 항목이 쌓이는 순서를 변경함으로써 사용자는 TRaVis를 통해 시간에 따른 순위 변화가 정렬 기준에 따라 어떤 특징이 있는지를 관찰 할 수 있으며, 이는 사용자의 다양한 호기심 있는 탐색 과정을 지원한다. 끝맺음으로, 세 가지 연구를 기반으로 배운 교훈을 논의하고 향후 연구 방향을 제안한다.

주요어: 정보 시각화; 셀프 모니터링; 사용자 상호작용; 색상

학번: 2015-22900