DATA SCALABLE APPROACH FOR IDENTIFYING CORRELATION IN LARGE AND MULTIDIMENSIONAL DATA

by

Hoa Thanh Nguyen

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

School of Computing

The University of Utah

August 2017

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Hoa Thanh Nguyen**

has been approved by the following supervisory committee members:

Paul Rosen ,	Chair(s)	**05 Jan 2017** Date Approved
Christopher Johnson ,	Member	**05 Jan 2017** Date Approved
Charles Hansen ,	Member	**05 Jan 2017** Date Approved
Matthew Might ,	Member	**05 Jan 2017** Date Approved
Edward Wes Bethel ,	Member	**09 Jan 2017** Date Approved

by **Ross Whitaker** , Chair/Dean of

the Department/College/School of **Computing**

and by **David B. Kieda** , Dean of The Graduate School.

ABSTRACT

Correlation is a powerful relationship measure used in many fields to estimate trends and make forecasts. When the data are complex, large, and high dimensional, correlation identification is challenging. Several visualization methods have been proposed to solve these problems, but they all have limitations in accuracy, speed, or scalability. In this dissertation, we propose a methodology that provides new visual designs that show details when possible and aggregates when necessary, along with robust interactive mechanisms that together enable quick identification and investigation of meaningful relationships in large and high-dimensional data. We propose four techniques using this methodology. Depending on data size and dimensionality, the most appropriate visualization technique can be provided to optimize the analysis performance.

First, to improve correlation identification tasks between two dimensions, we propose a new correlation task-specific visualization method called correlation coordinate plot (CCP). CCP transforms data into a powerful coordinate system for estimating the direction and strength of correlations among dimensions. Next, we propose three visualization designs to optimize correlation identification tasks in large and multidimensional data. The first is snowflake visualization (Snowflake), a focus+context layout for exploring all pairwise correlations. The next proposed design is a new interactive design for representing and exploring data relationships in parallel coordinate plots (PCPs) for large data, called data scalable parallel coordinate plots (DSPCP). Finally, we propose a novel technique for storing and accessing the multiway dependencies through visualization (MultiDepViz).

We evaluate these approaches by using various use cases, compare them to prior work, and generate user studies to demonstrate how our proposed approaches help users explore correlation in large data efficiently. Our results confirmed that CCP/Snowflake, DSPCP, and MultiDepViz methods outperform some current visualization techniques such as scatterplots (SCPs), PCPs, SCP matrix, Corrgram, Angular Histogram, and UntangleMap in both accuracy and timing. Finally, these approaches are applied in real-world applications such as a debugging tool, large-scale code performance data, and large-scale climate data.

CONTENTS

ACKNOWLEDGMENTS

This journey would not have been possible without the support of my advisor, my committee members, and my friends and family. I would like to thank my advisor, Paul Rosen. His guidance has helped me achieve success in my doctoral program and prepared me for my research journey. I thank him for his time and energy, which has inspired me in my research. I would like thank my committee members, Chris Johnson, Chuck Hansen, Matthew Might, and Wes Bethel. They have been invaluable mentors to me, and have helped to improve the quality of my research and dissertation. I would like to thank Christine Pickett for greatly helping me with revising my research papers and dissertation.

I would also like to thank my colleagues who gave me valuable feedback on my research and dissertation. I would also like to thank my funding agents, NSF CIF21 DIBBs (ACI-1443046), Lawrence Livermore National Laboratory, Pacific Northwest National Laboratory Analysis in Motion (AIM) Initiative, Lawrence Berkeley National Laboratory, and the Scientific Computing and Imaging Institute.

I would like to thank my family for encouraging me in all of my pursuits. I am grateful to my parents and my husband, James King, who supported me emotionally and helped to take care of my son while I was working on my research. I also would like to thank to my son, Ken King, who is my motivation to go through this journey.

CHAPTER 1

INTRODUCTION

1.1 Challenges

Correlation is a powerful metric [1] that provides a predictive relationship between variables used in many areas of science [2], engineering, and business [3]. A correlation coefficient is a measure of the strength and direction of such a relationship. Correlation is a powerful statistical tool, but visual examination is critical to interpret data.

The many-to-one relationship between data and a correlation coefficient may obscure important features of the data. In Anscombe's quartet (see Fig. 1.1) [4], four distributions (i.e., the many relationship) have identical correlation coefficients (i.e., the one relationship). Visual examination can disambiguate the variations to outliers (case 1), noise (case 2), nonlinearity (case 3), and nonrelationship (case 4).

Many visualization methods have been proposed to improve correlation identification in large and multidimensional data. Both scatterplots (SCP) [5], as shown in Fig. 1.2, and parallel coordinates plots (PCP) [6], as shown in Fig. 1.3, are capable of being used to investigate correlation. However, one should not infer that these are the *ideal* tools for performing such a task. The critical shortcoming of these methods is in their design goal—they are designed as general purpose tools for performing a wide variety of analytic tasks. No special consideration has been made to any single task, meaning that while they *can be* used to identify correlation, they are *not designed optimally* for it.

Additionally, the challenge of correlation identification is exacerbated by the increasing desire to analyze large multidimensional data. A number of multiattribute visualization techniques exist for this analysis, with scatterplot matrices (SPLOMs) [7] and PCPs [8], [9] remaining the most popular. SPLOMs simultaneously show all possible combinations of attribute, but the plots become small as the number of combinations grows quadratically. For PCPs, the series of axes grow linearly. However, the limit of pairwise comparisons requires offloading the task to interaction.

Furthermore, in large datasets with a high number of data points, parallel coordinates plots are

commonly used for correlation identification in many analyses. Parallel coordinates plots (PCPs) have been widely studied in visualization, yet their adoption outside the community has been slow. The number of publications with the term "parallel coordinates" in the title has been rising steadily from 14 in 1991 to 543 in 2011, with 5620 total publications as of December 2012 [10]. Although some in the community find PCPs to be a valuable way to analyze and interact with their multiattribute data, the challenges faced in widespread adoption are twofold. First, PCPs can be difficult to interpret for inexperienced users, requiring training [11]. Second, technical issues with overdraw [12], order of axes, line tracing, nominal and ordinal data, time series, pattern recognition [13], and uncertainty [14], make them impractical for many scenarios.

The visualization methods presented above deal with large data but not many dimensions. Corrgrams [15] display a matrix of correlation glyphs. These glyphs scale well and give the user quick access to summary statistics, but they may lose important data features (e.g., Anscombe's quartet). Finally, multivariate relationship analysis is an important task in visual analytics [16]. To gain insight from the complex multivariate data [17], a number of analysis approaches have been proposed, such as sampling [18], clustering [19], reducing the number of variables [20], or the introduction of object and dimensional correlation during projection from multidimensional space to 3D [21].

UnTangle Map [22] proposed a triangle mesh layout based on a greedy algorithm to represent triangle meshes for sets of three variables. This method does not represent all possible relationship triangles but chooses the most relevant ones. It works well to identify the multiway dependencies that can break down to a set of three-way dependencies as sets of triangles. However, when users need to understand the whole picture of all potential multiway relationships, both relevant and irrelevant ones, they need a new layout that supports them to quickly perform this task.

When the number of dimensions is not large, providing methods that show detail to improve correlation identification is possible. However, when the number of dimensions is high, identifying correlations at the summary level, while showing detail is difficult. Therefore, methods that improve correlation identification in both large and multidimensional data are challenging.

1.2 Dissertation Statement

Concentrating on interactive and data scalable design for visualizing correlation can improve its identification in large and multidimensional data. This methodology provides new visual designs

along with robust interactive mechanisms that show details when possible, and aggregates when necessary, and enables quickly identifying and investigating meaningful relationships. Depending on data size and dimensionality, the most appropriate visualization technique can be selected to boost analysis performance.

First, with these limitations of SCP and PCP, we have developed a new, correlation task-specific visual design called correlation coordinate plots [23], [24], or CCPs as shown in Fig. 1.4. CCPs use design attributes, such as axis shape and a simple, yet effective, point transform to enable quick and accurate determination of correlation direction and strength. Our user study confirmed that CCPs help users identify correlation more accurately and at a faster speed than SCP and PCP.

Second, to overcome the challenge of correlation identification in the medium-level scope, we have further developed a new focus+context style circular layout for CCPs, called the Snowflake Visualization, as shown in Fig. 1.5. This visualization represents a compromise where the screen space needed to represent additional attributes grows linearly for the focus region and quadratically for the context region, with some reliance on interaction for full investigation. We have also extended the visual metaphors of the CCP to support principal component analysis of data, which provides a single visual interface for multidimensional analysis.

Third, PCP use is popular in the visualization community because users can identify trends of the data. However, arguably the greatest technical challenge for PCPs is that of overdraw, which occurs when large numbers of overlapping lines obscure the patterns in the data. Unfortunately, this problem makes standard PCPs difficult to use for large, noisy, or complex data. Three important visual features are used in analysis with PCPs: the angles of line segments, the locations of line segment crossings, and the distribution of line segments. The majority of approaches to correct overdraw in parallel coordinates have unfortunately broken one or more of these properties. We propose a new paradigm for representing relationships in PCPs that overcomes the overdraw problem, while simultaneously maintaining these properties. We call this approach data scalable parallel coordinates (DSPCP) [25], [26], as shown in Fig. 1.6. DSPCP exploits the point/line duality property of PCPs and a local linear assumption of data to extract and to represent relationship summarizations. DSPCP simultaneously shows relationships in the data and the consistency of those relationships. It supports various visualization tasks including cluster analysis, mixed linear and nonlinear pattern identification, hidden pattern detection, and outlier detection, all in large data. We demonstrate the results on multiple synthetic and real data sets.

Finally, we propose a method that improves the identification of correlation in large and multidimensional data [27], [26]. This approach is a novel interactive context+focus visualization technique (MultiDepViz) as shown in Fig. 1.7. Our data structure is inspired by the algebraic topology notion of a simplicial complex, where one-, two-, and three-simplices are used to model the pairwise, three-way, and four-way dependencies, respectively. We also inherit UnTangle Map to represent the probability relationship between pairwise and three-way dependencies in detailed view and extended visual encodings to represent four-way dependencies. Exploration is supported by a variety of operations placed on the complex, and interactive visualization enables flexible investigations through overview and detailed views of the data. The overview provides a global visual exploration interface for filtering and selecting a candidate set of relationships, and the detailed views helps in developing inferences by providing specific information about those relationships selected. We provide various use cases and compare them to the prior work to demonstrate how our proposed approach helps to explore these multiway dependencies in data efficiently.

Fig. 1.8 shows a comparison between current methods (in blue) and our four new approaches (in red) that we propose to improve correlation identification with the low and high size of dimensions and size of data items. The circle glyph shows that the method represents individual data items, and the box glyph shows that the method provides a summary of data items. CCP and snowflake visualization represent individual data items, which improves correlation identification in low and medium number of dimensions and the size of data. DSPCP represents large-scale data, and MultiDepViz could handle both the high numbers of dimensions and the large size of the data. Our approaches improve correlation identification in large and multidimensional datasets compared to the conventional methods such as SCP, PCP, UntangleMap, Angular Histogram, and Corrgram as demonstrated by user studies and case studies in the following chapters of this dissertation.

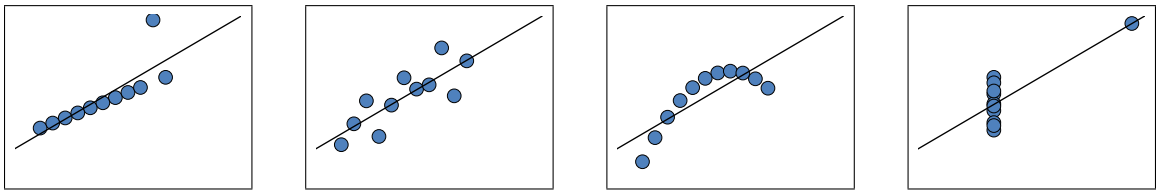


Figure 1.1: Anscombe's quartet (data from [4]) shows four visually different distributions that have the same correlation coefficient of 0.816.

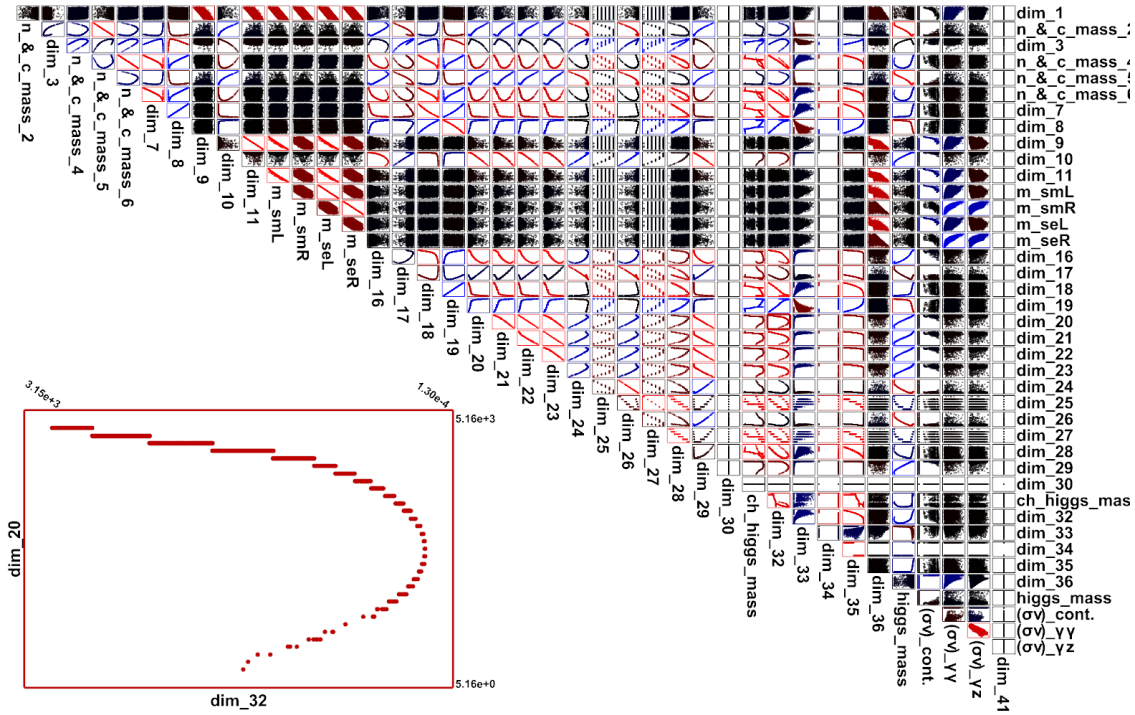


Figure 1.2: Scatter plots matrix.

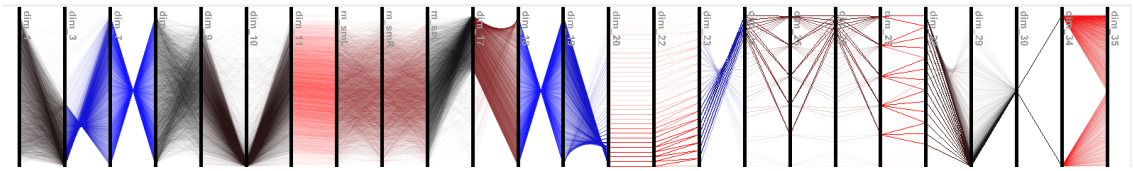
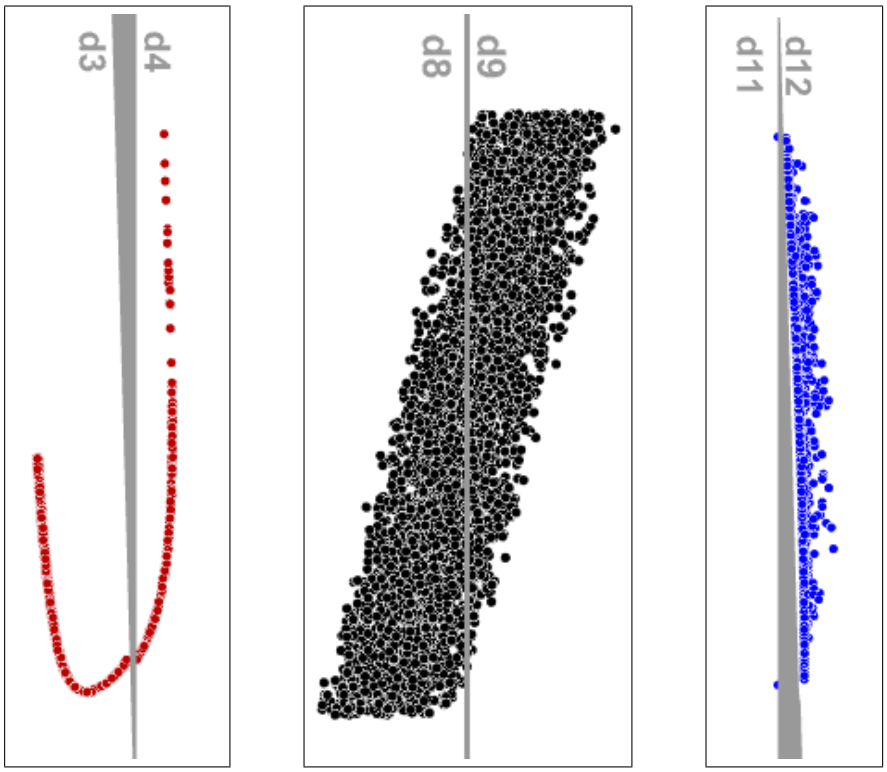


Figure 1.3: Parallel coordinates.



(a) Positive (b) No Correlation (c) Negative

Figure 1.4: Correlation coordinate plots (CCPs) transform data into a coordinate system more suited to identifying correlation between two attributes. (a-c): Example CCPs show positive correlation, no correlation, and negative (or anti-) correlation, respectively.

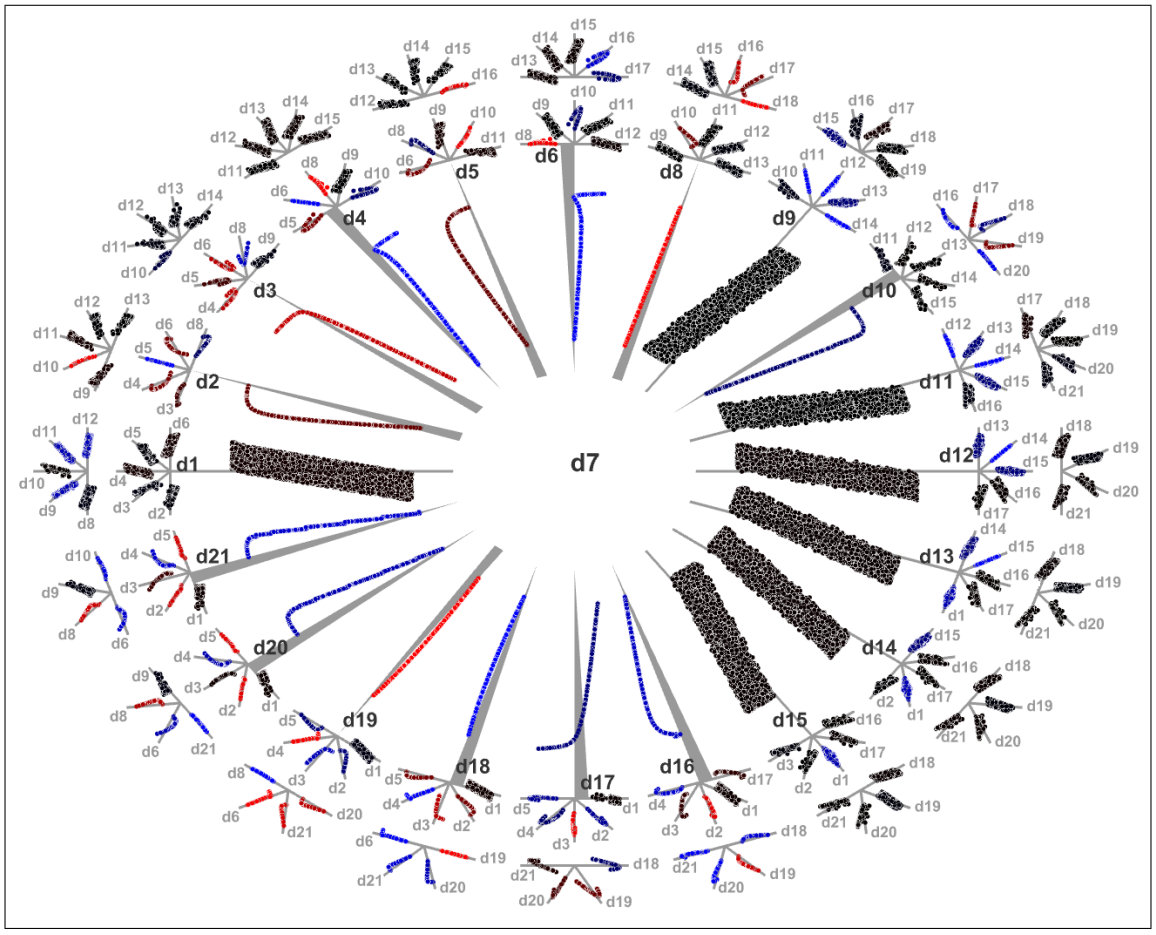


Figure 1.5: The snowflake visualization is a focus+context interface that combines CCPs for one attribute to all others in the middle (i.e., the focus) and CCPs for all other attribute pairings on the perimeter (i.e., the context).

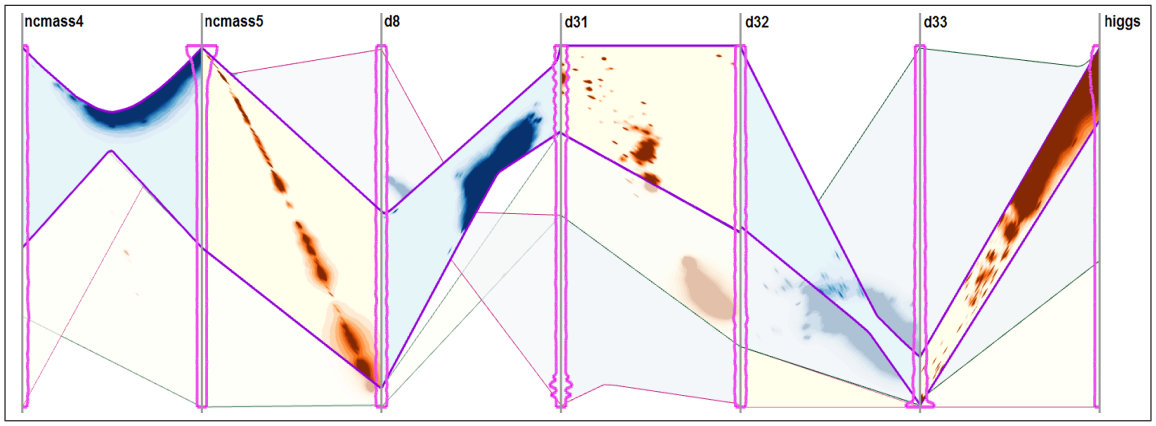
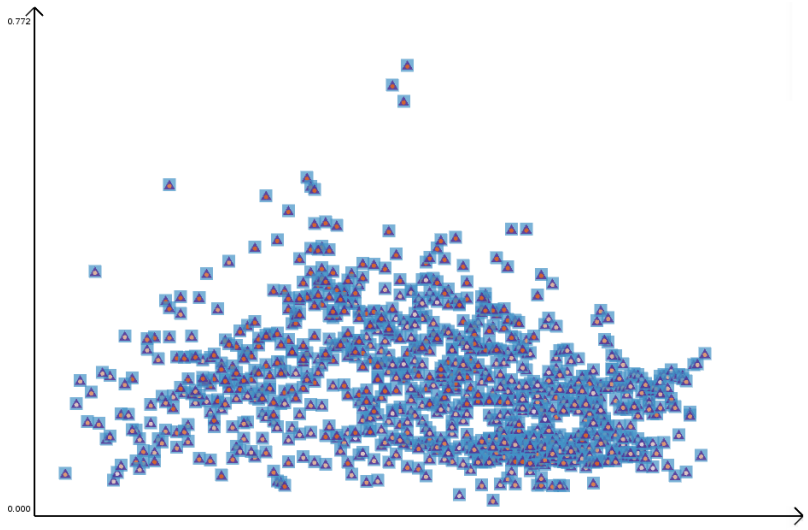
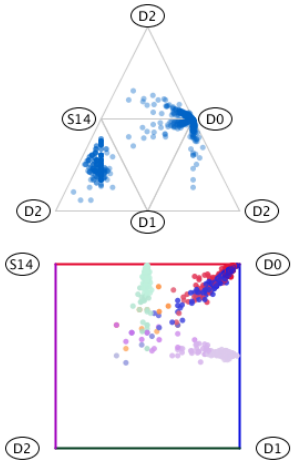


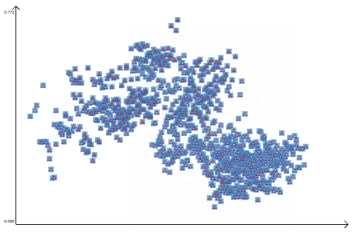
Figure 1.6: While large data overwhelms conventional PCPs, our approach uses flexible relationship clustering and summarization to identify large-scale trends in the data, simultaneously highlighting adherence to the trend and showing outlier behavior. Here, trends are tracked across multiple data attributes.



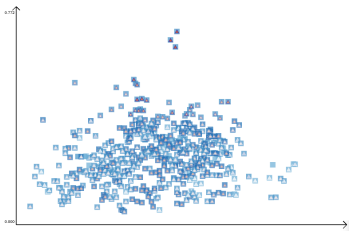
(a) Overview.



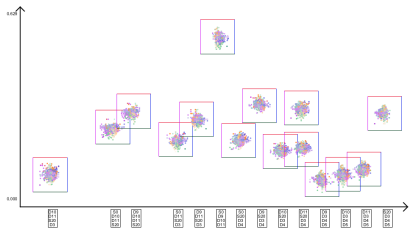
(b) Detailed View.



(c) Reordering.



(d) Filtering.



(e) Selection.

Figure 1.7: Overview+detail of the marketing research dataset. Our approach for representing multiway dependencies in multivariate data begins with (a) an overview supported by a glyph representation of pairwise, three-way, and four-way relationships for all sets of four variables. The overview can be (c) reordered, (d) filtered, and (e) zoomed until a relationship of interest is identified. A selected glyph then populates the detailed view (b).

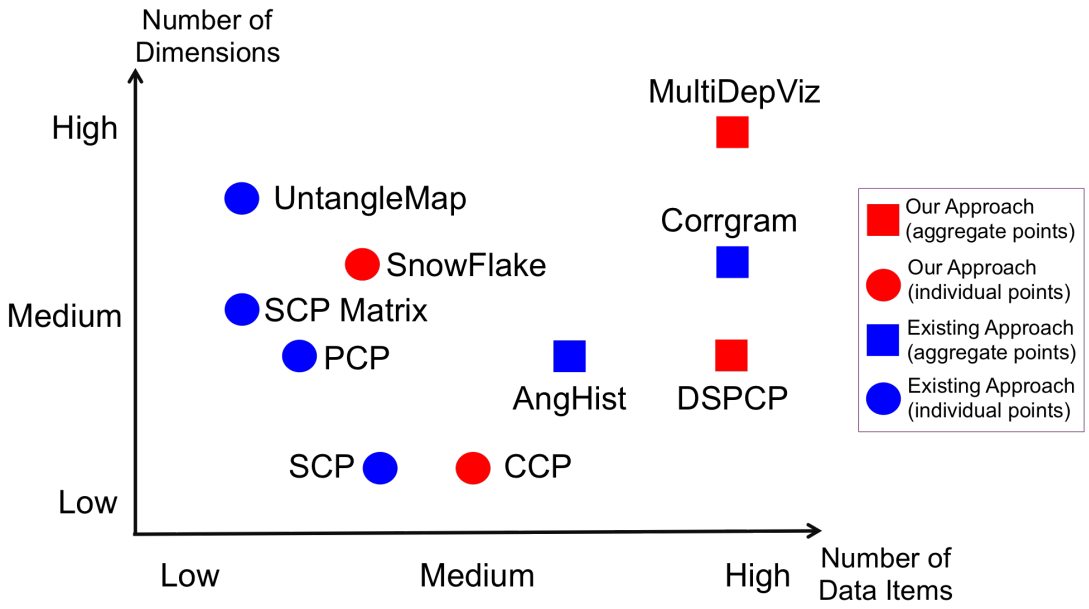


Figure 1.8: Visualization methods for correlation identification.

CHAPTER 2

PREVIOUS WORK

2.1 Correlation

Correlation is a statistical relationship between data and can be used to model and predict relationships [1], [3]. The "quality of the relationship" is often measured using a correlation coefficient [28], [29], with a positive correlation indicating two attributes are increasing together, whereas negative or anti-correlation indicates that one attribute increases and the other decreases. There are several correlation coefficient measures, the most common of which is the Pearson correlation coefficient (PCC) [30], [31]. PCC, $\rho(x,y)$, measures the linear relationship between two attributes x and y with means $\bar{x}$ and $\bar{y}$ and standard deviations σ_x and σ_y [32], [33]. It is defined as

$$\rho(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y} = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y}. \tag{2.1}$$

When identifying correlations, finding nonlinear relationships [28] can also be important. While linear correlation represents a line that best describes the relationship between two attributes, the nonlinear case can take any shape that implies a relationship between the attributes.

2.2 Multivariate Dependency and the Coefficient of Determination R^2

The coefficient of determination, or R^2, is used to measure how well actual data values fit those derived from a model-fitting procedure [34], [35]. Our usage of the measure is under the context of multiple correlation, which is a measure of how well a given (dependent) variable can be predicted using a linear combination of other (independent) variables. The value of R^2 ranges between zero and one, where a higher value indicates a better predictability of the dependent variable. A value of one indicates that the predictions are exactly correct, and a value of zero indicates that no linear combination of the independent variables is a better predictor than the fixed mean of the dependent variable.

Multiple correlation requires the selection of a set of independent variables, $x_1, x_2, ..., x_N$, and a single dependent, y. R^2 can then be computed using the following equation.

$$R^2 = \mathbf{c}^T R_{xx}^{-1} \mathbf{c}. \tag{2.2}$$

where the vector $\mathbf{c} = (r_{x_1 y}, r_{x_2 y}, ..., r_{x_n y})^T$ contains the pairwise correlation $r_{x_i y}$ between the independent variables x_i and the dependent variable y. The correlation matrix R_{xx} represents the intercorrelations between independent variables and takes the form

$$R_{xx} = \begin{pmatrix} r_{x_1 x_1} & r_{x_1 x_2} & \cdots & r_{x_1 x_N} \\ r_{x_2 x_1} & \ddots & & \vdots \\ \vdots & & \ddots & \\ r_{x_N x_1} & \cdots & & r_{x_N x_N} \end{pmatrix} \tag{2.3}$$

If all the independent variables are uncorrelated, the matrix R_{xx} is the identity matrix, and R^2 simply equals $\mathbf{c}^T \mathbf{c}$, the sum of the squared correlations with the dependent variable. If the predictor variables are correlated among themselves, R_{xx}^{-1} will account for this.

2.3 Scatterplot

A scatterplot (SCP) [5], [36] is a simple plot of points used to investigate the linear and nonlinear relationships between two attributes [37]. The patterns of importance in the context of correlation are when the data points slope from lower left to upper right, suggesting a positive correlation, whereas a pattern sloping from upper left to lower right suggests a negative correlation. The direction of correlation (positive or negative) can be confusing to novice users. More importantly, the strength of the correlation (high versus low correlation) can at times be difficult to interpret.

For multidimensional data, a scatterplot matrix (SPLOM) [37], [38] shows the relationships of all pairs of attributes by organizing a grid of SCPs with each attribute occupying one row and one column. As the number of attributes increases, the number of plots grows quadratically, making it difficult to present all the data. This problem can be mitigated by approaches such as Corrgrams [15], which display a matrix of correlation glyphs. These glyphs scale well and give the user quick access to summary statistics, but they may lose important data features (e.g., Anscombe's Quartet). In other cases, navigation can be used to search larger spaces [39].

Navigation also can be used to help search larger spaces [39]. Another method, based on flow field analysis and applied to scatterplots, uses sensitivity coefficients to highlight local variation of

one variable with respect to another [16]. In other work, multivariate data are projected from their attribute space to 2D, such that points with similar attributes are located close to each other [40].

2.4 Parallel Coordinates Plot

Parallel coordinates plots (PCPs) [6], [41], are a well-known visualization technique for exploring multidimensional datasets that display *n* parallel axes, one for each attribute. Every data point is mapped to a vertex on each parallel axis and connected by line segments. In general cases, the ambiguity created by the crossing lines in PCPs associated with many noisy data makes correlation direction and strength, in particular, difficult to interpret. In addition, previous studies have shown that PCPs are slower and less accurate than SCPs for correlation tasks [42].

The advantage of a PCP is that it provides a continuous and comparative view across the axes, and the screen space needed for the visualization scales linearly with the number of attributes. For various reasons, PCPs can become ineffective as the number of data items or attributes becomes large. For large numbers of data items, the overdraw caused by overlapping line segments may obscure important patterns, retaining only outlier visibility [43]. Various modifications to PCPs have been proposed by using color [10], opacity [44], [45], smooth curves [46], [47], frequency [48], density [49], [50], or animation [51] to address overdraw. At the same time, PCPs do not show all possible combinations of attribute pairs, requiring significant user interaction for exhaustive exploration. A PCP matrix [52] is one method that can help overcome this limitation.

2.4.1 Parallel Coordinates Approaches

The majority of these approaches can be placed into one of three categories: *geometry-based*, *frequency-based*, or *density-based* approaches [53]. Unfortunately, most of these techniques have some form of scalability limitation in visualization tasks such as identifying clusters, noise, outliers, etc.

- Geometry-based Approaches: Data items are most often represented as linear splines intersecting each of the axes at their respective coordinates. As lines overlap, they may prevent understanding the data. Smooth and continuous curves can replace the lines for visualizing multiple correlation, facilitating line tracing, reducing overdraw, and visualizing clusters of data [46]. Some techniques have also used clustering algorithms to identify similar items based on proximity of lines or line density [51].

- Frequency-based Approach: Frequency-based approaches aggregate and filter data in a binning process [44]. Frequency-based PCPs avoid overdraw but still suffer from limitations in identifying the principal trend in the data or interpreting mixed trends in the data. In the angular histogram, each polyline axis intersection is considered a vector. It visualizes the magnitude and direction of these vectors. This method helps users explore clustering, linear relationship identification, and outlier detection in data, while avoiding the overdraw problem of classic PCPs. However, angular histogram PCPs still have limitations in identifying nonlinear relationships and finding the crossing locations of data. Furthermore, angular histograms aggregate the frequency of the lines between pairs of axes, which means users can identify only the principal trend of data and will have a difficult time interpreting mixed trends within the data.

- Density-based Approach: Density-based approaches visualize a continuous density function of underlying data instead of discrete samples [54]. Density-based approaches avoid overdraw by replacing opaque lines with a density representation [55]. For example, distance-based weighting constructs a multiattribute density function [49]. Anisotropic diffusion of noise textures [56] has been employed to visualize line orientations. These approaches avoid overdraw; however, they lack a good mechanism to map patterns found using the approaches back to the original data items since they remove individual lines as in geometry-based PCPs.

Interaction is important to explore data efficiently in PCPs. The order of axes defines which attributes are compared. Drag-and-drop axis swapping is commonly used to allow multiple comparisons. Brushing allows users to select a subset of data for highlighting, labeling, replacing, etc. This technique was originally used in scatterplots, but it has been applied to PCPs, for example in angular brushing [57]. Extending brushing to multiple axes can construct multidimensional brushes [58]. Brushing an item on an axis is equivalent to the selection of a line in the Cartesian domain. Line-based and polygon-based brushes can be employed in the spaces between axes. Brushing can be used to select data items in PCPs based on the slopes of lines between axes. For large data, brushing techniques have used wavelets [59] and clustering [60].

2.4.2 Point Line Duality in PCPs

A well-known but not fully exploited quality of PCPs is point/line duality—namely, the property that a point in Cartesian coordinates maps to a line in parallel coordinates. However, less well known

is that a line in Cartesian coordinates maps to a single point in parallel coordinates.

Given a line in Cartesian coordinates specified by a point (x_0, y_0) and a direction specified by $< \tilde{u}, \tilde{v} >$, a point (x_1, y_1) can be found as $(x_0 + \tilde{u}, y_0 + \tilde{v})$. The points (x_0, y_0) and (x_1, y_1) can then be transformed into lines in parallel coordinates as seen in Fig. 2.1.

The intersection point (q, r) can be found by representing the lines parametrically, where $r = x_0 + (y_0 - x_0) \cdot q$ and $r = x_1 + (y_1 - x_1) \cdot q$, and solving

$$q(\tilde{u}, \tilde{v}) = \frac{\tilde{u}}{\tilde{u} - \tilde{v}}. \tag{2.4}$$

$$r(\tilde{u}, \tilde{v}) = x_0 + (y_0 - x_0) \frac{\tilde{u}}{\tilde{u} - \tilde{v}}. \tag{2.5}$$

2.5 Multidependencies Visualization Methods

Many methods use correlation coefficients to calculate the relationship between variables in data and visualize the information. Gosink [61] presents a method that increases the utility of query-driven techniques by visually conveying statistical information about the trends that exist between variables in a query. In this method, correlation fields, created between pairs of variables, are used with the cumulative distribution functions of variables expressed in a user's query. Qu [47] used the correlation coefficient to calculate the strengths between different data attributes in weather data analysis and visualization. Glatter [62] used two-bit correlation to study temporal patterns in large multivariate data. Sukharev [63] proposed a method based on analyzing pairwise correlation in time-varying multivariate data by using pointwise correlation coefficients and canonical correlation analysis. Another pairwise correlation visualization approach used local anisotropic correlation structures in the vicinity of uncertain isosurfaces and used glyphs to visualize these dependencies [64]. Jen introduced a design for exploring correlations between two scalar fields [65]. Some methods have used data mining techniques to gain insight. Gu and Wang presented three hierarchical clustering methods based on quality threshold, k means, and random walks to investigate the correlations with varying levels of detail [66].

Several approaches deal with large and complex correlation fields. The multifield graph is used to give an overview of how multiple fields correlate and to show the strength of their correlation [67]. The approach in [67] is the computation of correlation fields, which are scalar fields containing the local correlations of subsets of the multiple fields. Chen also introduced a sampling scheme to sum-

marize the correlation connection in time-varying multivariate datasets [18]. This scheme consists of three steps: selecting important samples from the volume, prioritizing distance computation for sample pairs, and approximating volume-based correlations. This sample-based approach enables users to obtain an approximate correlation coefficient in a cost-effective manner, making it scalable for large datasets.

UnTangle Map [22] is an effective way to investigate the relationship between data items and their probabilistic labels, as well as the relationships among labels as shown in Fig. 2.2. Its design extends the traditional ternary plot into an interactive mesh of triangles in order to effectively show item label relationships, and to enable the scattering patterns of items to aggregate into a visual summary of the underlying labels. However, this method cannot represent all possible relationships in multidimensional data.

When users have questions related to the whole picture of all possible multiway relationships, they need to use some algorithm to filter the data and feed it to their triangular mesh layout. UnTangle Map is a suitable visualization tool to identify multiway dependencies when users do not need to see all possible multiway relationships in data. A new layout that helps users answer this type of question and quickly implement this task is necessary to improve the capability of addressing more information seeking.

In all cases, the designs of these techniques are not optimal for correlation identification and have limitations in representing multiway dependencies in multivariate data. Therefore, we propose four visualization techniques to help users identify multiway dependencies efficiently for complex and large multivariate data: CCPs to represent detail levels of relationships, snowflake visualization, data scalable parallel coordinates (DSPCP), and multiway dependencies visualization (MultiDepViz) for large and multidimensional data.

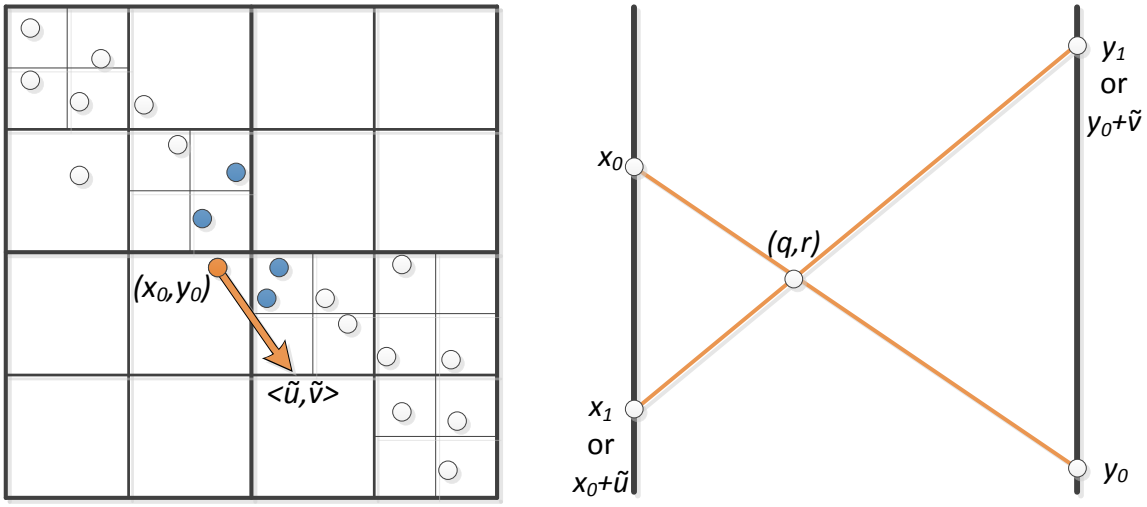


Figure 2.1: Demonstration of the point/line duality property of Cartesian coordinates (left) and parallel coordinates (right).

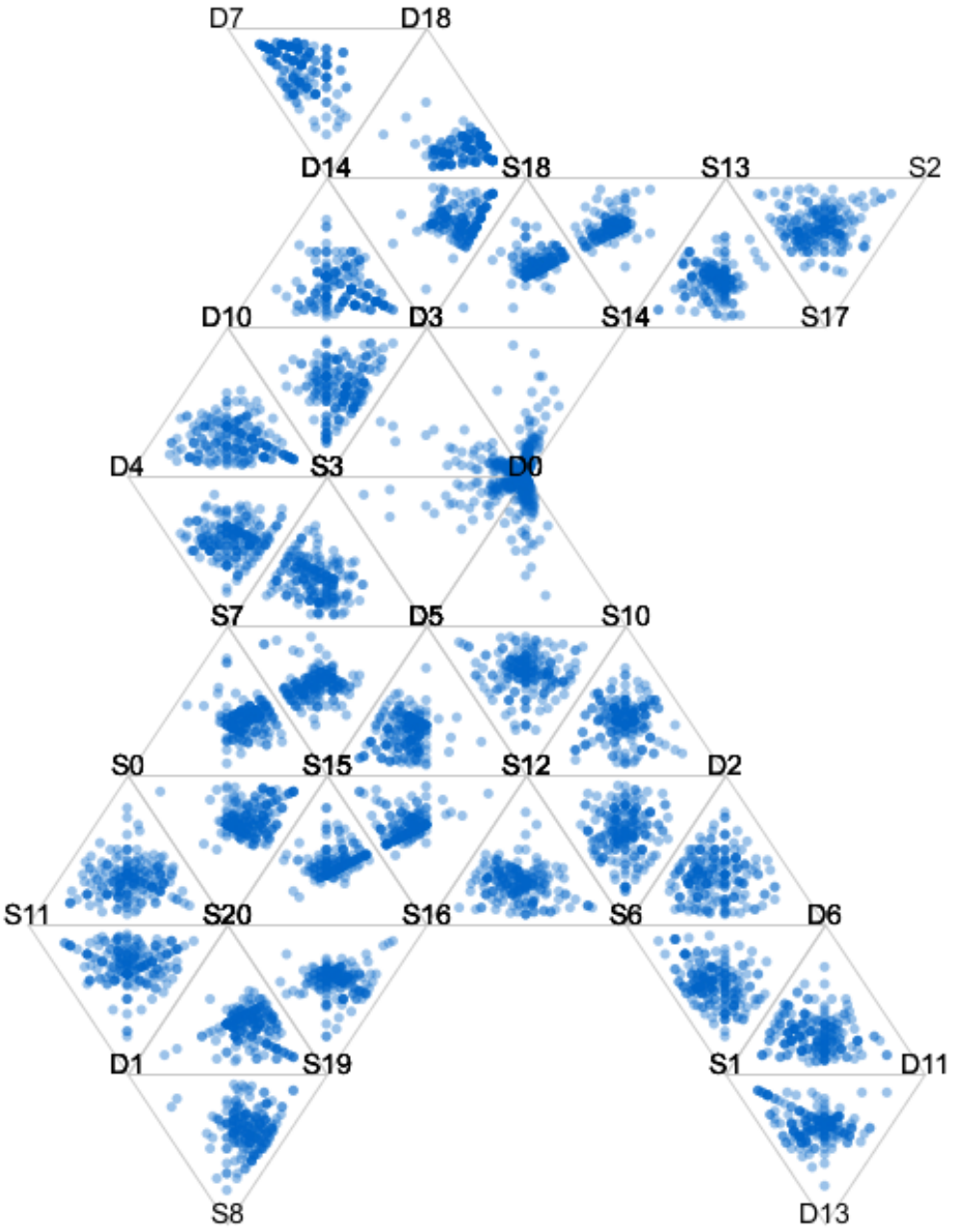


Figure 2.2: UnTangle Map for marketing data.

CHAPTER 3

CORRELATION VISUALIZATION

3.1 Challenges

Task generality (i.e., the support for many tasks) is both an advantage and disadvantage for the SCP and PCP. Either method is capable of being used for correlation tasks, but they are not necessarily the most efficient methods available. This inefficiency has led us to develop a new visual encoding focused specifically on correlation tasks, called correlation coordinate plots (CCPs) as shown in Fig. 3.1. The proposed method is centered on helping users quickly identify the existence, direction, and strength of pairwise correlations. The visual design is motivated by our desire to make the correlation task one of comparison using a position along a common baseline.

For clarity in notation, we assume a dataset X contains n attributes and m data points, with X_i indicating a single data attribute of m values and X_{ij} indicating data point j of attribute i.

3.2 Coordinate System

We propose using a correlation coordinate system that differs from the Cartesian coordinate system, so as to highlight how well points adhere to the correlation. The coordinate system can be seen as a one-dimension parameterization of the data to an underlying model, in this case a line. The vertical position of a data point is the parameterization of the data. The position horizontally is more important, demonstrating the quality of the fit. Therefore, identifying correlation primarily relies on visibility of points to the left and right of the axis.

Transforming the data from a Cartesian domain into the correlation coordinate system is a two-step process laid out in Fig. 3.2, with the top panel showing the positive relationships and the bottom panel demonstrating the negative relationships.

The first step is a scaling operation (Scl) that forces the data into a square region (see Fig. 3.2 panels 1 and 2). The second step is the projection (P_{major} and P_{minor}) operation, which measures the location of the point relative to the positive correlation diagonal (lower left to upper right) as shown in Fig. 3.2a or negative correlation diagonal (upper left to lower right) as shown in Fig. 3.2b. That

measure is used to place the points in the CCP (see panels 3 and 4).

The process begins by normalizing the data to $[-1,1]$.

$$Scl(X_i) = \frac{X_i - \arg\min_{X_i} X_{ij}}{\arg\max_{X_i} X_{ij} - \arg\min_{X_i} X_{ij}}. \tag{3.1}$$

Once normalized, the location of a point i from attributes j and k can be determined. The location on the major (vertical) axis is

$$P_{major}(X_{ji}, X_{ki}) = X_{ki}. \tag{3.2}$$

The position on the minor axis is

$$P_{minor}(X_{ji}, X_{ki}) = \begin{cases} \alpha \cdot (X_{ji} - X_{ki}) & \text{possitive or no correlation} \\ \alpha \cdot (X_{ji} + X_{ki}) & \text{negative correlation.} \end{cases} \tag{3.3}$$

The variable α is a scalar that affects the spread of data points when plotting. We selected a constant value based upon the width of the CCP.

The plot orientation was initially chosen to be vertical in order to pack many plots side by side on the display. Ultimately, the choice of a vertical plot is somewhat arbitrary and will be relaxed in forthcoming sections. Nevertheless, we present and evaluate our approach based upon the vertical orientation.

3.3 Coordinate Axis

We designed the coordinate axis to serves as a visual indicator of the existence and direction of correlation. For two attributes of a dataset, X_i and X_j, PCC is used to indicate a positive correlation by $\rho(X_i, X_j) > \varepsilon$, a negative correlation by $\rho(X_i, X_j) < -\varepsilon$, and uncorrelated by all other values. The major coordinate axis is laid out vertically and represented by a triangle whose base is at the top for positive correlation (Fig. 3.1a), the bottom for negative correlation (Fig. 3.1c), and a straight line for uncorrelated (Fig. 3.1b) data.

We have also considered mapping PCC to the width of the axis, where higher values are wider and lower values thinner. Due to the relatively small width of the axis, we decided this mapping was not particularly informative. Instead, to identify the strength of correlation, users should investigate the distribution of data in the correlation coordinate system, presented in the following sections.

3.4 Coloring Data Points

A number of figures have had their data points colored based upon their PCC value $[\{-1 : blue\}, \{0 : black\}, \{1 : red\}]$. Strictly speaking, this encoding is redundant and not required. However, if colors are interpolated based upon PCC value, they do carry some additional information, and in general, we find them more aesthetically pleasing. Because our focus is on the use of the coordinate axis and coordinate system, our method does not rely on color, and color was *not* used in the user study.

3.5 Correlation Identification

Using CCPs for correlation tasks is fairly simple. Depending upon the users's goal, we suggest

- First, use the axis to determine if the data are, positive, negative, or uncorrelated.
- Next, use the shape of the data points to determine the basic relationship between the attributes (i.e., linear, nonlinear, etc.).
- Finally, the distance of the points from the axis can be used to estimate the strength of correlation, with small distances indicating high correlation, and other conditions such as outliers, noise, etc.

For example, in Fig. 3.1c, by checking the axis, a negative correlation can be seen. By observing the closeness of the data points to the axis, CCP shows a strong linear relationship with a small amount of noise. On the other hand, in Fig. 3.1a, the axis indicates a positive correlation. From the shape of the data, it is apparent that a nonlinear relationship exists with weak linear correlation properties.

3.6 Implementation

Algorithm 1 contains pseudocode for the CCP Visualization. We have also included a sample visualization tool[1] that can be built in Processing. This is pseudocode to draw the CCP of two input data attributes with M items as we described in the above sections.

---

[1] CCPs: https://github.com/hoa84/CCPs_SnowflakeViz

Algorithm 1: DRAW CORRELATION COORDINATE PLOT

```
1:
2:  // Draw axis
3:  if PCC(X,Y) > ε then
4:      drawAxis(upper-triangle-axis)
5:  else if PCC(X,Y) < ε then
6:      drawAxis(lower-triangle-axis)
7:  else
8:      drawAxis(straight-line-axis)
9:  end if
10:
11:
12: // Draw items
13: for i = 1 : M do
14:     [x_n, y_n] := normalize(X_i, Y_i)
15:     p_major := x_n
16:     if PCC(n−1, i) > ε then
17:         p_minor := (y_n − x_n) / 2.0f
18:     else
19:         p_minor := (y_n + x_n) / 2.0f
20:     end if
21:     drawPoint(p_major, p_minor)
22: end for
23:
```

3.7 Evaluation

3.7.1 Performance

The software was built using C++, OpenGL with Qt and Processing, and run on a MacBook Pro with a 2.5 GHz Intel Core i5, 4 GB RAM, and 512 MB Intel HD Graphics 4000. We used the threshold of Pearson correlation coefficient from -0.02 to 0.02 for the low correlation for all of our experiments. The performance comparison of our method, CCP, with SCP and PCP is provided in the following user study. The requires precomputation of the CCP is $1.09x$ time greater than that of the SCP or PCP, since the CCP computes the Pearson correlation coefficient. The rendering time for the CCP is comparable to the rendering time of the SCP and PCP.

3.7.2 User Study Setup

To further evaluate our visualization methods, we conducted a user study comparing CCP with SCP and PCP. In this study, we performed three experiments that asked subjects to perform typical correlation-related tasks.

We invited 25 participants to take part in our study, 9 female and 16 male, all graduate students from a variety of science and engineering fields. Their ages ranged from 23 to 35 years old. We asked the subjects to self-report their level of familiarity with visualization—3 reported themselves as experts, 9 reported themselves as familiar, and 13 reported themselves as not familiar.

In each experiment, subjects started with a short set of slides and/or video to introduce the necessary background. Subjects were then given practice questions where, after answering, the correct answers were provided. Finally, they performed the experimental tasks. For each test, the

subjects' answers and response times were recorded. Following the experiment, subjects completed a short survey. In total, the study lasted less than one hour, including training and testing. For all visualizations, the color gray was used for axes and labels, and black was used to present data points, as shown in Fig. 3.3.

In this user study, we used a particle physics dataset containing 41 output attributes and 4000 data items per attribute. The data represent a parameter space search of 25 input attributes generated by a series of tools that simulate the theoretical physical properties of subatomic particles under the supersymmetic extension of the standard model of particle physics.

The independent and dependent variables used in each experiment can be found in Table 3.1. We used a mixed experimental design using t-testing to calculate t-value, p-value, mean difference, and 95% confidence interval to confirm our hypotheses. Only mean value and p-value are reported, but other data can be provided upon request.

3.7.3 Experiment 1 - Speed and Accuracy in Pairwise Correlation

Since the CCP was designed specifically for the task of quickly and accurately identifying pairwise correlations, our first experiment focused on this.

When looking at SCP and PCP, two challenges persist. First, it can be confusing to determine positive versus negative correlations. Granted, for experts, this is a trivial task, but for others, it can be confusing.

In many ways, the identification of correlation direction is easier with PCP than SCP—parallel lines positive and crossing lines negative. Second, there is some ambiguity when trying to identify the strength of correlation between two attributes. Ambiguity is a much larger problem for PCP. When the relationship is noisy or nonlinear, overlapping lines quickly obscure detail.

When comparing CCP with these other methods, CCP: 1) provides simple visual cues making identification of the direction of correlation fairly trivial; 2) and reduces (not eliminates) the ambiguity by concentrating on correlation in the formulation of the coordinate system. Given these factors, we developed two hypotheses as follows:

H1 | **H2**: *Using a Correlation Coordinates Plot will enable more accurate and faster identification in direction and strength of correlation between two attributes than a [H1: Scatterplot | H2: Parallel Coordinates Plot].*

3.7.3.1 Method

The experiment is summarized in Table 3.1 (**H1** & **H2**). For a block of trials, we showed a participant a plot between two random attributes using either the SCP, PCP, or CCP method (see Fig. 3.3) and asked a forced choice question. Subject accuracy and time were measured.

At the start of the experiment, participants were given an introduction to correlation; instructions on finding correlation in SCP, PCP, and CCP; and six training questions. Participants were then given 21 experimental questions (seven for each plot type, rotating between type).

3.7.3.2 Results and Discussion

The results of both the measured speed and accuracy of our experiments are shown in Fig. 3.4.

The results from Fig. 3.4a show that when comparing accuracy, CCP showed improvement over SCP on average 91% compared to 69%, with statistical significance ($p = 0.001$). We also looked at subjects' performance in identifying only the direction of the correlation, where CCP had an accuracy of 99% compared to 79% for SCP, though not quite with statistical significance ($p = 0.06$). The response times (Fig. 3.4b) showed similar results with CCP responses averaging $11.71s$ compared to $23.4s$ for SCP ($p = 0.001$). Given that in our experiments, CCP outperformed SCP in both speed and accuracy, we consider hypothesis **H1** confirmed.

A similar analysis shows that the accuracy of CCP was 91% compared to 48% for PCP ($p = 0.001$). For identifying type only, CCP had an accuracy of 99% compared to 76% for PCP, though not with statistical significance ($p = 0.09$). The response times (Fig. 3.4b) showed a similar result with CCP coming in on average $11.71s$ compared to $24.5s$ for PCP ($p < 0.001$). Given that CCP outperformed PCP in speed and accuracy, we consider hypothesis **H2** confirmed.

Although not explicitly selected as a hypothesis, we are also able to compare the performance of SCP and PCP. The results showed that SCP had a higher overall accuracy on average, 70%, compared to 58% for PCP ($p = 0.048$). However, when looking at type accuracy only, SCP had no statistical significance in average accuracy of 76% compared to 85% for PCP ($p = 0.25$). The results showed no statistical significance in average response times of SCP and PCP of $24.54s$ and $25.78s$ ($p = 0.774$), respectively. This accuracy confirms prior work [42]. However, for response time, our sample size was insufficient.

The results of experiment 1 confirmed the hypotheses **H1** and **H2**, indicating that using CCP subjects can identify correlation in less time and with higher accuracy compared to SCP and PCP.

In our informal discussions with subjects after the experiment, they indicated that the shape of the axis and the distribution of points in CCP greatly assisted their comprehension of the correlation. Subjects complained that with both SCP and, in particular, PCP, it was more difficult to distinguish positive and negative correlations in scenarios with low correlation. However, they found using CCP enabled them to easily recognize both the direction and strength.

3.7.4 Experiment 2 - Differentiating Linear, Nonlinear, and Uncorrelated

Identifying nonlinear relationships between attributes can also be an important task. When comparing CCP with other methods, CCP provides simple visual cues making identification of correlation direction easier. Beyond that, CCP and SCP give similar visual cues, (i.e., the tasks performed are basically the same) for the shape of the relationship, linear or nonlinear. This motivates our next hypothesis:

H3: *Using a Correlation Coordinates Plot and a Scatterplot will result in similar accuracy and speed for identification of linear, nonlinear, and uncorrelated relationships in two attributes.*

For PCP, identifying these relationships is far more challenging. The overdraw ambiguity that plagues linear correlations becomes significantly worse as even more lines overlap each other in nonlinear cases, which will slow and confuse users. This problem leads to our next hypothesis:

H4: *Using a Correlation Coordinates Plot will result in more accurate and faster identification of linear, nonlinear, uncorrelated relationships in two attributes than a Parallel Coordinates Plot.*

3.7.4.1 Method

The experiment is summarized in Table 3.1 (**H3** and **H4**). At the start of the experiment, participants were given instructions on linear and nonlinear correlation. Participants were then given three training questions followed by nine experimental questions (three for each plot type, rotating between types). For each question, participants saw a plot from two random attributes and were asked a forced choice question. Subject accuracy and time were measured.

3.7.4.2 Results and Discussion

The results of the measured speed and accuracy of our experiments are shown in Fig. 3.5, with all differences showing statistical significance ($p < 0.005$).

The results of our experiment showed that CCP outperformed SCP. Our hypothesis **H3**, however, had predicted that the performance of CCP and SCP would be identical. This result leads us to reject

H3. In our discussions with subjects after the experiment, they indicated that the shape of the axis and the distribution of points in SCP were more difficult to distinguish and that CCP assisted their comprehension of these specific types of correlation.

Due to CCP substantially outperforming PCP in both speed and accuracy, we consider hypothesis **H4** confirmed. As anticipated, participants complained that the overdraw problems made it difficult to differentiate linear versus nonlinear correlations in PCP.

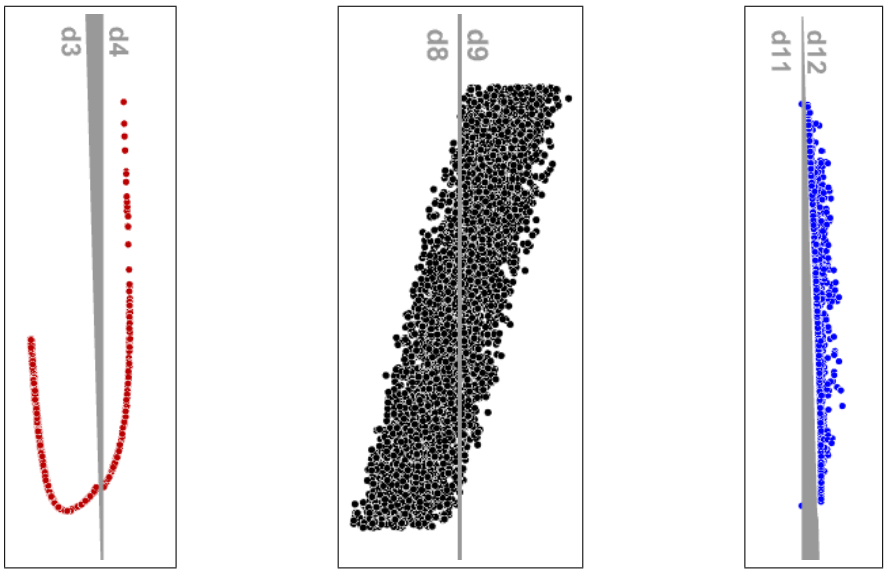
3.8 Discussion

Selecting realistic tasks for a user study is a challenging problem when users are unfamiliar with the data and potentially visualization altogether. We have selected a number of simple tasks, which are building blocks for more complicated data analysis tasks that are commonly performed. The overall outperformance of the CCP over SCP and PCP stands as evidence of its superiority, which should translate to more complex tasks.

Correlation coordinate plots have been developed with the specific task of correlation identification. They have distinct advantages when compared to general task visualizations such as SCP and PCP. The advantages, as confirmed by our user study and real-world datasets, include

- Provision simple visual cues that make identification of the existence and direction of correlation fairly trivial.
- Improved estimation of correlation strength by focusing the coordinate system on model fit.
- Improved identification of linear, nonlinear, and uncorrelated data by reducing ambiguity in the visualization.

We believe that the CCP represents a complementary approach to existing techniques, replacing existing approaches only where correlation is the major feature of focus in data.



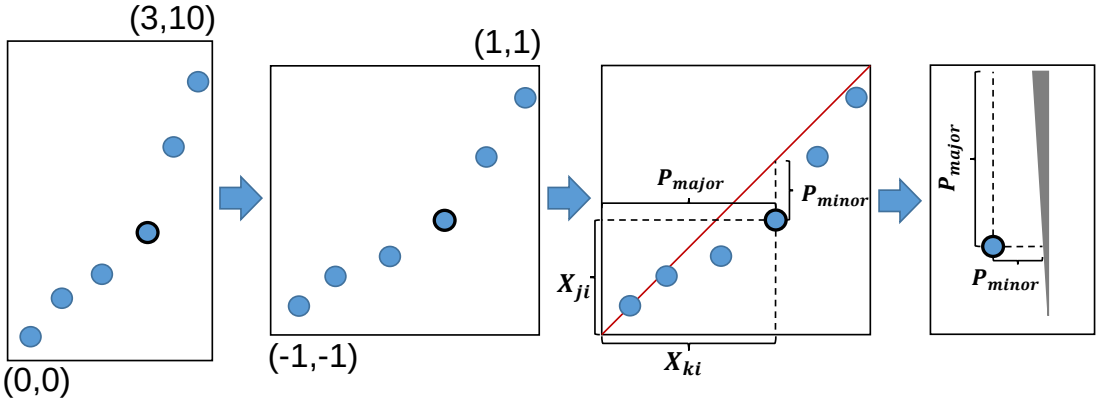
(a) Positive (b) No Correlation (c) Negative

Figure 3.1: correlation coordinate plots (CCPs) transform data into a coordinate system more suited to identifying correlation between two attributes. (a-c): Example CCPs show positive correlation, no correlation, and negative (or anti-) correlation, respectively.

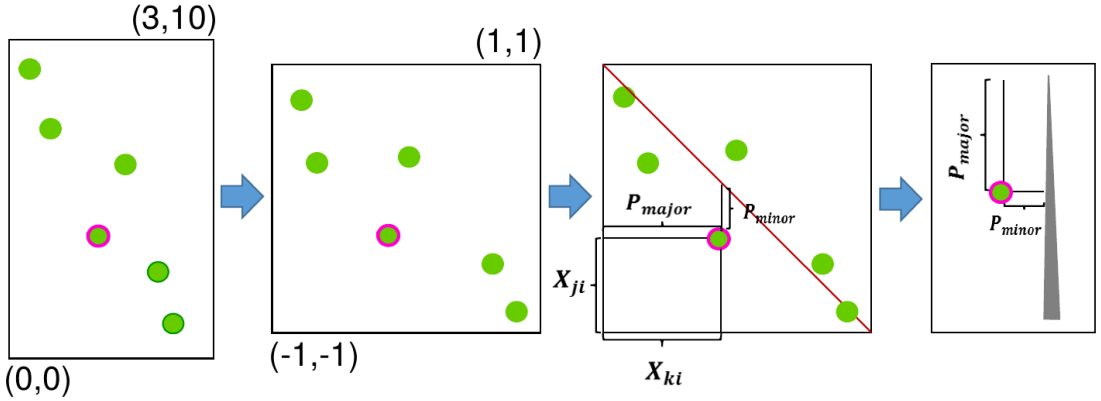
Table 3.1: Variables used to test hypotheses in CCP user study.

Independent Variables	**Potential Values**
Data [**H1** | **H2** | **H3** | **H4**]	2 random attributes from 41 attribute data
Plot [**H1** | **H2** | **H3** | **H4**]	SCP/PCP/CCP
Question [**H1** | **H2**]	How are the 2 attributes correlated?
Question [**H3** | **H4**]	What is the type of correlation?

Dependent Variables	**Potential Values**
Answer [**H1** | **H2**]	High Positive Correlation
	Low Positive Correlation
	No Correlation
	Low Negative Correlation
	High Negative Correlation
Answer [**H3** | **H4**]	Nonlinear Correlation
	Linear Correlation
	No Correlation
Response Time [all **H**]	Time recorded automatically

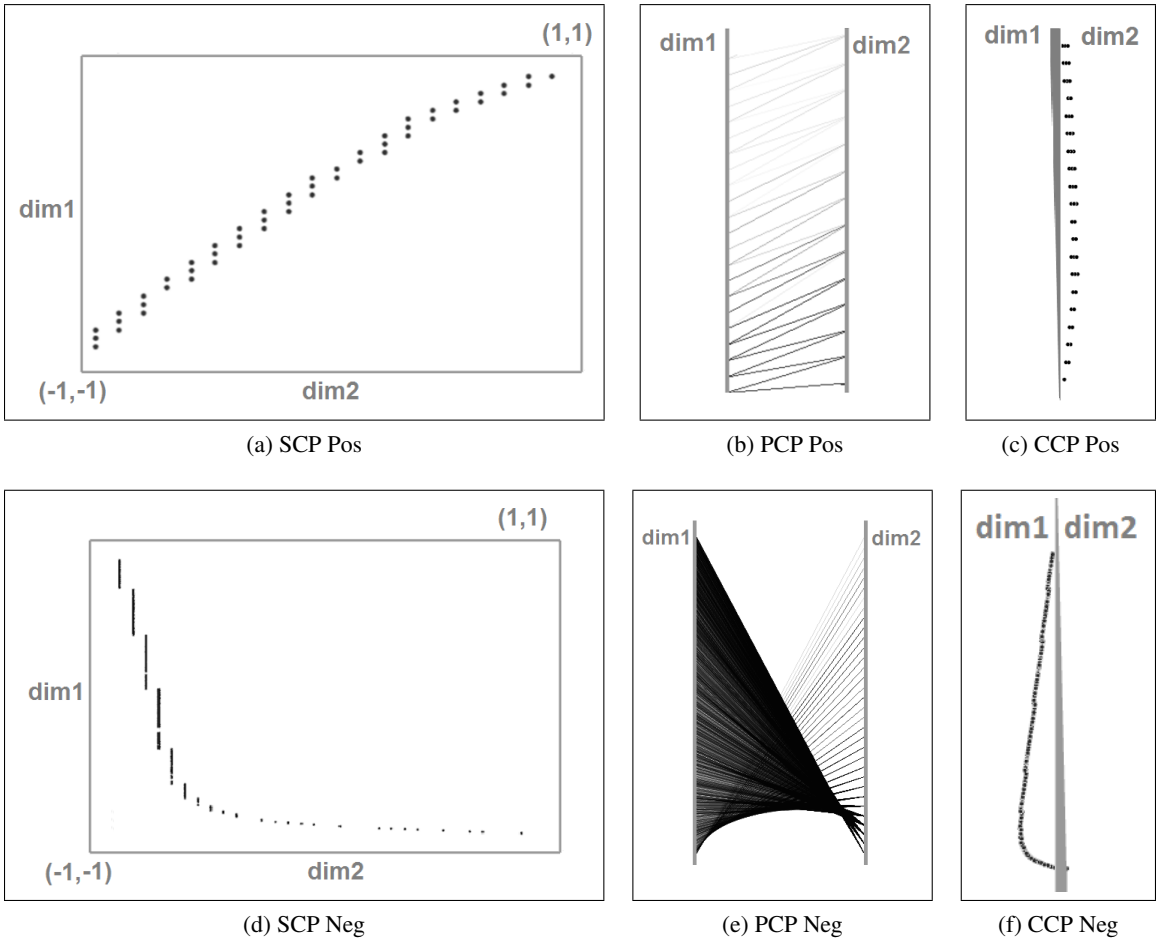


(a) Positive Case



(b) Negative Case

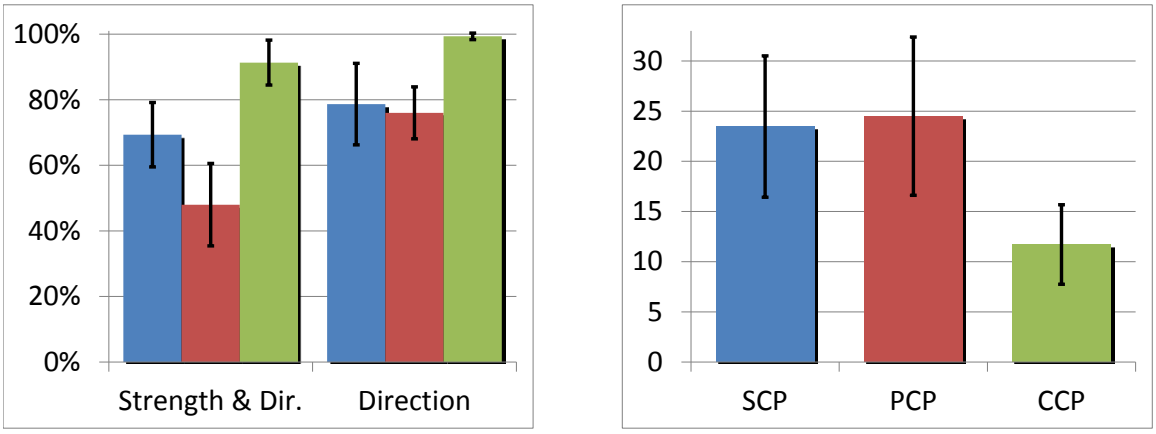
Figure 3.2: Conversion to correlation coordinate system.



(a) SCP Pos (b) PCP Pos (c) CCP Pos

(d) SCP Neg (e) PCP Neg (f) CCP Neg

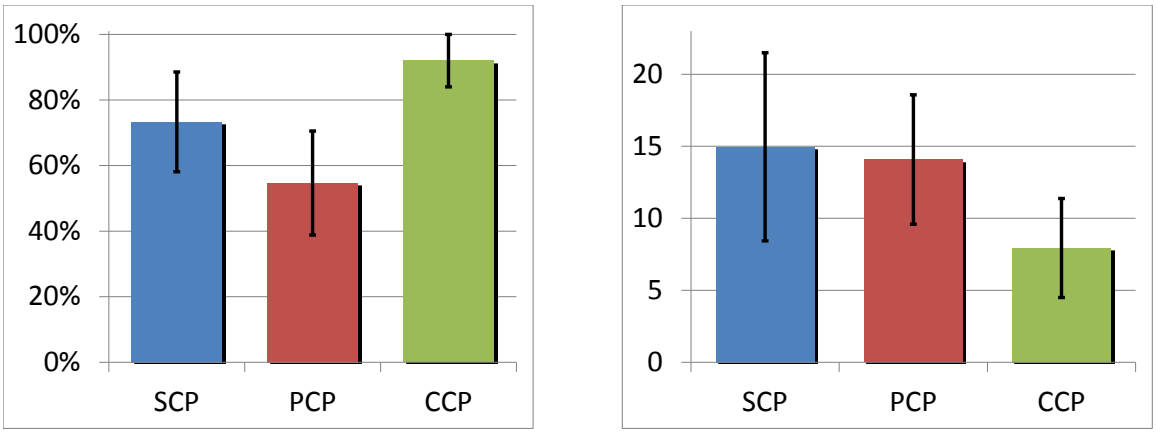
Figure 3.3: Three visualization techniques for comparing two attributes, used in experiment 1 and 2.



(a) Accuracy (%)

(b) Response Time (sec.)

Figure 3.4: Results of experiment 1 show CCP (green, column 3) outperforming SCP (blue, column 1) and PCP (red, column 2) in speed and accuracy. In all figures, error bars indicate standard deviation.



(a) Accuracy (%)

(b) Response Time (sec.)

Figure 3.5: Experiment 2 results show that CCP (green, column 3) outperformed SCP (blue, column 1) and PCP (red, column 2) in speed and accuracy. In all figures, error bars indicate standard deviation.

CHAPTER 4

CORRELATION VISUALIZATION FOR MULTIDIMENSIONAL DATA

4.1 Parallel CCP

Thus far, our approach, CCP, can be used to investigate pairwise correlations. Our next goal was to develop an approach for investigating multiattribute data. We began by looking at SPLOMs, which have the advantage of showing all possible combinations of attributes at the cost of the number of plots needed growing at a rate of $O(n^2)$. This problem may leave little screen space for each individual plot. On the other hand, the number of plots in PCPs grows at a rate of $O(n)$ resulting in more available space for each.

In the PCP spirit, we first applied CCPs to multidimensional data through a series of equally spaced vertical parallel CCP axes as seen in Fig. 4.1. To explore additional combinations of attributes, users can drag an axis to configure the corresponding relationship.

Much like PCP, this approach does not provide immediate access to all attribute pairs, instead of relying on user interaction to fully explore the data. As a compromise between the plot size benefits of PCPs and the comprehensiveness of SPLOMs, we developed a new correlation visualization layout, the snowflake visualization.

While the number of parallel axes grows at a rate of $O(n)$, the number of user interactions to explore all data grows at a rate of $O(n^2)$ (i.e., moving every axis next to every other axis).

4.2 Snowflake Visualization

The snowflake visualization (Fig. 4.2a) is an interactive focus+context style circular design containing two components. The focus region enumerates the correlations between one attribute and all other attributes. This design enables the number of plots in this region to grow at a rate of $O(n)$. The outside is a context region that enumerates all other attribute pairs. The number of plots in this region grows at a rate of $O(n^2)$. Simple user interaction enables swapping the attributes in and out of the focus region for deeper investigation.

4.2.1 Focus View

The focus view (Fig. 4.3a) enables investigating the correlation of one attribute with all other attributes. Given n attributes, there are $(n-1)$ pairs laid out around the center of the circle with equal angular spacing. By default, the final attribute of data is the initial focus attribute. Attributes are initially sorted by ID but can be reordered with other sorting methods. The inner radius (i.e., the start of the CCP axes) is chosen such that none of the data points between CCPs will overlap. The outer radius (i.e., the end of the CCP axes) is adjustable to give more or less space to the context views.

4.2.2 Context View

Given the attributes covered by the focus view, we designed the context view to give complete coverage of the remaining attribute pairs. These context views (Figs. 4.3b and 4.3c) are attached to the branches of the focus view. The organization is based on the parity of n.

When n is odd ($m = (n-1)/2$), the organization, shown in Fig. 4.4, contains all pairwise correlations in data. Each pair, (d_i, d_j), where $i = 0, 1, ..., 2m-1$ and $j = i+1, i+2, ..., 2m$, presents the correlation between two attributes, d_i and d_j. The red box in Fig. 4.4 contains all the attribute pairs that are presented in focus view, pairing the last attribute d_{2m} and all other attributes $(d_0, ..., d_{2m-1})$.

The context view has two groups—the upper branches and lower branches. In the upper branches, where $i = 0, ..., m-1$, the i^{th} correlation coordinate branch presents correlation between attribute d_i and m other attributes that are $(d_{i+1}, d_{i+2}, ..., d_{i+m+1})$. There are m pairwise correlations in each upper branch. We can see one upper branch in Fig. 4.3b that has four pairwise correlations when the number of attributes n is nine and m is four.

In the lower branches, where $i = m, m+1, ..., 2m-1$, the i^{th} correlation coordinate branch presents correlation between attribute d_i and other attributes as shown in the i^{th} column in Fig. 4.4. There are $(m-1)$ attribute pairs in each lower branch. Fig. 4.3c shows one lower branch that presents three pairwise correlations when the number of attributes $n = 9$ and $m = 4$.

The organization is similar when n is even ($m = n/2$). The focus view presents the correlations between last attribute d_{2m-1} and all other attributes in data $(d_0, d_1, ..., d_{2m-2})$. The context view has only a single branch type that has $m-1$ attributes pairs in each upper or lower branch.

Here, only a single type appears with the focus+context views that are combined to form the final visualization. In the case where the number pairings m is large, the context branches can be

split, such as in Fig. 4.2a.

4.2.2.1 Detail View+Interaction

Typically, a single large CCP detail view is also included with the snowflake visualization (a similar practice to SPLOMs). A few interactions are included with the snowflake visualization, including

- *Click-to-swap*: When the user clicks an attribute, it becomes the focus attribute. After swapping, outer attributes are reordered based upon a sorting criterion (by attribute ID).

- *Zooming*: The outer branch of the selected attribute is zoomed out twice to see more correlation of this selected attribute with other attributes on the outer branch.

- *Over-to-detail*: As the mouse moves over a plot, the detail view is updated to that pairing. The user can change the attributes displayed in detail view by mousing over a CCP or by clicking a CCP in the focus+context view.

4.3 Implementation

The following algorithms contain pseudocode for the snowflake visualization. We have also included a sample visualization tool[1] that can be built in Processing. Drawing the snowflake visualization is presented in two parts. The focus view, based on attribute j, can be drawn using algorithm 2 by drawing a series of CCPs plots around the center, with equal angular spacing. Algorithm 3 presents the method to draw context view of snowflake visualization, based on the parity of n.

Algorithm 2: DRAW FOCUS VIEW OF SNOWFLAKE

```
// Draw attributes before focus j
for i = 1 : j − 1 do
    setPosition(cen, rad, (i − 1) · 360°/(n−1))
    drawCCP(A_j, A_i)
end for
// Draw attributes after focus j
for i = j + 1 : N do
    setPosition(cen, rad, i · 360°/(n−1))
    drawCCP(A_j, A_i)
end for
```

[1] CCPs: https://github.com/hoa84/CCPs_SnowflakeViz

Algorithm 3: DRAW CONTEXT VIEW OF SNOWFLAKE

```
// Parity bit for even vs. odd n
even = (n is even) ? 1 : 0

// Draw attributes after focus j
m = ⌊n/2⌋

// Loop ranges
range_0 := m − even
range_1 := 2m − 1 − even

// Angular separation for plots
b_0 = 180^o/(m − 1 − even)
b_1 = 180^o/(m − 2)
for i = 0 : range_0 do
    for j = 0 to m − 2 do
        setPosition(cen_i, ang_i + b_0 ∗ j)
        drawCCP(A_i, A_{i+j+1})
    end for
end for
for i = range_0 + 1 : range_1 do
    for j=0 to i-m+1-even do
        setPosition(cen_i, ang_i + b_1 ∗ (2m + j − i − 2))
        drawCCP(A_i, A_j)
    end for
    for j=i+1 to 2m-2 do
        setPosition(cen_i, ang_i + b_1 ∗ (j − i − 1))
        drawCCP(A_i, A_j)
    end for
end for
```

4.4 Many-Attribute Correlations

Pairwise correlations are frequently important to understanding data. However, as the number of attributes increases, the desire to explore relationships of multiple attributes simultaneously increases. The snowflake visualization partially addressed the need by presenting many pairwise relationships simultaneously. Comparing three or more attributes requires looking at an exponentially increasing number of plots and mentally fusing the distributions. We can extend CCP design for presenting certain types of multidimensional relationships.

To do this, we slightly modify visual metaphors of the CCP. First of all, we remove the positive/negative metaphor encoded via the axis because multidimensional relationships tend to not have a directional measure, only magnitude. Now, the parameterization model can be relaxed to any invertible function, $[s,t] = g(\bar{x})$. The vertical axis still represents a 1D parameterization of the data, s. The horizontal axis can now represent a secondary model parameterization, t. Finally, we represent information lost in this encoding via a series of partially transparent boxes, one per data point, that form a "haze" surrounding the data points. The size of the boxes is computed by the residual, $r = ||\bar{x} - g^{-1}(s,t)||$.

For our experiments, we have used principal component analysis (PCA) to parameterize the data. The PCA could be replaced with any other model that fits our functional definition. Using PCA, we set $g(\bar{x})$ equal to the magnitude of the first two principal components of the data, and

the size of the box is set to the residual. Fig. 4.5 shows two examples. The SPLOM on the left (Fig. 4.5a) shows all of the attributes of the dataset. Two subsets have been selected in red and blue. The red subset are attributes that all appear pairwise linear. When we use the many-attribute CCP (Fig. 4.5b), we can see that all of the attributes are linear with respect to one another. On the other hand, the blue attributes appear nonlinear. When visualized with the many-dimensional CCP (Fig. 4.5c), we can see a relatively simple nonlinear 2D pattern within the data.

4.5 Evaluation

4.5.1 Performance

The software was built using C++, OpenGL with Qt and Processing, and run on a MacBook Pro with a 2.5 GHz Intel Core i5, 4 GB RAM, and 512 MB Intel HD Graphics 4000. We used the threshold of Pearson correlation coefficient from -0.02 to 0.02 for the low correlation for all of our experiments. The performance comparison of our method, snowflake visualization, with SPLOM and PCPs is provided in the following user study. The precomputation of snowvlake visualization is 1.12x greater than the SPLOM and PCP, since the snowflake visualization requires the Pearson correlation coefficient. The rendering time of the snowflake visualization is comparable to the SPLOM and PCP.

4.5.2 User Study

To further evaluate our snowflake visualization method, we conducted a user study comparing our approach with SCP matrix and PCP. This user study was conducted in tandem with the user study presented in section 3.7.

The snowflake visualization was designed specifically for the task of quickly and accurately exploring pairwise correlations in multidimensional data as compared with SPLOMs and PCP. As the number of dimensions increases, each SCP within a SPLOM becomes quite small and the number of plots becomes overwhelming. For PCP, as the number of attributes increases, the interaction required for many tasks puts increased pressure on the user to explore features of interest. With these factors in mind, we developed three hypotheses:

H5: *Using a snowflake visualization will enable more accurate and faster identification of correlation between two attributes in multidimensional data than a scatterplot matrix or parallel coordinates plot.*

H6: *Using a snowflake visualization will enable more accurate and faster identification of how*

many attributes are correlated with a chosen attribute in multidimensional data than a scatterplot matrix or parallel coordinates plot.

H7: *Using a snowflake visualization will enable more accurate and faster identification of which attributes are correlated with a chosen attribute in multidimensional data than a scatterplot matrix or parallel coordinates plot.*

4.5.2.1 Method

The experiment is outlined in Table 4.1 (**H5**-**H7**). Each participant was given an introduction and demo video for each visualization method and completed 12 sample questions using data unrelated to experimental trials. Then, each performed 21 experimental questions rotating first between visualization types, then question types.

4.5.2.2 Results and Discussion

The results of measured speed and accuracy in Fig. 4.6 (Type 1) show the snowflake visualization improved accuracy and speed over SPLOMs and PCPs with statistical significance (all $p < 0.05$). The average accuracy for snowflake visualization was 89% compared to 67% for SPLOM and 64% for PCP. The response times (Fig. 4.6b) for snowflake visualization came in at an average of 19.9s compared to 31.3s for SPLOM and 26.1s for PCP. Given that snowflake visualization outperformed SPLOM and PCP in speed and accuracy, we consider hypothesis **H5** confirmed.

Again, the results of the experiments showed that the snowflake visualization improved accuracy and speed over SPLOMs and PCPs (see Fig. 4.6, Type 2) with statistical significance ($p < 0.05$). Therefore, we consider hypothesis **H6** confirmed.

The results of this final test also showed improved accuracy and speed over SPLOMs and PCPs (see Fig. 4.6, Type 3) with statistical significance ($p < 0.05$), leading us to also consider hypothesis **H7** confirmed.

This user study confirmed the hypotheses **H5**, **H6**, and **H7**, indicating that using the snowflake visualization, users can identify multiple correlations in multiattribute datasets in less time and with higher accuracy than SPLOMs and PCPs. Our discussions with subjects after the experiment indicated that snowflake visualization's focus+context style greatly assisted their interactions and comprehension when working through multiple pairwise correlation questions.

The participants complained that small SCPs made the SPLOM difficult to use, due to the inability to see individual plots and difficulty tracking rows or columns of plots. Using PCP, participants

complained that the number of dragging operations required to explore multiple correlations made it very difficult for them.

4.5.3 Case Study

We applied three visualization methods, including the snowflake visualization, SPLOM, and PCP, to three publicly available datasets including Boston house price data[2], Pollen data[3], and Hurricane Isabel data[4].

4.5.3.1 Boston House Price

Boston housing data (see Fig. 4.7) is multivariate dataset containing 506 items across 14 attributes. The data contain several variables that try to explain variation in home values in the Boston area.

When comparing this dataset in a snowflake visualization and SPLOM, a number of features are observable in both visualizations. For example, in both visualizations, the Age/Rad pairing is fairly clearly a case for segmentation into two data groups. In the SPLOM, it will likely take longer.

A big advantage in snowflake visualization is that it makes way for exploiting additional visual channels. Take the Age/Ind pairing. In all visualization approaches, the coloring scheme we have used makes it fairly easy to see that there is a strong positive correlation. However, without the coloring, that might not be the case. If color had been used for some other purpose, classification for example, suddenly we lose the ability in SPLOMs to quickly determine correlation, while observing classification. Since CCPs do not rely on color to communicate correlation, we can encode other information in the color channel without significant loss of correlation information.

4.5.3.2 Pollen Data

The pollen data (Fig. 4.8) contain 3848 items each with 6 attributes. This dataset summarizes geometric features of pollen grains.

The nature of the data makes it difficult to use the PCP due to overdraw. Take the Ridge/Weight and Ridge/Density. Even though we can be fairly certain that Ridge/Weight is more negatively

---

[2]http://lib.stat.cmu.edu/datasets/boston

[3]http://lib.stat.cmu.edu/datasets/pollen.data

[4]http://vis.computer.org/vis2004contest/

correlated than Ridge/Density, any other detail is lost. We are unable to determine if this problem is due to outliers, nonlinearity, noise, etc. Techniques such as clustering, density, and histogram PCPs can be used to further improve the representations. However, for correlation strength tasks, these approaches are not particularly beneficial.

For the snowflake visualization, these data prove to be little trouble. When Ridge is selected as the focus parameter, Density and Weight can be compared in detail. The thinner spread of Ridge/Weight indicates a stronger linear relationship compared to Ridge/Density. In addition, the details available in the view confirm that any weakness in the correlation is due to noise.

4.5.3.3 Hurricane Data

Hurricane Isabel (Fig. 4.9) data are provided as part of the IEEE Visualization 2004 contest. This dataset contains a variety of simulated variables related to Hurricane Isabel, a major Atlantic storm that occurred in September 2003. The Isabel dataset consists of 48 timesteps, each containing measurements of 11 attributes with a spatial resolution of $500 \times 500 \times 100$. We also show only seven of the more "interesting" attributes due to space considerations. Of the original data 25 million data items, we use only 10 million because approximately 15 million data items contain at least one invalid *NaN* field.

With 10 million data items in the Hurricane data, the overdraw in the PCP makes it hard to understand any relationships in the data. For example, the relationship Temp/Pres shows only the bowtie shape, losing the individual data patterns. In many ways, SCPs do a better job than PCPs. The Temp/Press relationship is visible with the SCP. However, clear interpretation is difficult, since as Temp increases, Press first decreases, then increases, and finally decreases.

Our approach presents these relationships more clearly. The direction and strength of relationship between Temp and Pres can be identified in snowflake visualization. The lower triangle shape of axis identifies the negative relationship. Additionally, the data points distribution, mostly being of similar distance to the axis with a few spread out, enables identifying that this relationship is not strongly negative and nonlinear.

4.6 Applications on Code Performance Data

Effective debugging and optimization of large scientific applications takes up half of the overall development time and requires a detailed analysis of their behaviors. Debug logging and perfor-

mance tracing are tools for such analysis tasks and are used in a wide range of applications and systems. It is natively supported by tools such as HPC Toolkit [68], scientific libraries such as SAMRAI [69], and frameworks such as the Fault Tolerance Backplane [70].

However, as these logs grow large, they overwhelm developers' ability to find the key bits of information among the MBs of GBs of aggregate text and numeric data, which makes it critical to develop visualization tools that help developers understand these complex logs and their relationship to application semantics and performance.

CCP and snowflake visualization can be applied for code performance data [71]. These visualizations help developers focus their attention on the key regions of their logs and identify important relationships among different dimensions of collected data. Our visualizations were implemented on top of the Sight application analysis framework.

4.6.1 The Complexity of Program Behaviors

The large size of application logs makes it necessary to organize and present them to developers in a way that enables them to focus in detail on just the regions that are relevant to a given task. To this end, we developed a hierarchical graph representation that supports zooming, expansion, and the collapse of interactions.

4.6.2 Interactive Flow Graph

4.6.2.1 Design

The proposed graph organizes segments of the application logs that share some property (e.g., collected during a given function call) into separate graph nodes. Numerical and text labels on the nodes denote the corresponding application region and arrows denote the flow of control and data among application regions. Repetitive structural patterns of program behavior (e.g., nested loops, recursion, or repeated communication) are shown by nesting subgraphs inside each other.

4.6.2.2 Flow Graph Algorithm

We implemented the above visualization and connected it to the logs exported by the Sight tool. Sight logs explicitly encode hierarchical containment relationships of log regions, which makes it easy to connect log regions to elements in our hierarchical visualization. By connecting the visual elements to the log regions, they encode our visualization and make it possible to interactively search through the overall structure of the log, zoom into the log regions of most interest, and

alternate between the high-level hierarchical view and detailed view of individual log entries.

4.6.2.3 Interactions

Developers can zoom out to see an overview of the graph or zoom in for a more detailed view. Developers can collapse or expand subgraphs to identify the informative graph regions. These interactions enable developers to explore large logs.

4.6.3 Evaluation

We conducted a case study to apply our visualization method to different programs, including a simple Fibonacci sequence program and the AMG2103 benchmark. Fig. 4.10 and Fig. 4.11 show the resulting visualizations, which confirm that developers can easily use our method to explore large program behaviors.

4.6.4 Correlation Coordinate Plots

Fig. 4.12 shows that multidimensional visualizations such as SCP, PCP, and our approach CCP can help application developers understand relationships in multidimensional performance and behavior data.

SCP (Fig. 4.12b) has the advantage of identifying relationships in three dimensions, but it cannot show all combinations of dimensions at the same time. The screen space of PCP (Fig. 4.12c) grows $O(n)$, however, it requires heavy user interaction for complete exploration. SCP (Fig. 4.12b) can help identify relationships in two, three, and four dimensions, but it cannot show all combinations of dimensions at the same time.

PCP's screen space requirements grow as $O(n)$, but it requires heavy user interaction for complete exploration (Fig. 4.12c). These problems led us to develop a new visual encoding called coorelation coordinate plot (CCP).

CCP is laid out on a single major axis, which represents the best linear fit through the data, and a minor axis that indicates the distance from this line. The major coordinate axis also serves as an indicator of the direction of correlation. Data elements are transformed into the correlation coordinate space and displayed accordingly as in Fig. 4.12a.

4.6.5 Snowflake Visualization

We applied snowflake visualization (Fig. 4.12d) to improve code performance tasks. The focus region enumerates the correlations between a given dimension and all other dimensions. The outer region enumerates all other pairs of dimensions to provide further context. User interactions such as swapping, zooming, and selection enable shifting dimensions in and out of the focus region to allow exploration of the entire dataset with $O(n)$ interactions.

4.7 Discussion

4.7.1 Study Task Selection

Selecting realistic tasks for a user study is a challenging problem when users are unfamiliar with the data and potentially visualization altogether. We have selected a number of simple tasks, which are building blocks for more complicated data analysis tasks that are commonly performed. The overall outperformance of the CCP over SCP and PCP stands as evidence of its superiority, which should translate to more complex tasks.

4.7.2 Abstraction Selection

SCP and PCP have served a straw man role in our evaluation. Any number of modifications could be applied to either technique to better inform the user about correlation. However, since there is no single de facto standard, we did not want our evaluation to be clouded by questions of abstraction selection in SCP or PCP. Therefore, we stuck to the basic formulations of each approach. We hope this method spurs the community to dig deeper into this subject and generate a more extensive evaluation of approaches, such as those of Harrison [72] and Kay [73].

4.7.3 Very High Attribute Count Data

For data with large numbers of attributes, we believe that approaches to extract the natural dimensionality of data, such as PCA, in combination with techniques such as CCP, will be critical in analysis. For all practical purposes, beyond 30 or 40 attributes, our approach is no longer viable. However, this is a similar limitation to SPLOMs and PCPs. We consider higher-dimensional cases to still be an open problem.

The snowflake visualization showed significant performance improvements over SPLOMs and PCPs. The snowflake visualization is an efficient focus+context style layout representing a fair compromise between space efficient design, comprehensive visualization, and reduced user interaction

for showing all pairwise correlations in multiattribute data.

We believe that the CCP and snowflake visualization represent complementary approaches to existing techniques, replacing existing approaches only where correlation is the major feature of focus in data. We believe that more of these task specific approaches are on the horizon and will provide data analysts better, faster access to relevant information in their data.

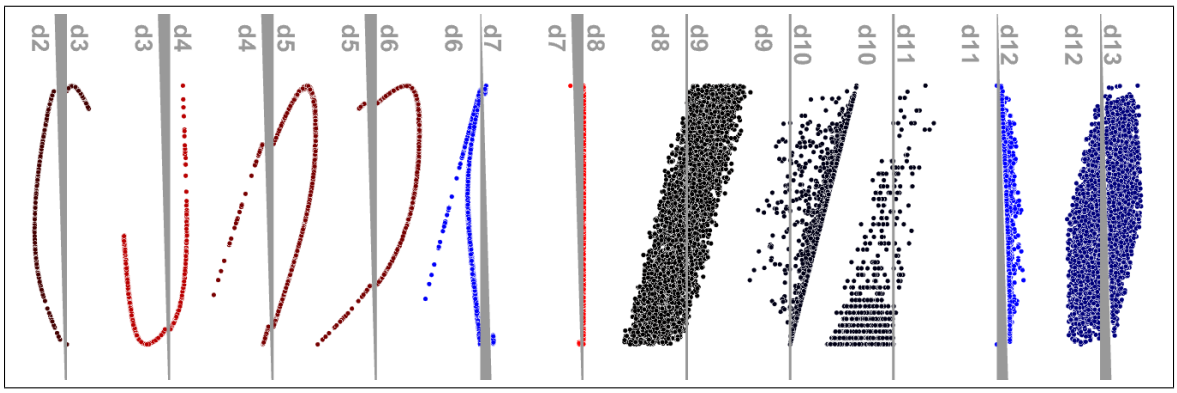
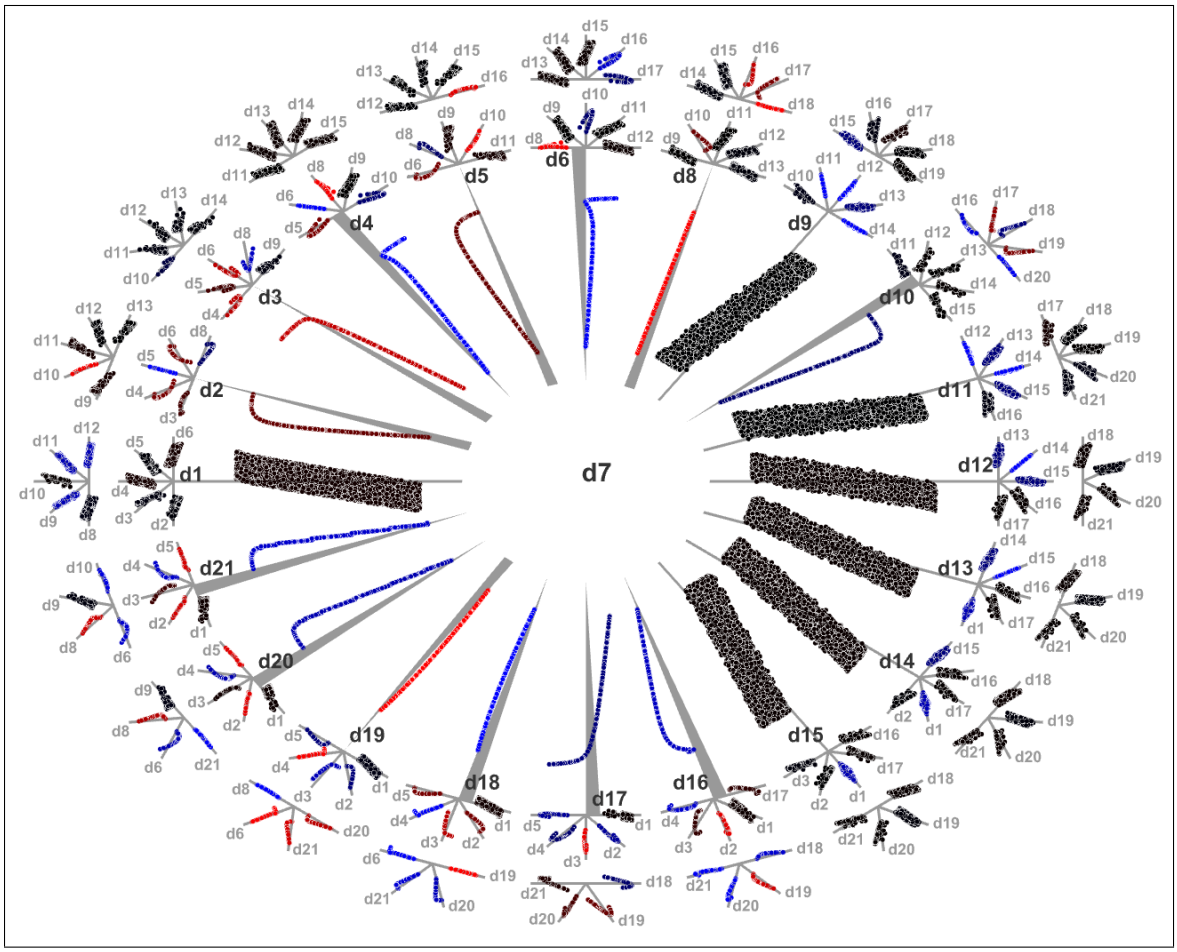
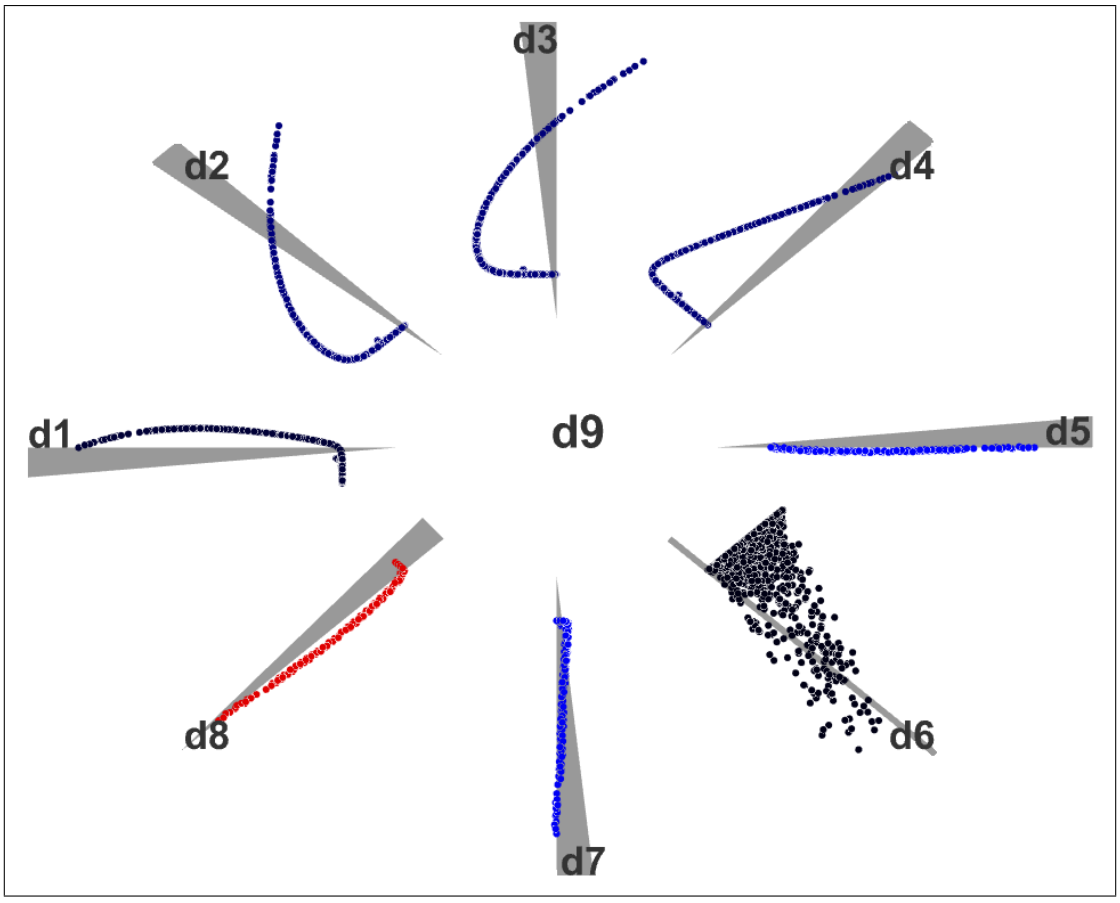


Figure 4.1: Parallel CCP for 10 attributes allows full exploration of the data, but, like PCP, it relies on heavy user interaction.

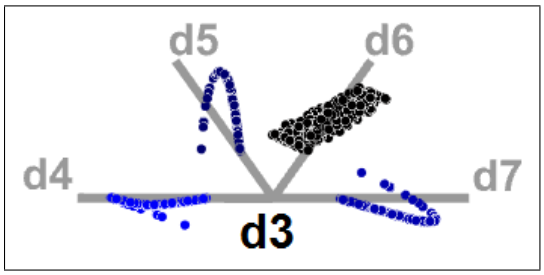


(a) Snowflake visualization

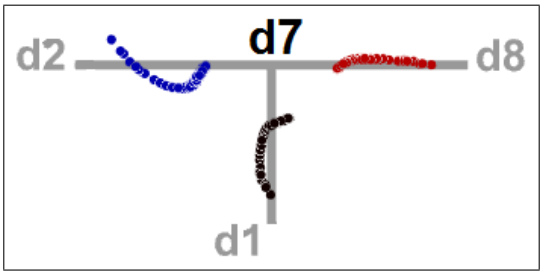
Figure 4.2: The Snowflake Visualization is a focus+context interface that combines CCPs for one attribute to all others in the middle (i.e., the focus) and CCPs for all other attribute pairings on the perimeter (i.e., the context).



(a) Focus view



(b) Upper branch of context view



(c) Lower branch of context view

Figure 4.3: A focus view (a) and multiple context views (b-c) for snowflake visualization.

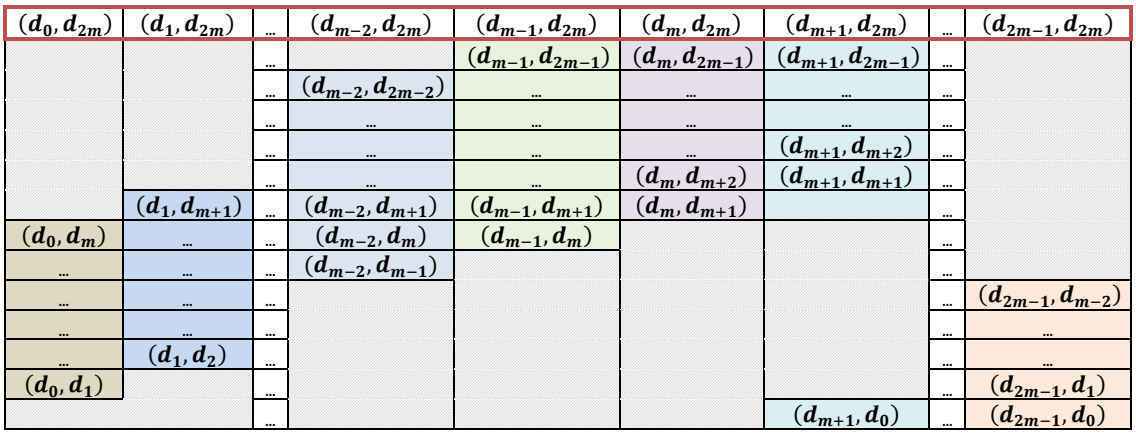
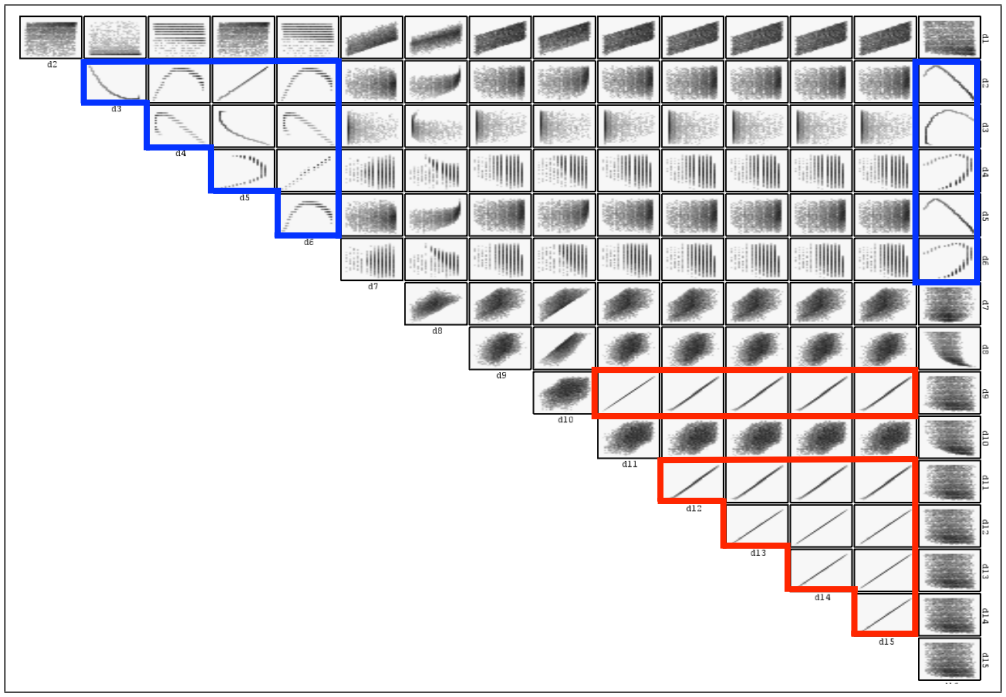


Figure 4.4: Branch view attribute pairing matrix for n attribute data when n is odd, and the focus attribute is d_{2m}. Each row and column represents 1 attribute of the data. Branch pairing for each attribute is found by selecting the column for that attribute and pairing with highlight attributes.

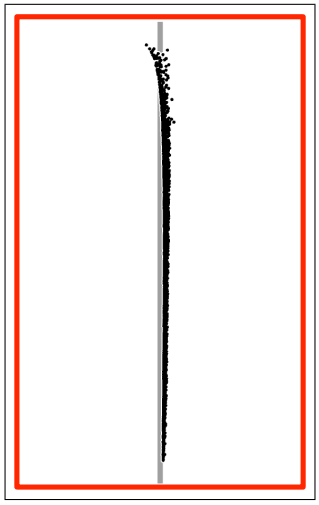
Table 4.1: Variables used to test hypotheses.

Independent Variables	**Potential Values**
Data [**H5** | **H6** | **H7**]	10 or 21 attributes from 41 attribute data
Plot [**H5** | **H6** | **H7**]	SPLOM/PCP/snowflake
Question [**H5**]	How are the two attributes correlated?
Question [**H6**]	How many attributes are correlated to *i*?
Question [**H7**]	Which attributes are correlated to *i*?

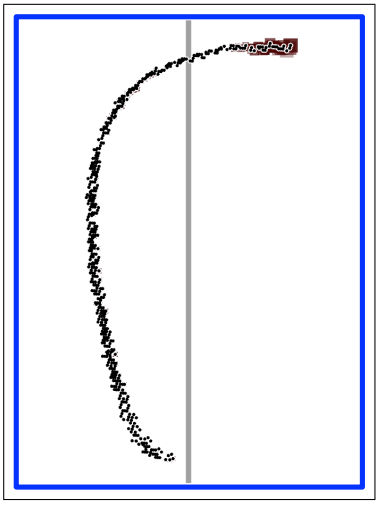
Dependent Variables	**Potential Values**
Answer [**H5**]	High Positive Correlation
	Low Positive Correlation
	No Correlation
	Low Negative Correlation
	High Negative Correlation
nswer [**H6**]	Number of attribute
Answer [**H7**]	List of attribute
Response Time [all **H**]	Time recorded automatically



(a) SPLOM

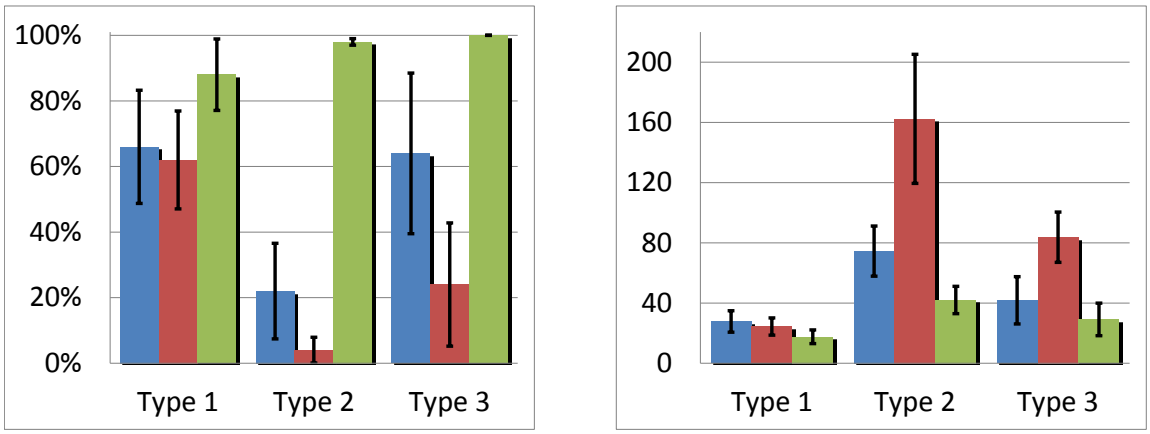


(b) CCP of linear feature



(c) CCP of nonlinear feature

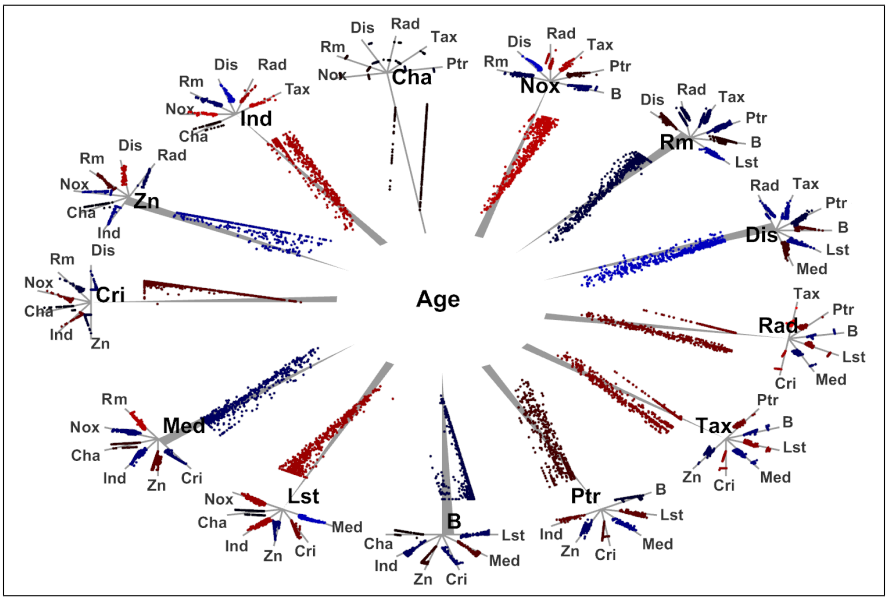
Figure 4.5: CCP for multiple attributes using PCA. (b) The attributes in red are a linear feature. (c) The nonlinear feature in blue is 2D, with the residual visible in the red haze.



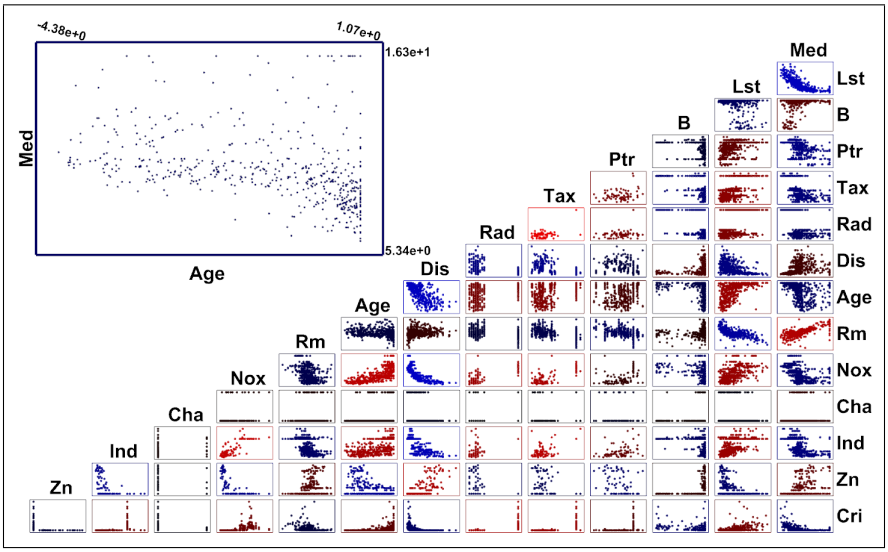
(a) Accuracy (%)

(b) Response Time (sec.)

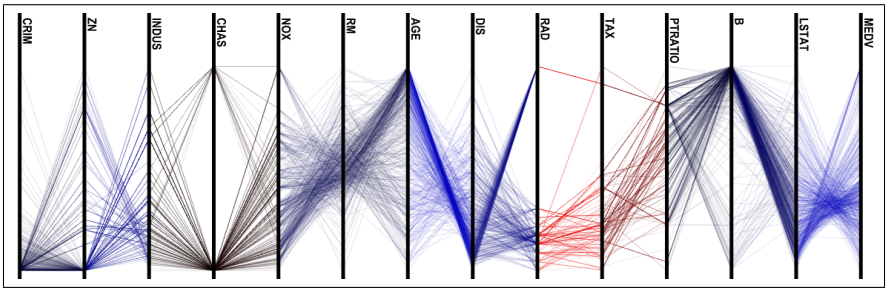
Figure 4.6: Exp. 3 results show CCP (green, col. 3) outperformed SCP (blue, col. 1) and PCP (red, col. 2).



(a) Snowflake visualization

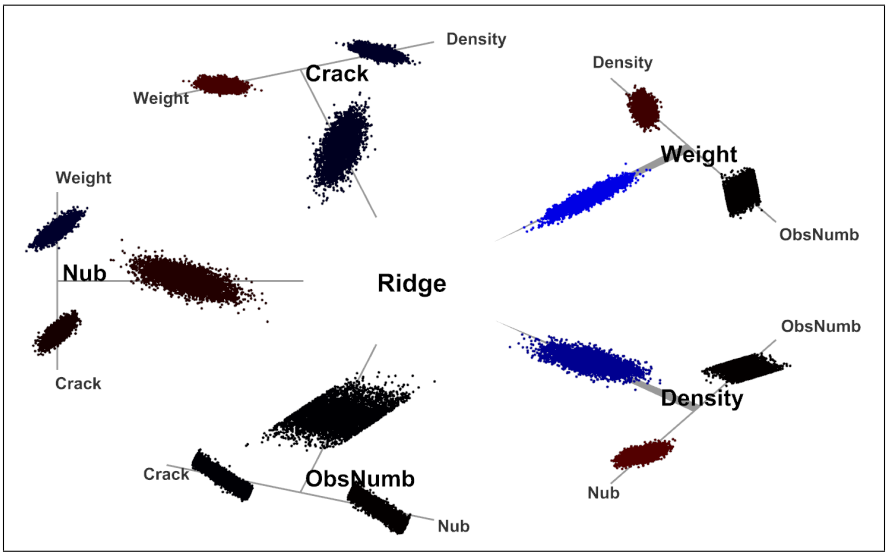


(b) Scatterplot matrix

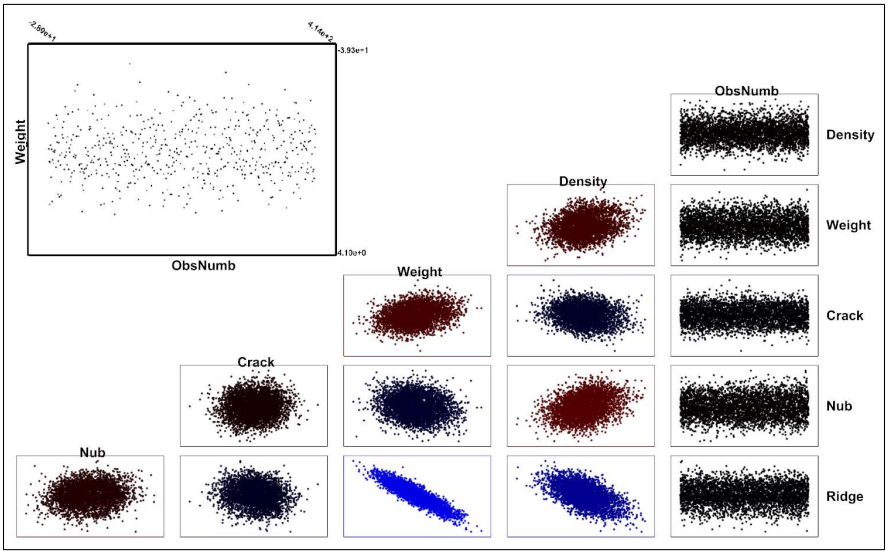


(c) Parallel Coordinates

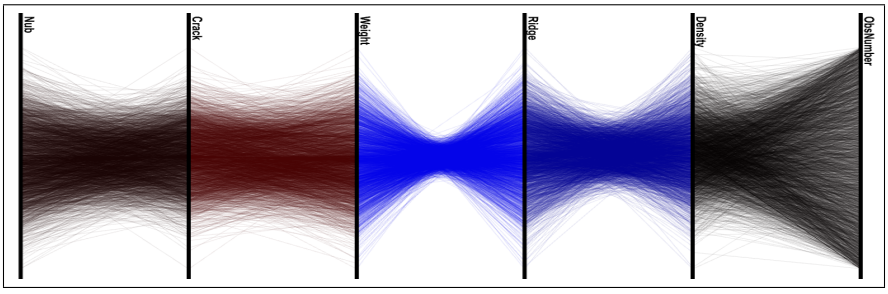
Figure 4.7: Visualizations for Boston House data.



(a) Snowflake visualization

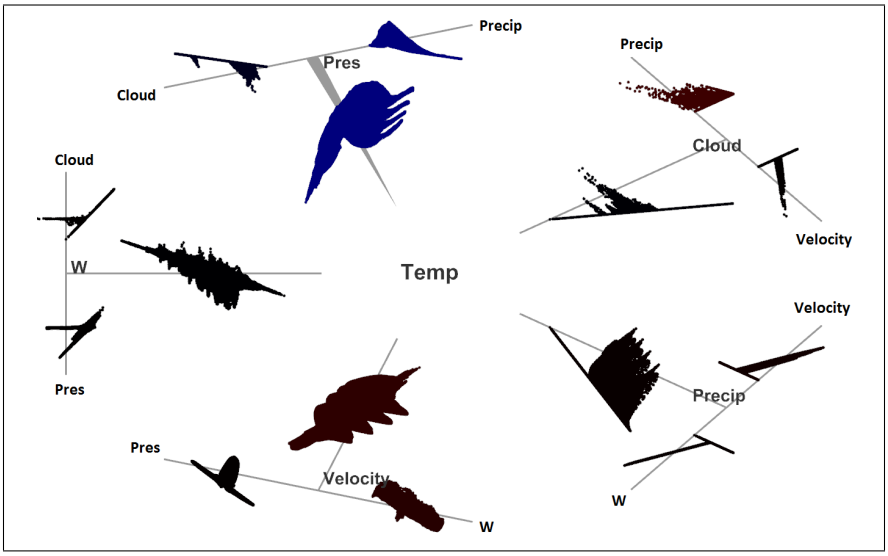


(b) Scatterplot matrix

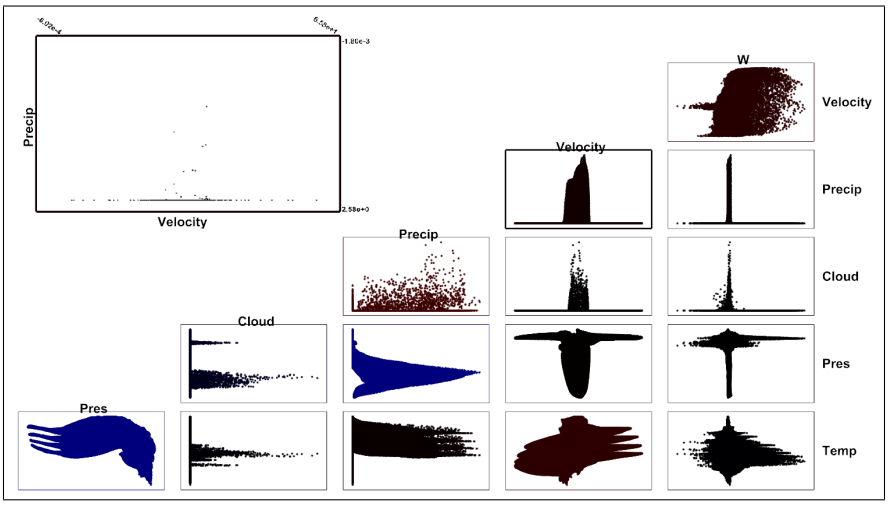


(c) Parallel Coordinates

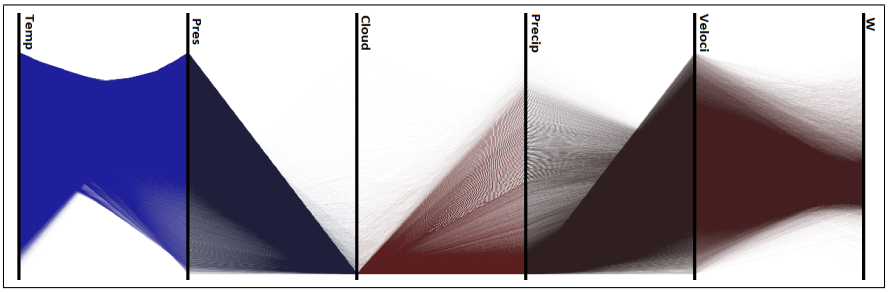
Figure 4.8: Visualizations for pollen data.



(a) Snowflake visualization

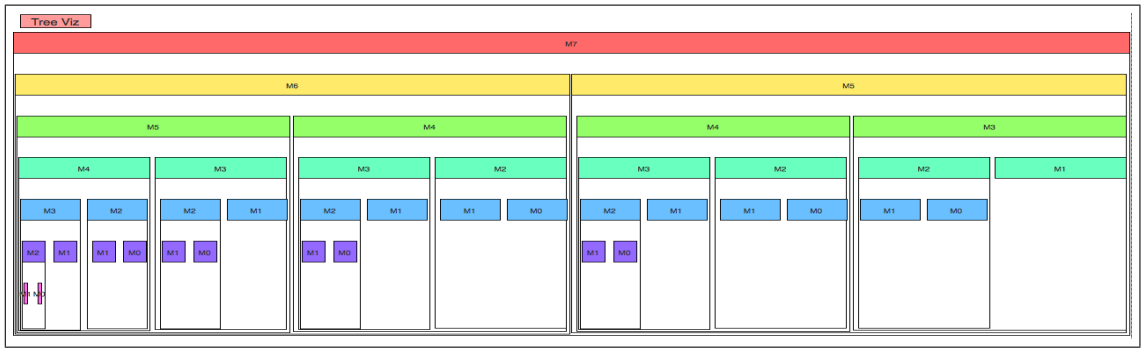


(b) Scatterplot matrix

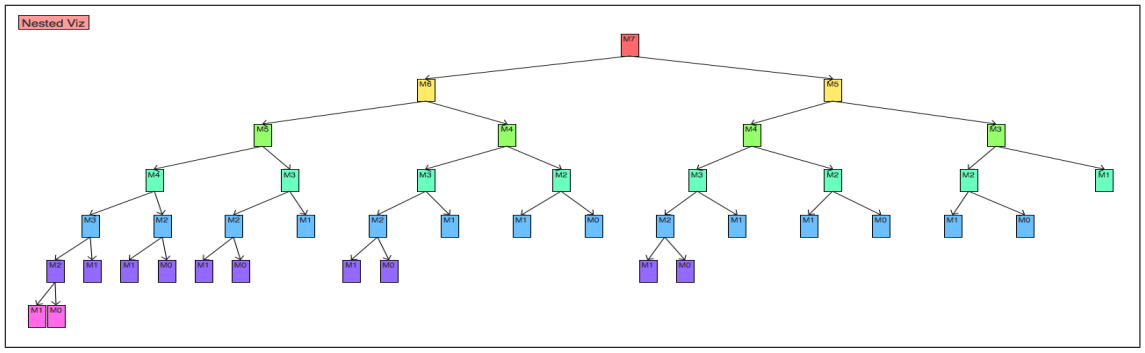


(c) Parallel Coordinates

Figure 4.9: Visualization techniques for Hurricane data.



(a) Nested Graph for Fib



(b) Tree Graph for Fib

Figure 4.10: Interactive flow graph.

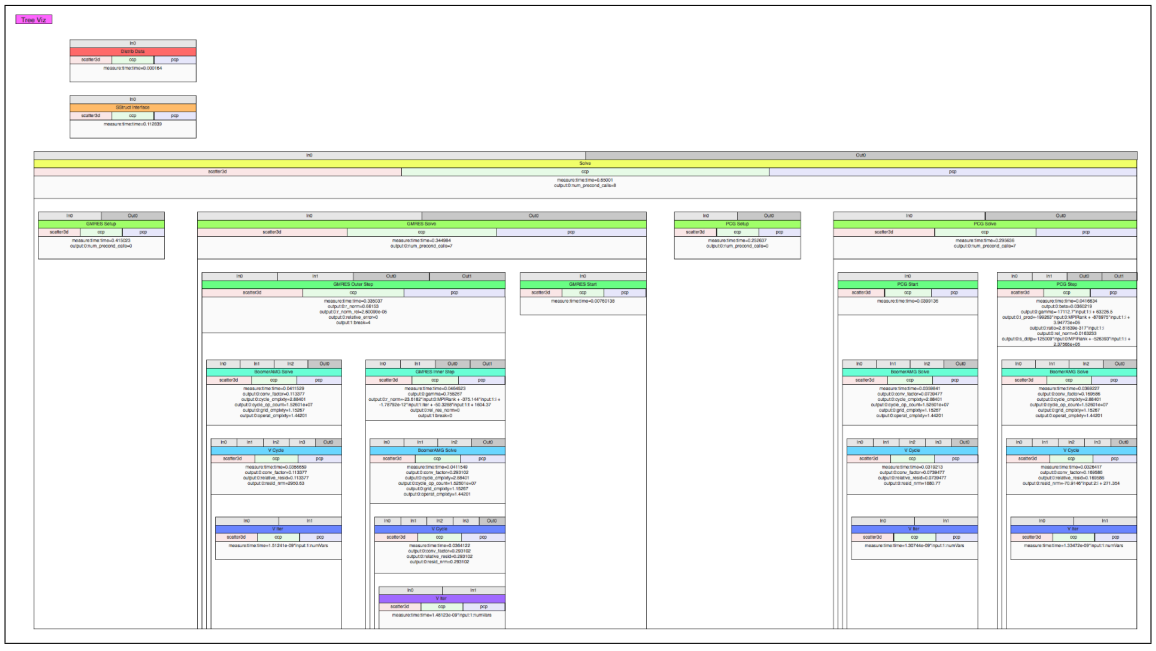
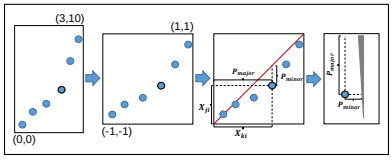
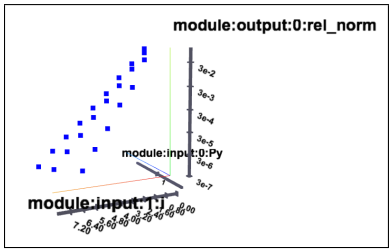


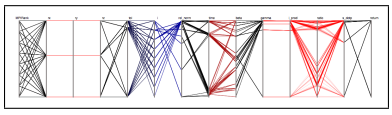
Figure 4.11: Visualization for AMG program.



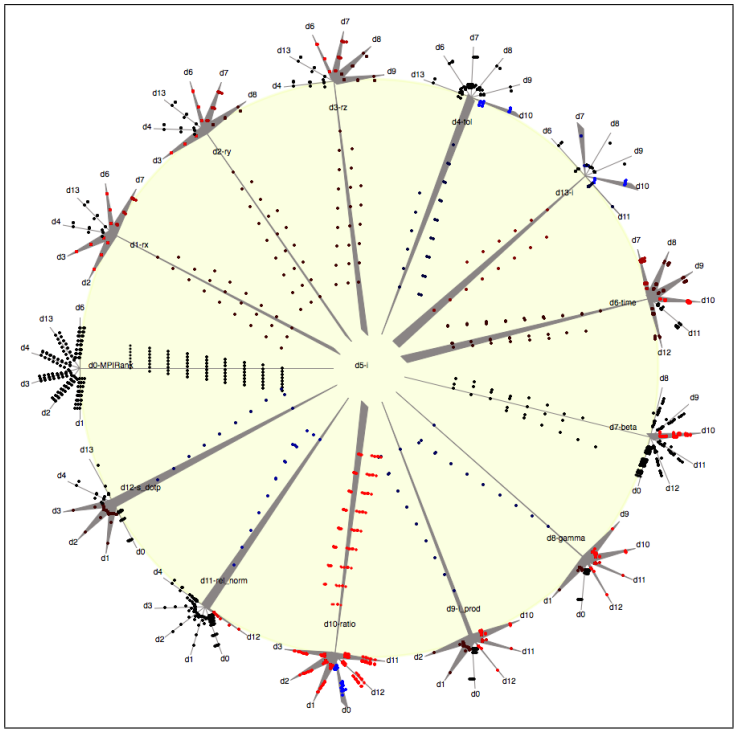
(a) CCP coordinate system



(b) Scatter Plot 3D



(c) Parallel Coordinates



(d) Snowflake visualization

Figure 4.12: Multidimensional visualizations for AMG.

CHAPTER 5

CORRELATION VISUALIZATION FOR LARGE-DIMENSIONAL DATA

5.1 Visual Design

CCP and Snowflake Visualization as in Fig. 5.1 help users to improve correlation identification when the number of dimensions and data points is not high. We now propose a new method that is data scalable and improves local and global relationship presentation in PCPs. This new representation shows trends in data as well as their consistency. Firstly, we transform data in the Cartesian domain into relationship histograms in the parallel coordinate domain. Then the new visual design for PCPs represents the relationship histograms as histogram contours.

To improve local and global relationship presentation in PCPs, we propose a new visual design. This new representation shows both trends in data, large and small, as well as their consistency.

5.1.1 Visual Encodings

Three important visual features are used when analyzing data with PCPs: 1) the angles of line segments, giving clues as to positive or negative relationships; 2) the colocation of line segment crossings, giving clues as to the strength of the relationships; and 3) the distribution or density of line segments, which can differentiate between trends and outliers.

For example, examine the basic PCP plots in Fig. 5.2. 1) In Fig. 5.2a, the angle of the lines relative to one another in the left example indicate a perfectly negative relationship, whereas the parallel lines of the right example indicate a positive relationship. 2) Notice that the position of line crossing in the Fig. 5.2b is not co-located. This spreading indicates a weak negative relationship. 3) Finally, the distribution line segments can differentiate trends and outliers. In Fig. 5.2c, the main trend appears to be the three points on the bottom with an outlier on the top.

Instead of using conventional visual encodings of PCPs, such as lines, density-based, or frequency-based visual encodings, we use the shape, a consistency map, and data distribution histograms in our visual encoding to bring new insight for PCPs.

Two important shapes come to mind when trying to understand the relationships of PCPs. Positive and negative relationships can be identified by seeing a comb and bowtie shape, respectively. We encode this important information by representing the extremities of the data as the overall shape by capturing the outline of the concave hull containing all PCP lines. This implied relationship is represented by the shapes in Fig. 5.3 (right column). This relationship supports PCP semantic 1, Fig. 5.2a.

We color plots red for positive relationships or blue for negative relationships as a secondary encoding to the shape. Since this is a support encoding, should color be needed for another purpose, the redundant encoding can be dropped.

The shape implies only a positive or negative relationship. Details of the trend are important as well. We use colored histogram contours to represent the underlying features of the data. As we will discuss in Section 5.2, these locations are calculated from the individual data and are akin to line segment crossings of geometry-based PCPs. Organized clusters of these points indicate strong trends, whereas scattered versions indicate noise. Similarly, the shape of the points gives clues as to the linearity or nonlinearity of the data, supporting PCP semantic 2, Fig. 5.2b.

The distribution of data items (i.e., the histogram for each axis) is represented as a distribution curve along the axes of the PCP. Without this information, outliers may cause users to misinterpret certain patterns [74]. The data distribution histogram is created by calculating a histogram of the data items and applying a Gaussian distribution to draw a smooth curve along the domain, which can be seen in Fig. 5.3 (right column) as the purple curve near the axes. The maximum height and thickness of these histograms are adjustable values, in case more or less emphasis is desired. This relationship helps to support determining the density of points, supporting PCP semantic 3, Fig. 5.2c.

Beyond the static visualization, the approach provides interactions such as locating and tracing individual and groups of data items through brushing. In addition, users can reorder axes in a drag-and-drop manner similar to classic PCPs. We have precomputation for the new mapping so the performance is not changed during rendering even when interaction is performed such as reordering axes.

5.1.2 Plot Interpretation

We examine the capabilities of our new visualization by using four synthetically generated datasets, each containing $50,000$ data items. The first two datasets are linear relationships ($y = ax + b + \varepsilon$), one positive (Fig. 5.3a) and one negative (Fig. 5.3b). The second two are quadratics ($y = a(x+\varepsilon)^2 + b(x+\varepsilon) + c$), again one positive (Fig. 5.4a) and one negative (Fig. 5.4b); ε is a noise factor.

5.1.2.1 Detecting Positive and Negative Relationships

To understand the direction of the relationship, two key visual encodings can be used, color and shape. The red rectangle shape represents a positive relationship. The blue bowtie shape represents a negative relationship. For the positive case, the strength of the relationship is indicated by the distribution of points in the consistency map. Spread in the horizontal direction, such as that of Fig. 5.4a (bottom) indicates noise in the relationship. In the negative case, both the spread in the consistency map and the loosening of the bowtie shape indicate weaker relationships. Fig. 5.4b (bottom) shows the effects of adding noise to the data, spreading both the contour and shape.

5.1.2.2 Detecting Linear Relationships

Quantifying linear relationships in PCPs is generally less accurate and slower than scatterplots, and large numbers of items can cause serious problems for both [75], [44]. Fig. 5.3a (top) and 5.3b (top) show that it is easy to identify linear relationships for large numbers of data items using our method. For positive relationships, the standard PCP shows the dense parallel lines as a filled square, whereas our approach shows the consistency map as a vertical bar between the axes. When negative, the PCP lines cross at a single point. Our method shows only the boundary of these lines and a consistency map that focuses around the intersection point.

When noise is presented, our approach can still detect global linear patterns in data. Fig. 5.3a (bottom) and 5.3b (bottom) both show noise spreading the consistency maps. However, we are still able to identify the overall trend, as well as the noise. For the angular histogram/opacity PCPs, the overall trend is still visible, but the extent of the noise is rather difficult to ascertain.

5.1.2.3 Detecting Nonlinear Relationships

Identifying nonlinear patterns is not well supported by most incarnations of PCPs. Fig. 5.4a (top) and 5.4b (top) show a quadratic relationship. Using our approach, the curved features of the

relationship between data attributes are easy to identify. In the positive case, this feature can be seen in the consistency map. In the negative case, this feature can again be seen in both the shape of the relationship and the consistency map.

When noise is added to the data (Fig. 5.4a (bottom) and 5.4b (bottom)), it can be difficult to identify the relationship in angular histogram/opacity PCPs. However, in our approach, the global trend as well as the volume of the noise is still visible.

This result illustrates that our approach supports identification of the relationship strength through co-located crossings (Fig. 5.2b).

5.2 Building Consistency Maps

A large and complex dataset requires a new data transformation method from the Cartesian domain to the PCP domain that retains the important features and supports a variety of visualization tasks [76]. The mapping of multidimensional data projections can support exploring the main features of large data [77], [78]. We propose a consistency map as a data transformation methodology that represents the important relationship patterns and overcomes the overdraw problem.

5.2.1 Global Trends Using Locally Linear Relationships

Given two attributes, we assume that the relationship between them is *locally linear* [79]. Observing this relationship with many local linear trends, we can model complex global relationships.

Given a small number of data items, principal component analysis (PCA) [80] can be used to extract the orientation $(\tilde{u}, \tilde{v})$ of the data (i.e., the eigenvector of the covariance matrix) as well as a magnitude m_1 (i.e., the square root of the eigenvalue) that can be considered a measure of the relationship strength [81].

PCA can also extract an orthogonal direction and magnitude, m_2, of the second principal component. The ratio of two magnitudes, $g = \frac{m_2}{m_1}$, can be used as a measure of the "linearness" of a local region. It is always true that $m_2 \leq m_1$. However, $m_2 = m_1$ implies that there is no clear orientation of the data points. On the other hand, when $m_2 \ll m_1$, the data items are configured with a strong linear trend.

5.2.2 Identifying Local Groups

We first identify local groups of data items in the Cartesian domain. For each item in the dataset, we use the k-nearest neighbors (*knn*) algorithm [82] to find those groupings as shown in

Fig. 5.5a. Our implementation is optimized by placing all items into a quadtree (see Fig. 2.1 (left)) and searching neighboring leaves. For a dataset of n items, n groups are extracted, each containing $k+1$ items (the center point plus k neighbors).

For each group, the direction $< \tilde{u}, \tilde{v} >$ and magnitudes m_1 and m_2 are extracted using PCA. The mean location of the group (x_m, y_m) and vector $< \tilde{u}, \tilde{v} >$ are then mapped to location (q, r) using point/line duality principal of PCP's, based upon Equations 5.1 and 5.2.

$$q(\tilde{u}, \tilde{v}) = \frac{\tilde{u}}{\tilde{u} - \tilde{v}} \tag{5.1}$$

$$r(\tilde{u}, \tilde{v}) = x_0 + (y_0 - x_0)\frac{\tilde{u}}{\tilde{u} - \tilde{v}} \tag{5.2}$$

Fig. 5.5 shows a schematic of the process. In this case, the many groups of similar direction map to the same general area in the PCP domain, which is a clear indication of directional similarity. Now, this approach works perfectly in the case of negatively related points. However, a problem arises as we look at positively related points. Namely, they do not properly map to the PCP domain.

5.2.3 Mappability of Positive Relationships

Mappability refers to the ability to calculate a valid output location (i.e., valid q and r values) within the drawing space for a data item. As q is currently defined, only values between 0 and 1 appear between the PCP axes. This set of q values consists exclusively of *negative relationships*. Fig. 5.6c demonstrates this mapping by showing the value of q plotted against the angular direction of $(\tilde{u}, \tilde{v})$. Negative relationships all exist in the range of $q \in [0, 1]$, but *positive relationships reveal two challenges*.

Point/line duality essentially boils down to an intersection of two lines mapped into parallel coordinates. First, by our definition, *no positive relationships* will be mappable because their values are $q \notin [0, 1]$. Secondarily, with line-line intersections, numeric instabilities occur when the lines are near parallel. For us, this occurs when $\tilde{u} = \tilde{v}$, or in other words, it occurs when the direction represents a *perfectly positive relationship*.

Since values for positive relationships cannot be mapped, we can make a simple choice: use the orthogonal vector, $(-\tilde{v}, \tilde{u})$, when the relationship is positive.

$$q'(\tilde{u},\tilde{v}) = \begin{cases} q(\tilde{u},\tilde{v}) & \text{if } 0 \leq q(\tilde{u},\tilde{v}) \leq 1 \\ q(-\tilde{v},\tilde{u}) & \text{otherwise} \end{cases} \tag{5.3}$$

$$r'(\tilde{u},\tilde{v}) = \begin{cases} r(\tilde{u},\tilde{v}) & \text{if } 0 \leq q(\tilde{u},\tilde{v}) \leq 1 \\ r(-\tilde{v},\tilde{u}) & \text{otherwise} \end{cases} \tag{5.4}$$

Using the orthogonal vector now guarantees that all relationships will map to a valid location in the output PCP. However, it is important to understand how that change impacts the location of points.

Fig. 5.6 shows the projection location for various angles of orientation, relative to the unit circle. The red lines represent angles of 22.5°, 45°, and 67.5°, respectively, as in Fig. 5.6a. When the red lines are transformed from Cartesian coordinates (top) to parallel coordinates (bottom), their intersection points extend beyond the extremes of the axes. However, the orthogonal versions in blue, as shown in Fig. 5.6b, all generate valid intersections.

The resulting q and q' values for a set of angles in Cartesian coordinates can be seen in Fig. 5.6c. The horizontal location of those angles in parallel coordinates can be seen in Fig. 5.6d.

Note, these relationships have orientation but no direction. Thereby, they create a consistent mapping wrapped around the unit circle.

5.2.4 Histogram Contours

The final step of the data transformation is placing the point (q', r') into a histogram. We use triangular histograms, such as those seen in Fig. 5.7a. The location (q', r') influences bins within a radius of influence found using $1 - g$, which means that more linear groups of data have a larger influence area.

To express the information contained within a single relationship histogram, we have chosen to use a variation of the triangular isobanding algorithm [83] to show the adherence to the local linear trend. Our approach defines bands along the range $[\beta, \infty)$.

5.2.5 Selecting k

Our approach requires selecting a constant k used in knn. Instead of specifying a single value, we generate multiple histograms based on powers-of-two values for k, which in effect enables finding patterns at many different scales. Small k values will grab small-scale linear relationships, whereas larger k values will tend to identify larger global linear relationships. In effect, we are scanning a range of possible frequencies for the Nyquist rate of features.

To demonstrate the behavior of multiple scales (i.e., multiple values of k), we composite isobanding results. Each value of k receives a different lightness value under the same hue. Fig. 5.7b shows individual values of k, and Fig. 5.7c (left) shows the composite.

5.3 Representing Multiple Relationships

Whereas many data are representable through a single trend, only supporting such data are incomplete. Support for representing and differentiating multiple relationships is important in real applications.

To accomplish representing multiple relationships, we classify data into subgroups representing various relationships. Each subset is treated independently with the process described in section 5.1 and 5.2 (i.e., each group has its own shape and consistency map calculated). Each is rendered separately and layered in the visualization, with the ordering of the layers controlled through clicking or scrolling. Our approach is agnostic of the method for classifying the subgroups. We present three approaches that we have found useful, two automatic and one user manipulated.

5.3.1 Global Clustering

We allow defining clusters globally [84]. The approach normalizes all attributes and then uses the ℓ^2-norm for distance (i.e., the Euclidean distance). We use the k-means clustering algorithm [85] for dividing data into subgroups. k-means clustering is an iterative approach to clustering that works by identifying $\hat{k}$ cluster centers (this $\hat{k}$ is a different from that of k-nearest neighbors), adding data items to the closest center, and repeating.

Our method iteratively searches for an appropriate number of clusters by first starting with one cluster. It calculates the Pearson correlation coefficient, $\rho(x,y)$, on all clusters and attributes, and if any $\mid \rho(x,y) \mid < \alpha$, the number of clusters is increased. When $\mid \rho(x,y) \mid < \alpha$, two attributes have very low correlation or no correlation. By this method, we find $\hat{k}$ valuable clusters. For all figures, we use $\alpha = 0.15$.

5.3.2 Pairwise Clustering

Our second clustering technique is a pairwise approach that works in a manner somewhat similar to that of the previous one, using k-means and the ℓ^2-norm for distance. However, it also aims at clustering items that have similar trends, not just those with similar values.

To accomplish this, we compute a specialized vector for each data item. The first component of

the vector is the normalized values of the attributes (x,y). The next component is the (q,r) value for each k used to model the multiscale relationships. The final vector used to segment is constructed as $[\ (x,y),\ (q,r)_1,\ ...,\ (q,r)_k\ ]$. The result of using this vector is that data items with both similar attribute values, as well as similar local trends, get clustered together.

Again, we iteratively search for an appropriate number of clusters by starting with 1 and using the Pearson correlation coefficient ($\mid \rho(x,y) \mid < \alpha$ with $\alpha = 0.15$) to determine if additional clusters are needed.

5.3.3 Brushing

We enable two forms of brushing. First, as with conventional PCPs, we provide users the ability to brush a region and have all crossing data items drawn individually. With this approach, the behavior of all items across all attributes can be observed (see Fig. 5.8a). Second, we enable brushing to select a cluster of data. Once selected a new relationship subgroup is created with the data items that have been brushed, and that group is visualized using our visual encoding approach. Fig. 5.8b shows in green the result of a brushing over four data attributes. As the display is brushed, all data items crossed by the brushing action are added to a new subgroup. When the mouse is released, the subset is recalculated and the resulting trend is displayed.

5.4 Evaluation

We demonstrate the capabilities of our method in identifying multiple trends on two datasets. The first dataset was the synthetic data in Section 5.1.2. Next, we use a particle physics dataset containing 41 output attributes and 4000 data items. The data represent a parameter space search of 25 input attributes produced by tools that model subatomic particles under the supersymmetric extension of the Standard Model. This dataset has clear linear and nonlinear relationship patterns without much noise.

Third, we use the IEEE Visualization 2004 contest dataset[1], Hurricane Isabel, consisting of 48 timesteps, each containing measurements of 11 attributes with a spatial resolution of $500 \times 500 \times 100$. Of the original 25 million data items, we use only 10 million because approximately 15 million items contain at least one invalid *NaN* field. This dataset is large and contains mostly low-level noise.

[1] http://vis.computer.org/vis2004contest/

Fourth, we use the HIGGS dataset[2], containing 28 attributes and 11 million data items. The dataset has been produced using Monte Carlo simulations. The first 21 features are kinematic properties measured by the detectors in a particle accelerator. The HIGGS dataset is both large and noisy.

Finally, we use the Planet dataset[3]. This dataset includes the data from ground and space-based observations. The dataset contains 1827 items and 16 attributes such as planet mass, planet radius, planet density, distance, optical magnitude, etc.

We use the IEEE Visualization 2004 contest dataset[4], Hurricane Isabel, consisting of 48 timesteps, each containing measurements of 11 attributes with a spatial resolution of $500 \times 500 \times 100$. We show seven of the more "interesting" attributes for the first timestep. Of the original resolution data 25 million data items, we only use 10 million because approximately 15 million items contain at least one invalid *NaN* field.

In the following sections, we evaluate our method in visualization tasks including cluster analysis, hidden pattern detection, noise detection, and outlier detection. We compare our methods with basic implementations of classic, opacity PCPs, angular histogram PCPs [44], and Johanson's work on PCP [50].

5.4.1 Performance

We built our software using C++, OpenGL, and Qt. All experiments were conducted on a PC with Intel Core i7 CPU 2.66GHz, NVIDIA GK104 graphic card. We use histogram bins of 2^9 and isoband threshold $\beta = 2^6$ in all of our experiments. The performance comparison of our method and opacity PCPs is provided in Table 5.1 for all datasets. Although our precomputational cost was always greater, the rendering performance per frame for our approach was 2.9 to 3.6 times faster than our angular histogram and opacity PCP implementation. Our precomputational cost consists primarily of consistency map calculations including k-nearest neighbors and clustering, but per frame rendering requires only a few primitives. On the other hand, the many lines drawn in the opacity PCPs make its rendering time burdensome and not scalable with additional data items. New

[2]http://archive.ics.uci.edu/ml/datasets/HIGGS

[3]http://exoplanetarchive.ipac.caltech.edu

[4]http://vis.computer.org/vis2004contest/

mappings are precomputed such that performance is not impacted when interaction, such as axis reordering, is performed.

5.4.2 Particle: Mixed Trends

As the number of data items becomes large or data become more complicated, cluster analysis is challenging for a classic PCP, ultimately relying on user interaction techniques such as brushing. PCPs somewhat alleviate this problem by highlighting major trends in the data. However, smaller trends may wash out. In the angular histogram, the direction and length of bars can help users identify curtain types of pairwise clusters, but do not make it easy to understand overlapping clusters or any kind of global clustering. Our method naturally supports cluster differentiation tasks for both global clusters and pairwise clusters. Fig. 5.1 highlights the usage for global clusters, and Fig. 5.9d highlights the usage for pairs of attributes.

In Fig. 5.9, we can see the difference between the classic PCP, opacity PCP, angular histogram, and our approach with the *Particle* dataset. For example, we consider the attributes *ncmass4* and *ncmass5*. Within the PCP, the values appear well distributed across the range of *ncmass4* but focus at a single value on *ncmass5*. The remaining points appear to be outliers. Both the opacity PCP and angular histogram emphasize this same conclusion. However, using our approach, the attributes have three clusters appear between them, one strong negative cluster and two weak positive clusters. Observing the scatterplot for these attributes reveals that this is a better representation of nonlinear structure. The points can be disassembled into three parts (see Fig. 5.9f): the positively associated portion on the top left; the positively associated portion on the lower right; and the negatively associated portion connecting them. This connection is completely missing from the other three PCP visualizations.

Another example of this problem can be seen in the *d8* and *d31* attributes. With the classic PCP, much of the complexity of the relationship is lost, though there are some clues to complexity. In the worst case, one would be tempted to assume this to be a single negative relationship. In the case of the angular histogram and opacity PCP, a bifurcated relationship is apparent, one negative, terminating at the top of *d31*, and one positive, terminating at the bottom of *d31*. Using our approach, four clusters, two strongly negative and two weakly positive clusters, are identified. In Fig. 5.9d, the primary negative cluster is visible in front and highlighted by thicker boundary lines. This cluster was also visible in the opacity PCP. The second negative cluster can be selected and

brought to the front as shown in Fig. 5.9e. The data points constructing this cluster are clearly visible in the PCP, although they are difficult to visually separate and lost in the opacity PCP, due to their low density. Observing the scatterplot and schematic view (Fig. 5.9f), we can spot the four clusters that make up these relationships.

5.4.3 Hurricane: Overdraw and Underdraw

An overarching challenge (and subject of numerous papers) for classic PCPs is overdraw, particularly with data containing many items. For datasets such as the Hurricane dataset, which contains 10 million data items, patterns can be hidden by the many layers of lines drawn. In Fig. 5.10b, the major relationships between most attributes are difficult to visually identify, and those identified should be treated with some skepticism. This problem also exists for scatterplots (also the topic of numerous papers) as shown in Fig. 5.10a. The angular histogram and opacity PCP (Fig. 5.10c) alleviates the problem to some extent by adapting to the density of the data, but nevertheless, remains limited. As the number of data items and complexity of relationships increase, lesser relationships may be lost.

Looking at the *Temperature* and *Pressure* attributes, we can immediately see an example of overdraw in the classic PCP. Without further investigation, we would assume a single negatively related trend. The angular histogram and opacity PCP correct this issue, making the true shape of the trend visible. Similarly, our approach reveals three trends, two negative trends (in blue) and one positive trend (in red). The key piece missing from the angular histogram and opacity PCP is any indication of the noise within the data. In the angular histogram and opacity PCP, the data appear uniform. Observing freckle pattern in our approach indicates that the relationship is noisy, which can be confirmed via the scatterplot.

More generally speaking, simultaneous representation of global trends and outliers is hard—most often visualization methods either only focus on global trends, or on the cost of hiding outliers, or focus on outliers, causing ambiguity among major trends [86].

The *Pressure* and *Cloud* attributes are a good example of this. Fig. 5.10b shows a classic PCP where the major trend and some outliers are visible. Unfortunately, the major trend is challenging to interpret because of overdraw, but at least some of the outliers are visible. The opposite problem occurs with the angular histogram and opacity PCP, as in Fig. 5.10c. Much of the detail of the major trends is now visible, at the cost of losing almost all outlier information. This is an example of

underdraw. The angular histogram can help identify outliers by tracing the small purple bars. One strength of Johanson's [50] and followup works is the use of such mappings to highlight specific features such as clusters and outliers.

Fig. 5.10d shows how our approach enables finding both major trends and detecting outliers between *Pressure* and *Cloud*. Our approach reveals two trends, one negative trend (in blue) representing the major trend and one positive trend (in red) that captures the outliers.

One concern at this point is to the ambiguity of which trend is the major trend versus the outlier trend. The visual clue that differentiates them is the purple curve representing data item density. The density is high at the bottom of the *Cloud* attribute, indicating that almost all data items fall into that particular cluster. This is a procedure similar to that for angular histograms. Should one wish to investigate further, item selection and brushing interactions enable a deeper dive.

5.4.4 HIGGS: Noisy Relationships

Visual detection of data relationship can be difficult with noisy data. With the 11 millions of items in the *HIGGS* dataset, understanding the relationships is difficult, particularly considering the noise.

Consider the relationship between *jet1eta* and *jet1phi*. Because of overdraw in the SCP, the complexity of the relationship (containing both local positive and negative relationships) [23], and because of the noise, it is difficult to identify any relationship through the SCP (Fig. 5.11a). The Pearson correlation coefficient between *jet1eta* and *jet1phi* is -0.102, showing that they have a weak negative relationship. However, this relationship is barely visible in the SCP.

The opacity PCP (Fig. 5.11c) helps to clarify the noisy nature of the relationship, but it does nothing to disambiguate the issue with the relationship direction. Similarly, the angular histogram (Fig. 5.11c) reinforces the positive relationship misconception.

Our approach, on the other hand, identifies three relationships, as shown in Fig. 5.11d. Two are minor positive relationships, whereas the third is a large negative relationships. Furthermore, the large size of the bowtie and freckled pattern contained within it indicate that the relationship is noisy and weak.

Another example of this can be found between *jet1phi* and *m_wbb*, where the Pearson correlation coefficient is -0.132. The additional visual encodes provided by our approach enable identification of this weak noisy negative relationship.

5.4.5 User Feedback

We have conducted four interviews with users related to our approach. Each interview was one hour and used a different dataset. One participant was an advanced visualization PhD student, and the other three were nonvisualization users.

5.4.5.1 Planet Data

Our first interview involved a demonstration and interview with an advanced visualization PhD student. The student's work involved developing an analysis tool for the *planet* data.

To begin, we first showed him our approach with the synthetic data (presented in Section 5.1.2) to acclimate him to using our tool to understand data relationships. After that process, we loaded in the *planet* data. Fig. 5.12b shows our approach for four dimensions, *vj*, *teff*, *mass*, and *rad*. Fig. 5.12a shows angular histogram and opacity parallel coordinates plots for the same dimensions.

With our approach, the student identified some interesting information. Among his observations, in Fig. 5.12b, he found that *stteff* and *stmass* have weak nonlinear and positive relationships, previously unknown. This is not clearly visible in the opacity PCP and angular histogram. He also found the complex relationship between *stteff* and *stmass* interesting using our approach.

In the end of the interview, he shared his opinions about our method. First, he commented that the method required remembering two mechanisms for reading the positive and negative cases. He agreed that this is similar to the standard PCP. He commented that once he learned how to use our approach, it was easy to understand the data relationships. Finally, he commented on our clustering mechanism. He stated that he would prefer to see some overview of relationships before looking through clusters and choosing the interesting ones. Our method partially supports this by highlighting the main cluster first.

5.4.5.2 Particle Data

We interviewed a second individual with the particle physics dataset as shown in Fig. 5.9. He thought the relationship of *d32* and *d33* was difficult to identify in the scatterplot, PCP, and angular histogram PCP, but it was easily seen in our method. The scatterplot shows the points are dense on the top left and spread out towards the lower right. However, this method does not indicate the true relationship. The traditional PCP shows the bowtie shape but with significant overdraw. The angular histogram PCP helped him to see the distribution of the data and guess the relationship, but it was difficult for him to identify the direction of the relationship. By using our approach, he easily

found the main negative relationship and could estimate its strength using the contours.

5.4.5.3 Hurricane Data

We conducted an interview with a third individual using the Hurricane dataset as shown in Fig. 5.10. After an explanation of the approach, he was interested in the relationship between *Pressure* and *Velocity*. Using scatterplot and PCP, he could not determine if it was positive or negative. Then he used the angular histogram and recognized that most data point looks like a band, making him think that the relationship between *Pressure* and *Velocity* was positive. When he used our approach, he noted that there is one negative relationship group and two other positive groups. The negative relationship group at the front means *Pressure* and *Velocity* primarily have a negative relationship. He was surprised that the methods led him to two different answers. The Pearson correlation coefficient of these two attributes is -0.13, so globally they have a weakly negative relationship.

5.4.5.4 HIGGS Data

We interviewed a final individual over the HIGGS data as shown in Fig. 5.11. When he saw all scatterplots and PCPs of the data, his immediate reaction was that the data are very noisy and would be difficult to understand. We asked his opinion about the correlation between *jet1eta* and *jet1phi*. First, looking at the scatterplot, he thought the attributes carried no relationship. Then, we showed him the PCP visualization, and he guessed that the attributes had a positive relationship because most of the lines seemed parallel. Seeing the angular histogram further reinforced that belief. However, when he saw our visualization, he was surprised to see it was a negative relationship with noise. Finally, we told him that these two dimensions had a Pearson correlation coefficient of -0.102, confirming the information presented using our approach.

5.5 Discussion

5.5.1 Comparison with PCP Alternatives

Generally speaking, geometry-based PCPs suffer from overdraw problems. Geometry-based PCPs can help users identify individual data items for pairwise attributes or across all data attributes. However, there are many limitations of geometry-based PCPs when data are large, including difficulty in identifying trends, outliers, and interpreting noise.

Frequency-based PCPs overcome many of the geometry-based limitations to help users explore

clusters, linear relationships, and outliers in data, while avoiding overdraw. However, frequency-based PCPs, such as angular histogram PCPs, are still limited in their ability to identify nonlinear relationships. Furthermore, angular histogram PCPs aggregate the frequency of the lines between pairs of axes, which means users can identify only the principal trend of data and will have a difficult time interpreting mixed trends or outliers within the data.

Density-based PCPs have addressed overdraw by replacing opaque lines with a density representation. Heinrich et al. did this with continuous parallel coordinates (CPC) [49], [54]. They provide a mathematical model of point density for counting discrete lines. CPC naturally avoids overdraw in the continuous domain, but the continuous domain lacks an efficient mechanism to map features back to the original data items. Furthermore, the CPC uses linear interpolation, which may produce less accurate results for higher-order characteristics. Finally, CPC visualizes data as uninterrupted, but discontinuities can represent structures that might be meaningful for the interpretation of some data [87]. Adopting this idea, Lehmann introduced the curve-curve duality and circle-area duality to highlight curves that are dominant structures [88].

Global clusters in multidimensional data can be identified in conventional PCPs and multivariate scatterplots [78], [76]. These multivariate scatterplot methods improve correlation identification accuracy, completeness, distortion, and interactions for less noisy data, but these methods become difficult to use when data are noisy. On the other hand, our approach reveals noisy global relationships well (assuming we select a global clustering technique), even when data are noisy.

Our approach does not suffer from overdraw, as drawing is independent of both resolution and data size, enabling performing the visual analysis tasks we have enumerated very effectively. These tasks include easily identifying both global and local trends, expressing nonlinear relationships, identifying outliers, and detecting noise.

5.5.2 Crossing Points and Extracting Relationships

In conventional PCPs, finding the crossing points between data items is an important part of understanding the relationships among attributes. For example, many lines crossing at a single point indicates a strong negative relationship. However, this methodology does not stand up as large numbers of data items overlap. Our approach addresses this problem by removing drawing of individual lines and instead focuses on representing the local relationships. The beauty of our approach is that the local relationships we extracted are, in fact, loosely correspondent to the

crossing point that we see in a conventional PCP. Our approach naturally focuses similar behaviors into the same area of the output plot, culls irrelevant crossing points, and removes the visual cluster of drawing many overlapping lines.

5.5.3 Features Through Variations of k

An important contribution of our work is the use of multiple values of k for modeling locally linear relationships (k in k-nearest neighbors algorithms). Variations in k enable extracting features on multiple scales. If the value of k is too small relative to a feature, then it may appear as noise, or when the value of k is large, our method will measure only the global relationship of data. However, the variation of k enables capturing all scales of relationship from local to global, giving us access to the right structure of the data.

5.5.4 Selecting the Number of Clusters

Selecting the correct number of clusters is, in general, an important problem. If incorrect, features may be mixed or split. Although we used k-means clustering, substituting another method such as k-means++ may be helpful in improving clustering results. However, the best choices for clustering (both algorithm and k) remain outside the scope of this particular work.

5.5.5 Distribution Curves

When compared with an angular histogram, the distribution curve in our method is also a histogram of data items that does not show the direction of those data. In our case, understanding data directions can be accomplished by inspecting the shape and consistency maps and using interaction. Nevertheless, an angular histogram could easily be substituted for our distribution curves, if desired.

5.5.6 Information Lost Through Abstraction

Overall, our abstractions loses very little information relative to overdrawn PCPs. The only significant downside we have identified is that it lends itself to false equivalency bias between trends of different importance. For example, take an imaginary dataset with two trends. Trend 1 contains 95% of the data points, whereas trend 2 contains 5%. These two trends may appear equivalent within our abstraction scheme. The differentiation could be made through the distribution curves on the axis and histogram visual encodings, although they remain a subtle feature.

We have proposed a data scalable approach for identifying relationships in the parallel coordi-

nates. In this approach, a new model is used for mapping data from its attribute domain into the parallel coordinates domain, which has two major advantages. First, our approach scales well with increases in the size of data and avoids the overdraw problem. Second, using thoughtful encodings, data clustering, and interactions helps users identify relationships previously difficult to find in other types of PCP. Our approach supports identification of linear patterns, nonlinear patterns, mixed patterns, and noise patterns, and enables finding outliers. The results of our experiments for simulated and real-world data demonstrate that our method is practical for high-performance analysis of large and complex data.

We expect that extremely large datasets will be those that most benefit from using our approach. There are also a number of possible works on user analysis using the approaches of Rados et al. [89] or Harrison et al. [90], which would help to understand the qualities of our approach in the context of many popular techniques.

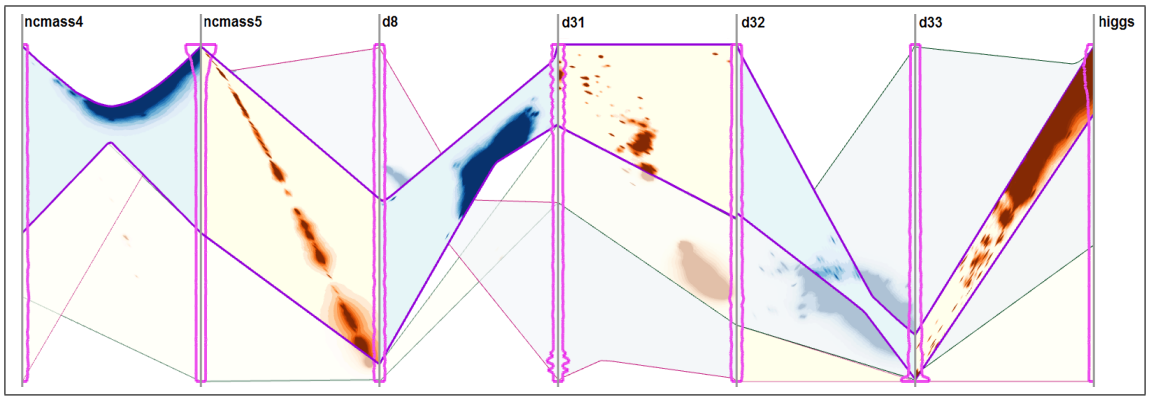
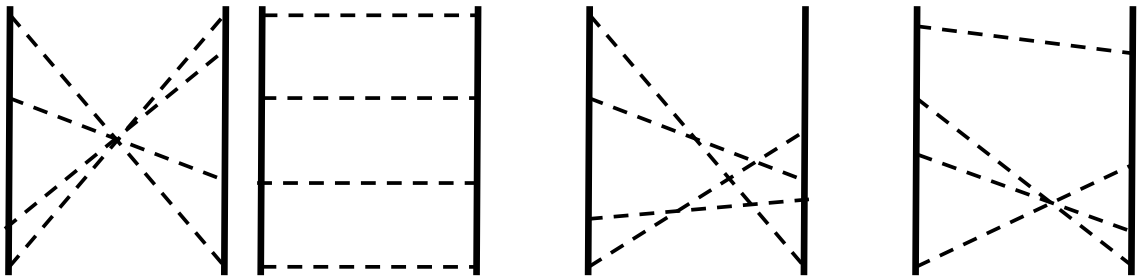


Figure 5.1: Whereas large data overwhelm conventional parallel coordinates plots, our approach uses flexible relationship clustering and summarization to identify large-scale trends in the data, simultaneously highlighting adherence to the trend and showing outlier behavior. Here, trends are tracked across multiple data attributes.

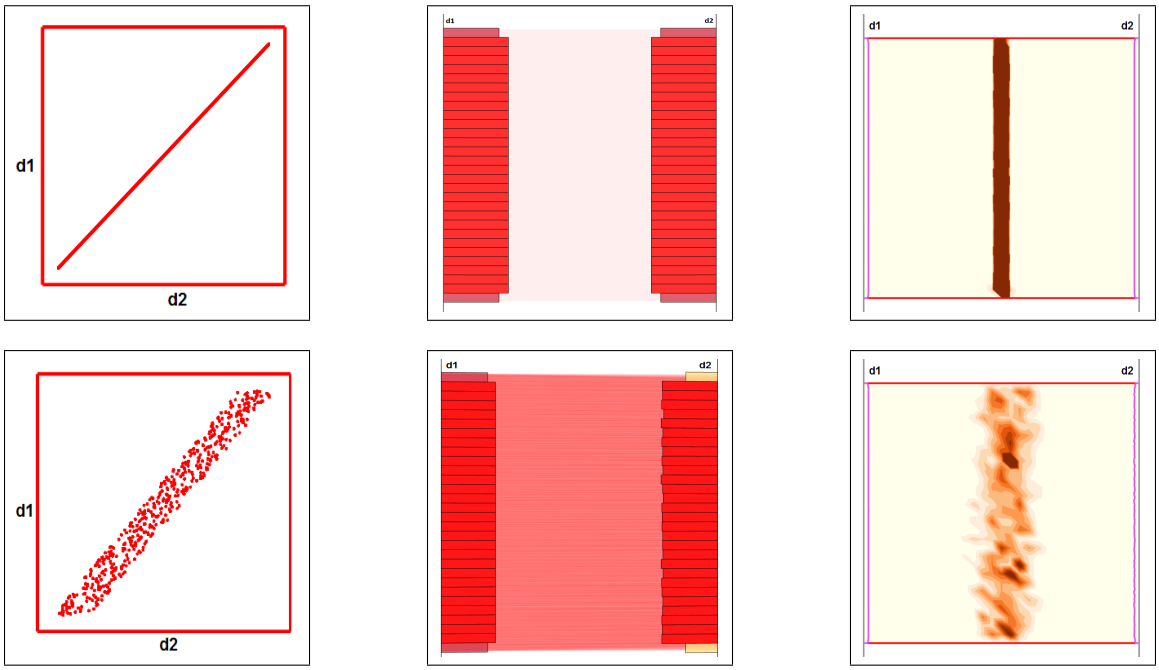


(a) Angle of line segments (b) Location of crossings (c) Density of segments

Figure 5.2: Examples of semantic features of PCPs.

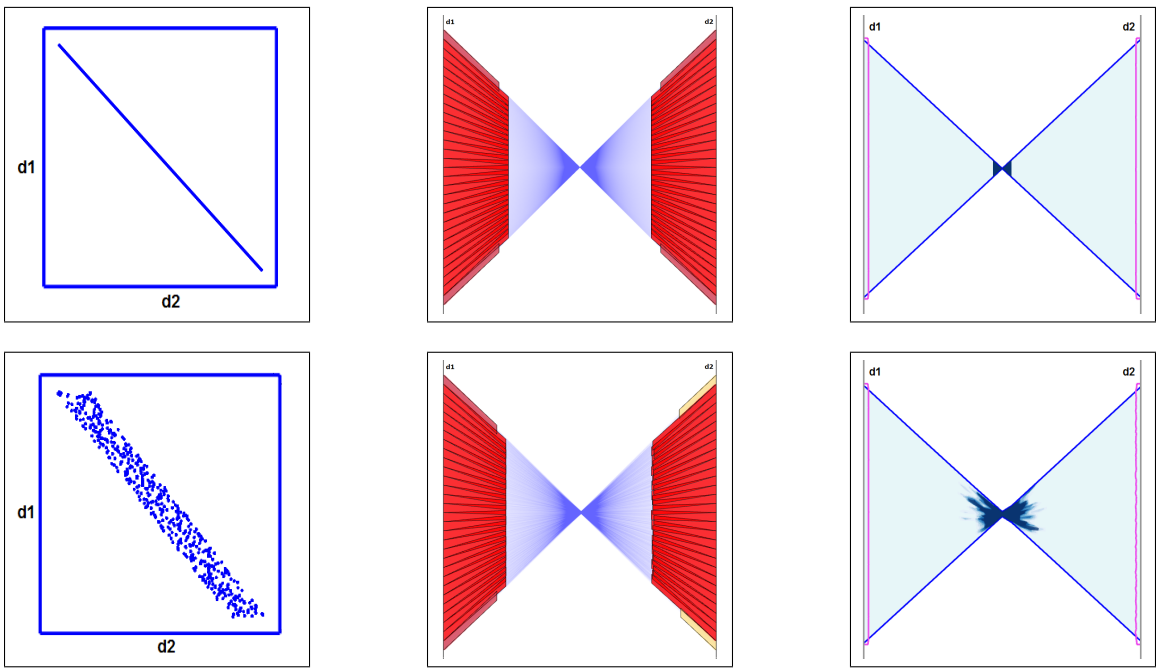
Table 5.1: Precomputation (Precomp) and rendering time per frame (Render) in milliseconds (ms) for our method and angular histogram and opacity PCPs.

	Synthetic (5K items)		Particle (4K items)		Hurricane (10M items)		HIGGS (11M items)		Planet (1.8K items)	
	Precomp	Render	Precomp	Render	Precomp	Render	Precomp	Render	Precomp	Render
Opacity PCP	3 ms	11 ms	2.5 ms	9.5 ms	30 ms	84 ms	38 ms	95 ms	1.1 ms	3.8 ms
Our Approach	8 ms	3 ms	6.2 ms	2.7 ms	104 ms	29 ms	129 ms	32 ms	2.6 ms	1.2 ms
Speedup/(Slowdown)	(2.6x)	3.6x	(2.5x)	3.5x	(3.5x)	2.9x	(3.4x)	2.96x	(2.4x)	3.2x
Brushing (Our Approach)	3.2 ms	1.2 ms	1.9 ms	1.1 ms	30.3 ms	11.4 ms	33.6 ms	10.7 ms	1.04 ms	0.47 ms



Scatterplot Angular Hist./Opacity PCP Our approach

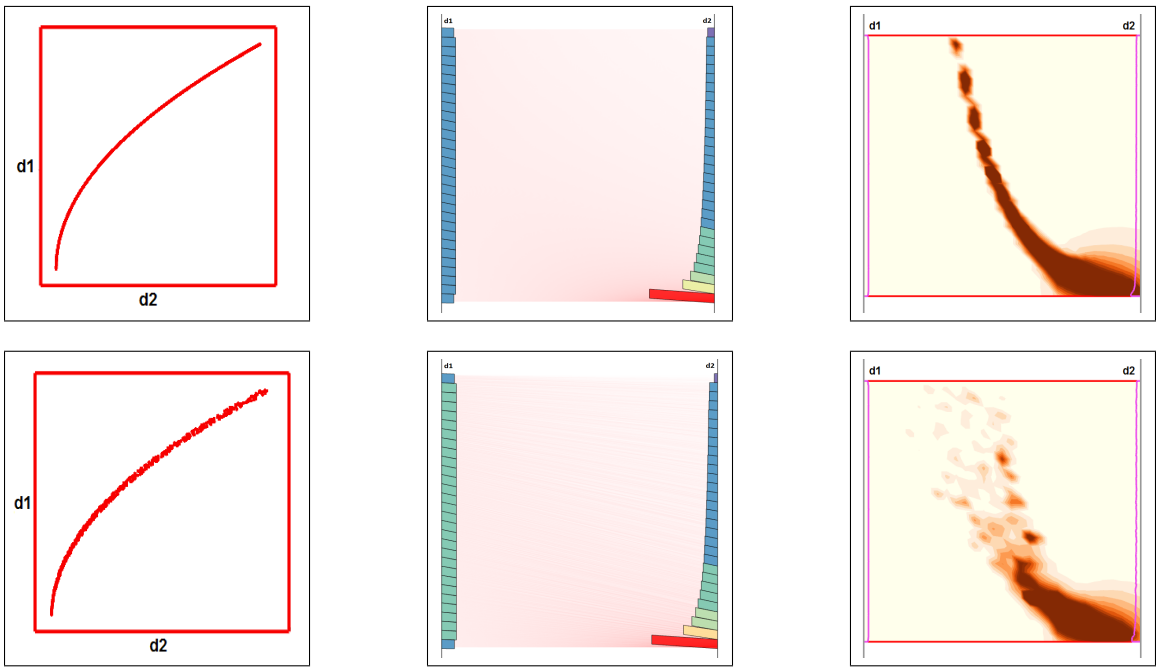
(a) Positive linear relationship with (bottom) and without (top) noise.



Scatterplot Angular Hist./Opacity PCP Our approach

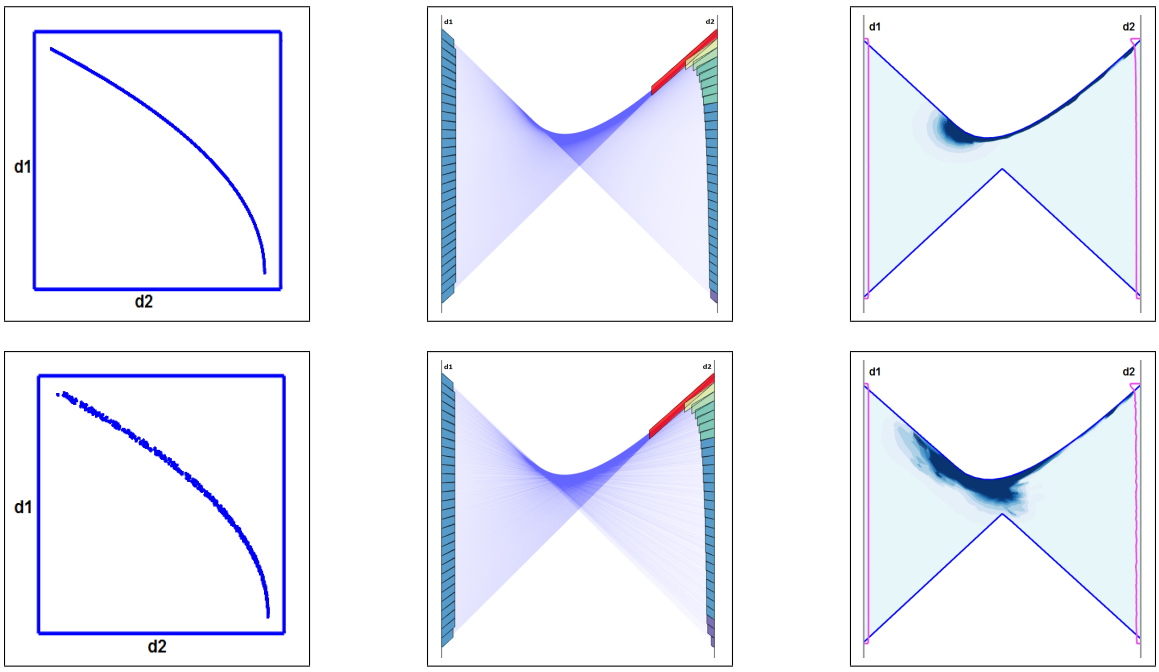
(b) Negative linear relationship with (bottom) and without (top) noise.

Figure 5.3: Positive and negative linear relationships.



Scatterplot Angular Hist./Opacity PCP Our approach

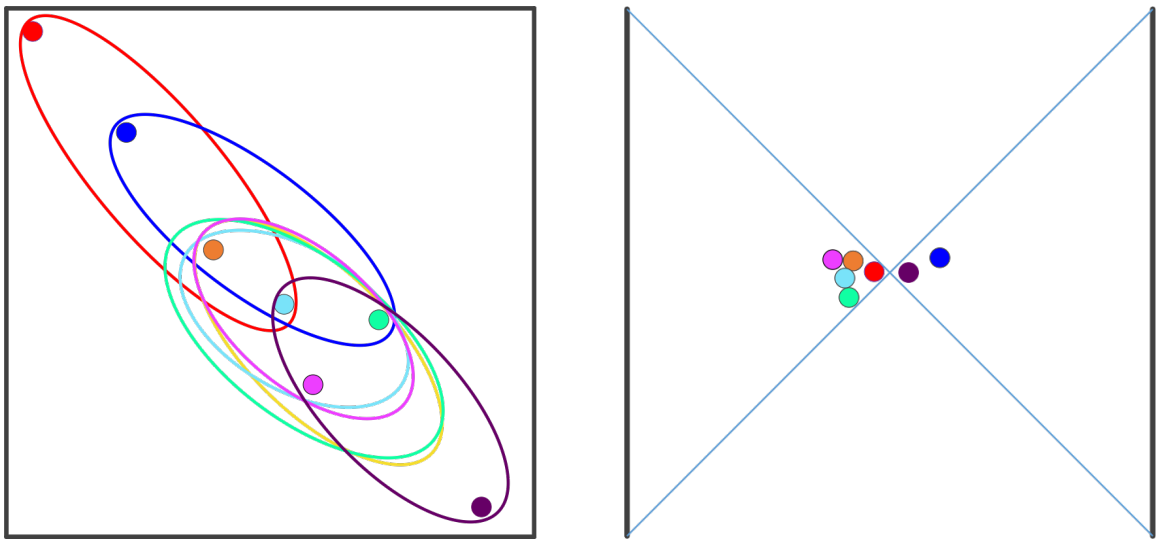
(a) Positive quadratic relationship with (bottom), without (top) noise.



Scatterplot Angular Hist./Opacity PCP Our approach

(b) Negative quadratic relationship with (bottom), without (top) noise.

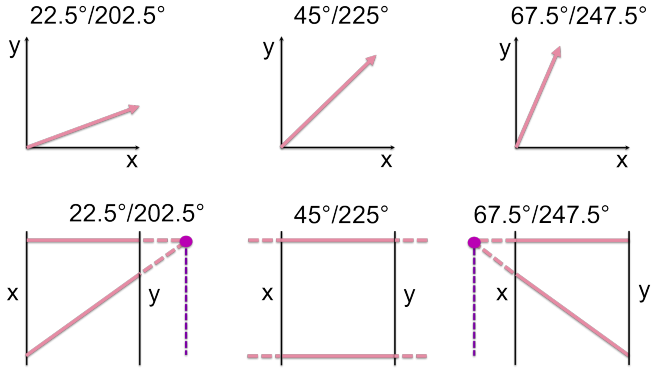
Figure 5.4: Positive and negative quadratic relationships.



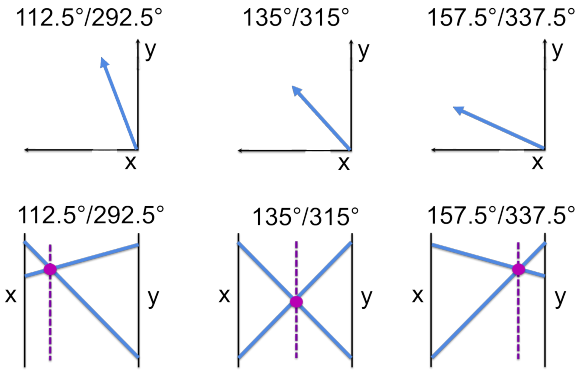
(a) Finding the subsets of k-nearest neighbors (knn)

(b) PCA in PCP domain

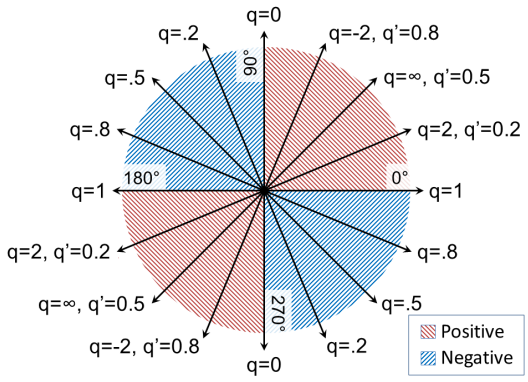
Figure 5.5: Diagram of the transformation from Cartesian domain to PCP domain by: (a) finding the subsets (using knn algorithm) and using PCA to find vectors of subsets; and then (b) mapping those subsets to points in the PCP domain.



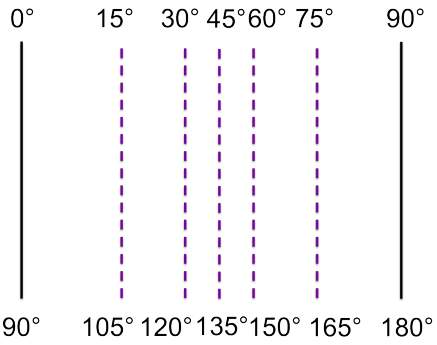
(a) Examples of positive relationships with Cartesian coordinates (top) and PCPs (bottom)



(b) Examples of negative relationships with Cartesian coordinates (top) and PCPs (bottom)

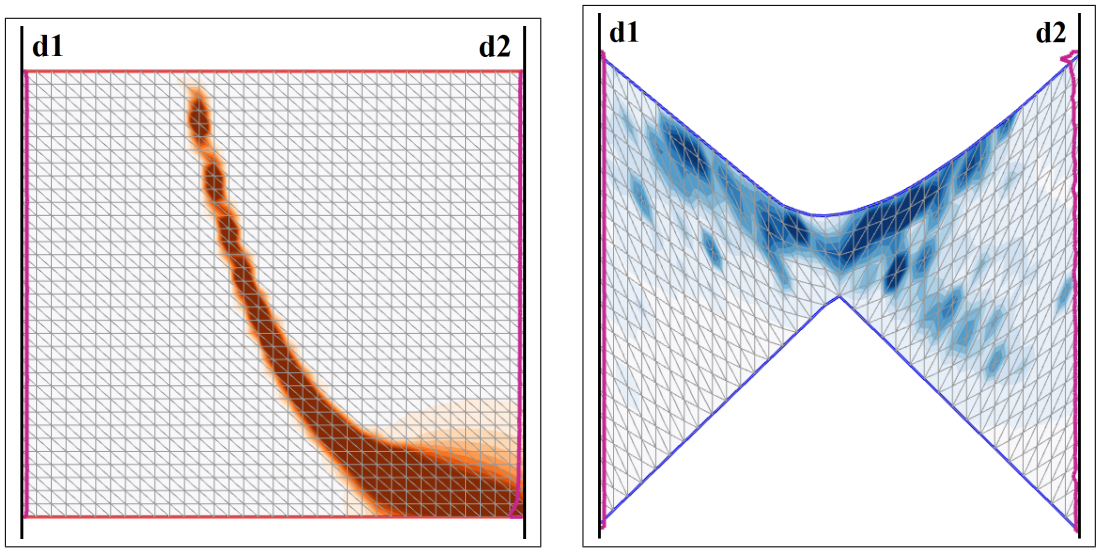


(c) Mapping of lines into PCPs

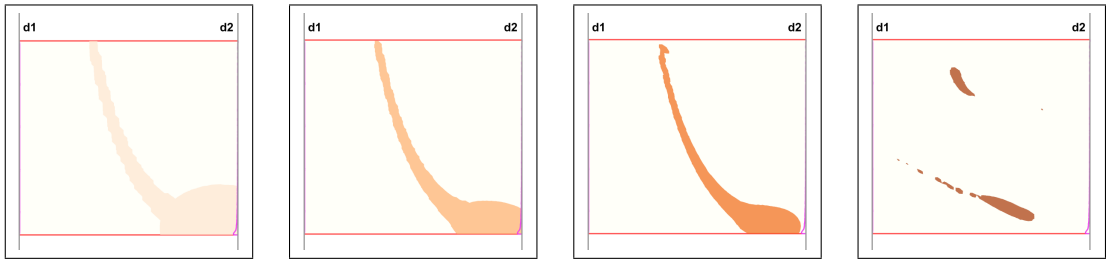


(d) Transformation

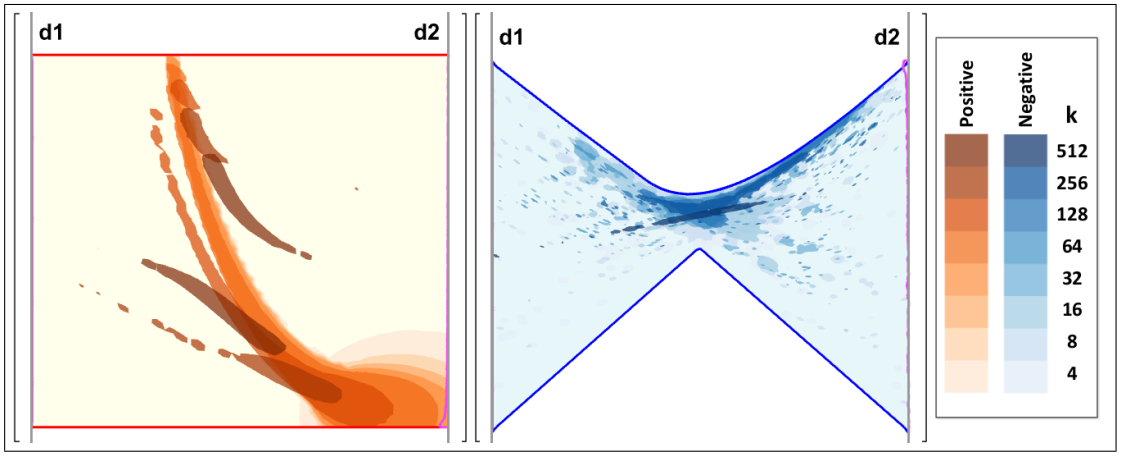
Figure 5.6: Transformation from Cartesian domain to PCP domain. (a) Mapping positive relations (red) from Cartesian coordinates to PCP does not result in a valid intersection (i.e., the intersection is outside of the PCP domain). (b) Mapping negative relationships (blue) from Cartesian coordinates to PCP results in valid intersection locations. (c) By rotating positive relationships 90°, the lines will now cross at valid locations, resulting in q' (orthogonal version of q) for those relationships around the unit circle. (d) The solid vertical lines represent the axes of the PCP, while the dotted lines show the horizontal projection location (q on top and q' on bottom) for a variety of angles.



(a) Histogram for positive (left) and negative (right) relationships.

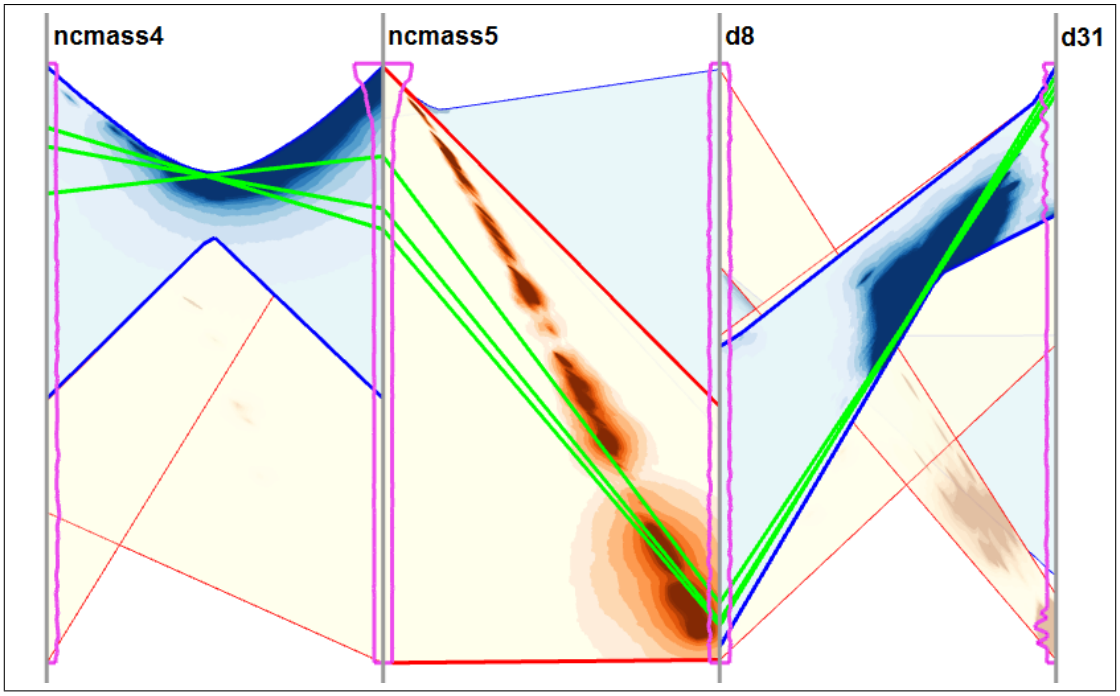


(b) $k = 4, 16, 64, 256$ from left to right.

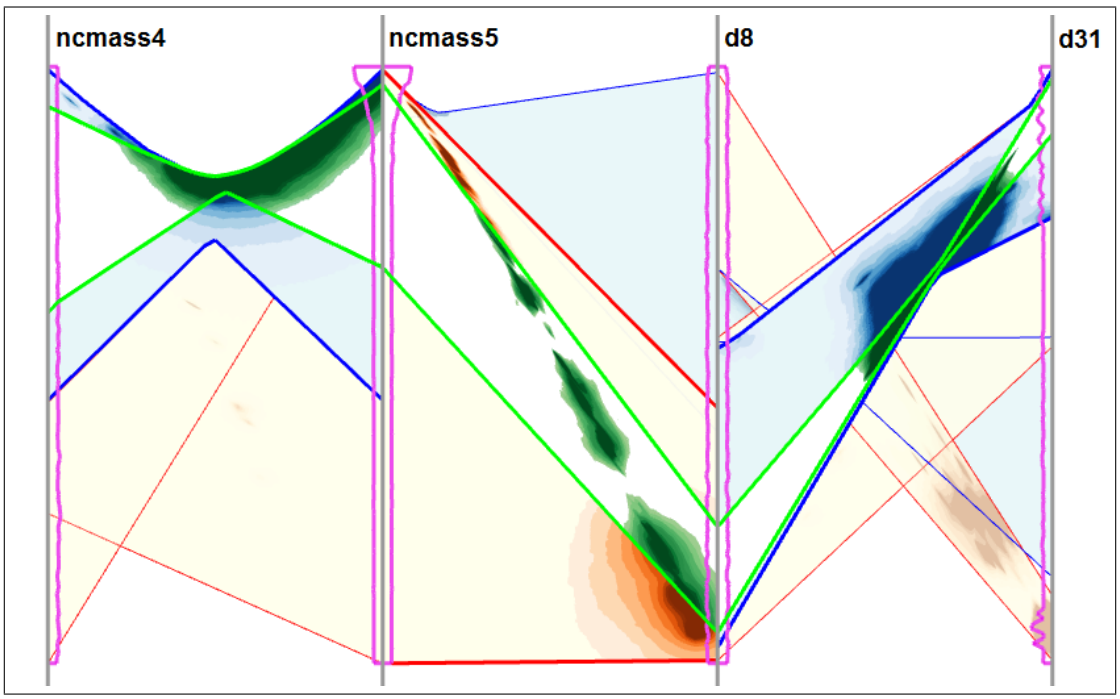


(c) Multiple k for positive (left) and negative (right) relationships.

Figure 5.7: Histogram contours calculated for consistency map.

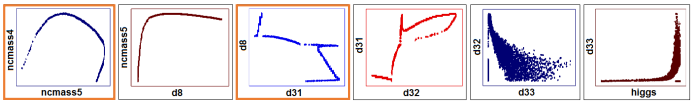


(a) Locating data items

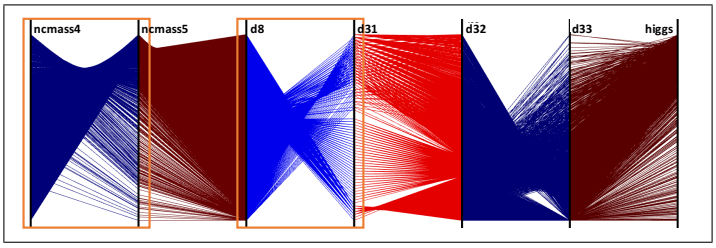


(b) Brushing new clusters

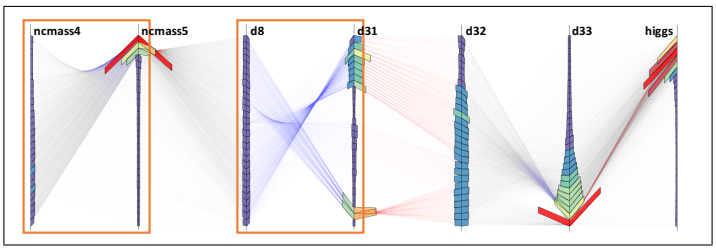
Figure 5.8: Brushing interactions for the *particle* dataset.



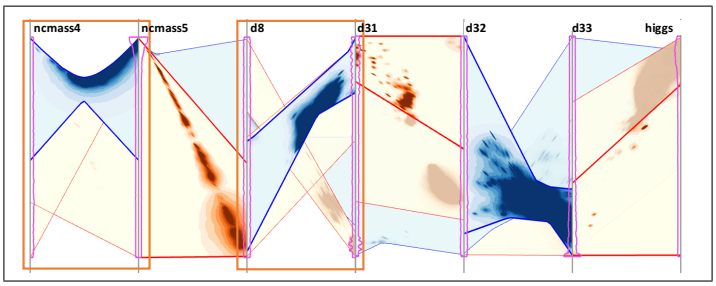
(a) Scatterplot



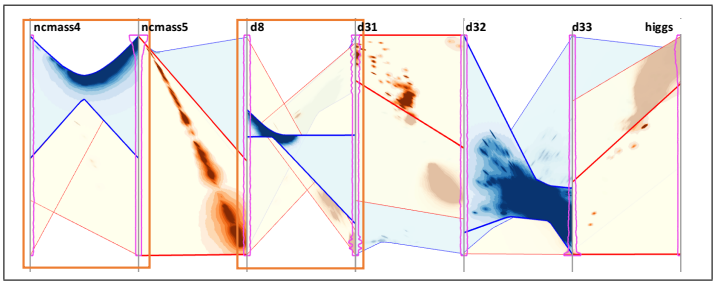
(b) PCP



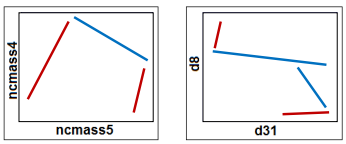
(c) Angular histogram and opacity PCP ($\alpha = 0.003$)



(d) Our approach

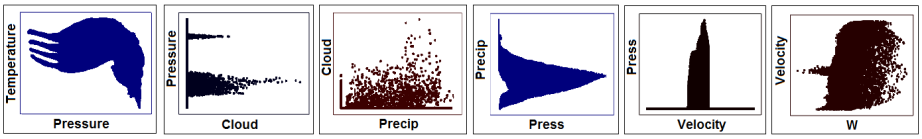


(e) Our approach (select the 2^{nd} negative cluster between *d8* and *d31*)

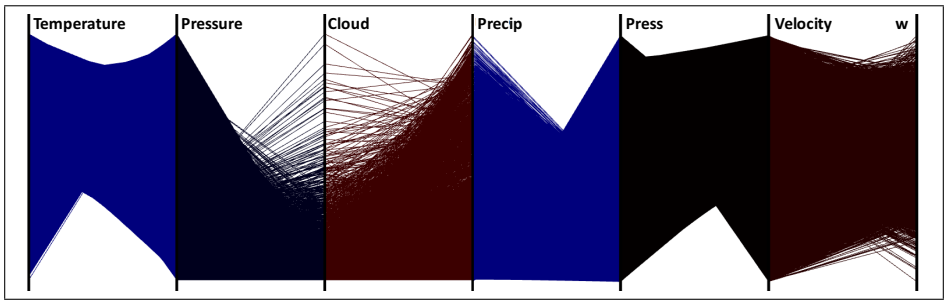


(f) Schematic view of clusters for *ncmass4*/*ncmass5* and *d8*/*d31*.

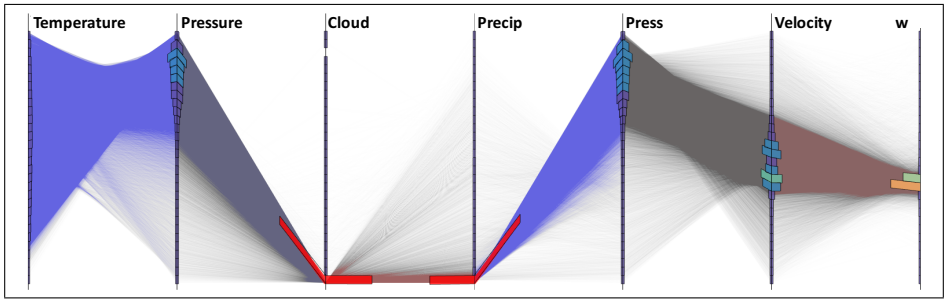
Figure 5.9: Classic PCP, angular histogram, opacity PCP, and our approach for the *particle* dataset. Our approach captures and enables simple investigation of these clusters.



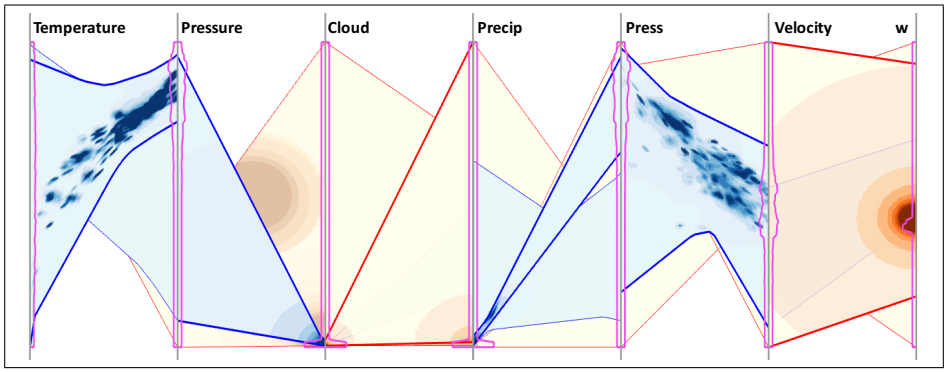
(a) Scatterplot



(b) PCP

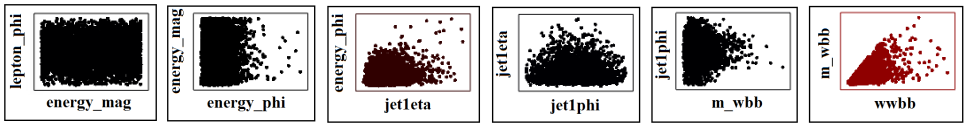


(c) Angular histogram and opacity PCP ($\alpha = 0.003$)

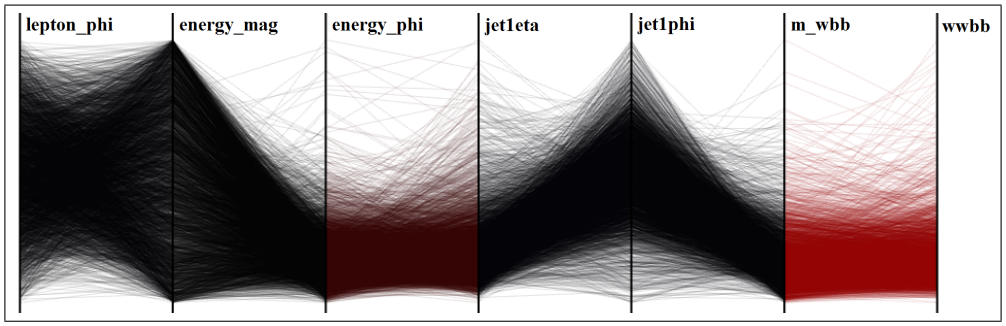


(d) Our approach

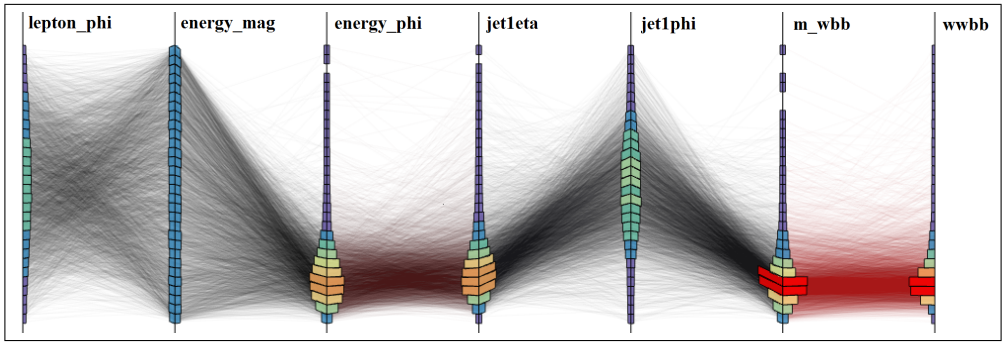
Figure 5.10: Classic PCP, angular histogram, opacity PCP, and our approach for the *Hurricane* dataset containing 10 millions items. Important data patterns can be hidden by the many layers of points in scatterplots or lines in PCPs. On the other hand, details can be lost in the summarizations provided by the angular histogram and opacity PCPs. Our approach balances the need for both overview and details.



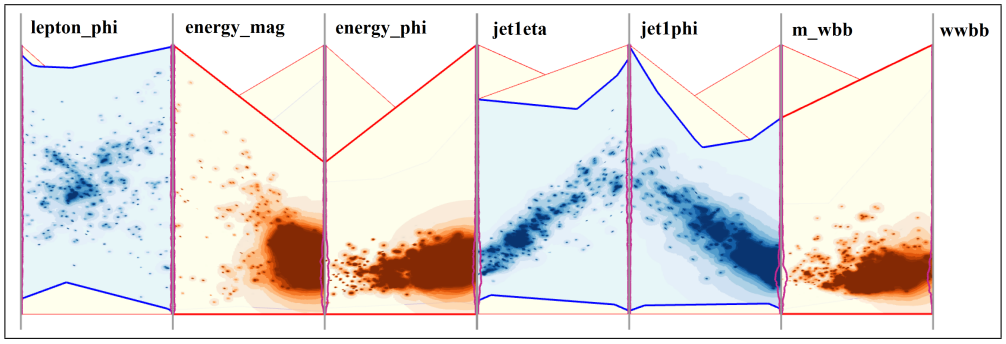
(a) Scatterplot



(b) PCP

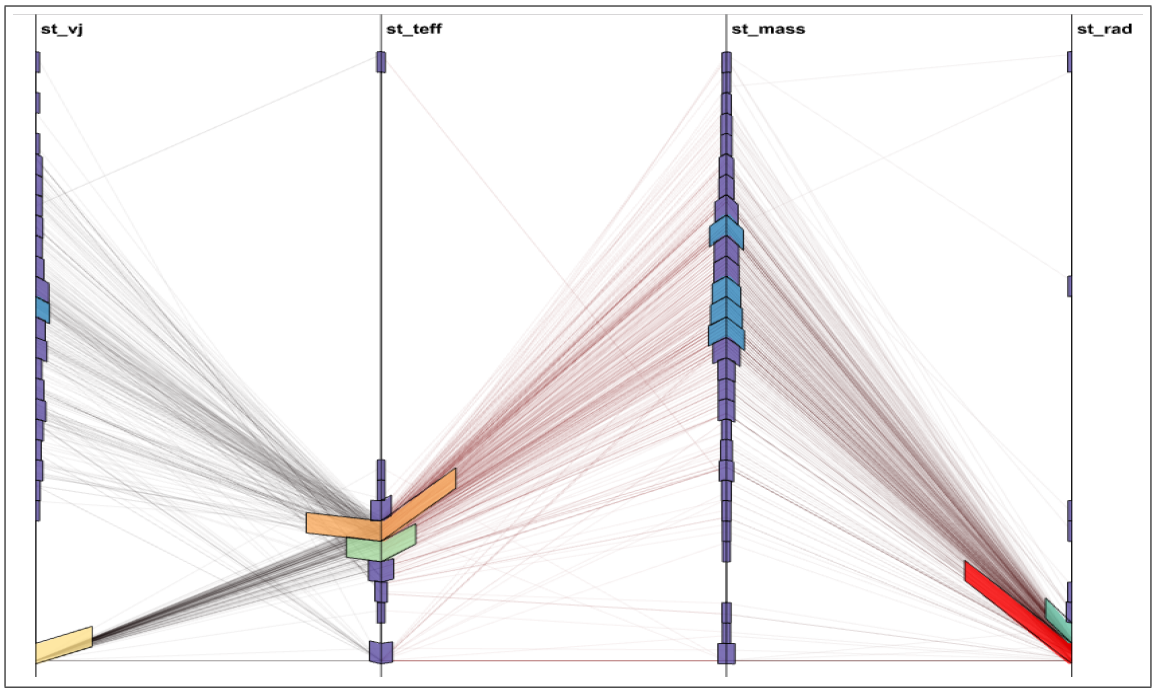


(c) Angular histogram and opacity PCP ($\alpha = 0.003$)

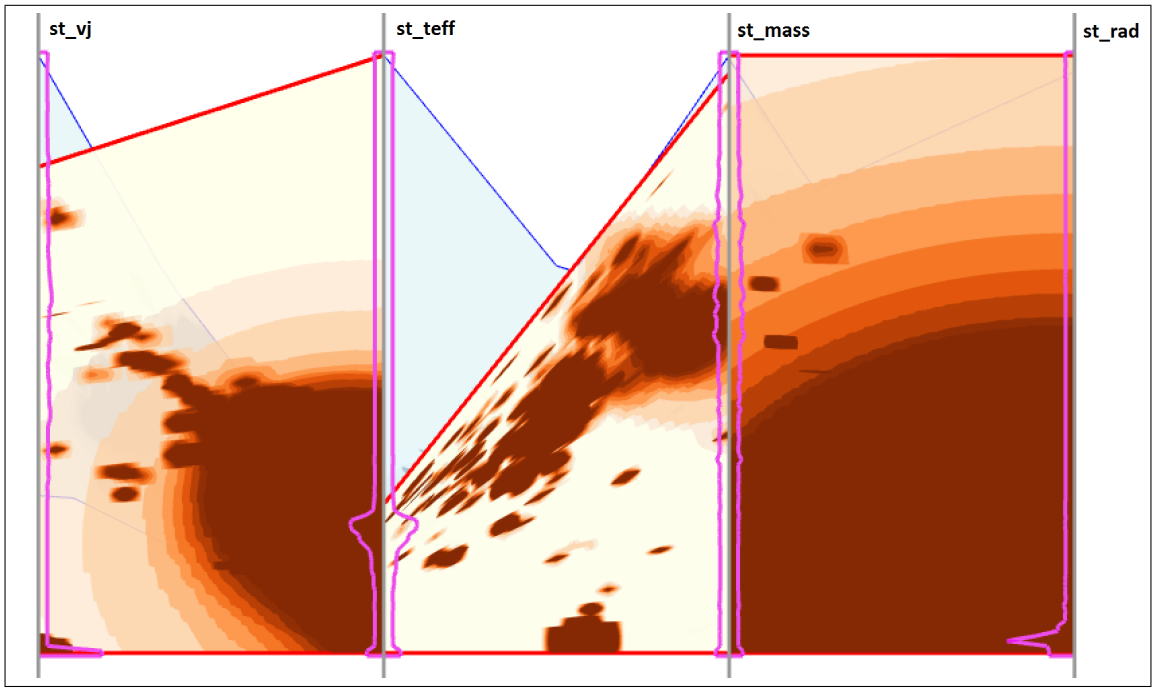


(d) Our approach

Figure 5.11: SCP, angular histogram, opacity PCP, and our approach for the *HIGGS* dataset containing 11 millions data items. Our approach can improve relationship identifying within noise detection. For example, *jet1eta* and *jet1phi* with their -0.107 Pearson correlation coefficient appear almost positive in the conventional PCP, opacity PCP, and angular histogram. However, the weak, noisy negative relationship can be easily spotted using our approach.



(a) Angular histogram and opacity PCP



(b) Our approach

Figure 5.12: Angular histogram and opacity PCP, and our approach for the *planet* dataset containing 1827 data items and 16 dimensions.

CHAPTER 6

CORRELATION VISUALIZATION FOR LARGE AND MULTIDIMENSIONAL DATA

6.1 Challenges

CCP, snowflake, DSPCP, and other existing techniques are not the best layout for multidimensional data when the number of potential relationships is high. Therefore, we developed a multiscale context+focus design (MultiDepViz) that visualizes all possible relationships and highlights the most relevant relationship information from the variables and scalable interactions for large-scale filtering and identification of low-level details as shown in Fig. 6.1.

Visualization of dependencies for multivariate data is a challenge due to the number of potential relationships. For a given dataset of n variables, the number of dependency relationships is $\binom{n}{2}$, $3 * \binom{n}{3}$, and $4 * \binom{n}{4}$ for pairwise, three-way, and four-way relationships, respectively. For a dataset of 20 variables, this is 190 pairwise, 3420 three-way, and 19,380 four-way relationships. A detailed static display showing all potential variable relationships for such multiway dependencies may be overwhelming, confusing, and overly complex. Therefore, we developed a multiscale context+focus design that visualizes all possible relationships and highlights the most relevant relationship information from the variables and scalable interactions for large-scale filtering and identification of low-level details.

6.2 Context View Design

6.2.1 Multiway Dependency Glyph

The first visual encoding mechanism represents the number of potential dependencies for a group of variables using the geometry of an equal numbers of vertices. We first consider the case of two-variable dependency. For these two-variables, the dependency is symmetric (i.e., either variable may be the dependent variable). As such, we represent the relationship with a simple circular glyph as seen in the top left of Fig. 6.2. We next consider the case of three-way dependency. For three variables, there are three potential dependencies because any individual variable may be dependent

upon the other two. Therefore, we represent this relationship as a triangle, such as in the middle left of Fig. 6.2. Finally, when considering a four-way relationship, any of the four variables may be dependent upon the other three. Thus, we represent these as squares, such as the bottom left of Fig. 6.2. We propose these simple visual encodings because they need to be simple so it is easy to be embedded in the context (overview) and also can carry some useful relationship information such as the strength of the relationships.

To help users quickly understand more about the range of interest, we enable users to threshold which R^2 values are important to them. We provide two mechanisms for this. The first method modifies the color of the glyph based upon the average R^2 score of the dependencies. A solid color represents $R^2 = 1$ and white represents $R^2 = 0$. This mode is useful when glyphs are small on the screen. The second mode places actual probability data points in a rectangle for four-way dependency and in a triangle for two- or three-way dependencies in each glyph. Some interactive mechanisms provide ordering, filtering, and selection to help users quickly and efficiently perform multiple statistics tasks.

We next consider a collection of four variables. To represent all potential relationships among these variables, we can composite the three glyphs from Fig. 6.2 (left) into the glyph now seen in Fig. 6.2 (right). We provide a detailed view of the selected dimension as in Section 6.3 that enables further analysis into the exact nature of the relationships.

6.2.2 Overview of Multiway Dependencies

6.2.2.1 Dependency Glyphs

Our composite glyph-based visual encoding effectively summarizes 22 potential relationships among four variables. However, most interesting multivariate datasets have many more than four variables. To handle larger numbers of variables, we now represent every combination of the four variables with its own glyph. Each glyph is then placed in a 2D layout based upon a flexible set of ordering mechanisms. Users also have the option to select between showing the composite glyph, such as in Fig. 6.3a, or the individual relationship glyphs, such as in Fig. 6.3b. To maintain visual context, this splitting and merging operation is handled through animation, as are all other cases of glyph movement.

6.2.2.2 Ordering

The glyphs are laid out such that both the x- and y-axis are controlled independently. For each axis, the user may select metrics including dimension sorting, R^2_{min}, R^2_{max}, R^2_{avg} and time (for time series data). The dimension sorting places the glyphs via ordered permutations. R^2_{min}, R^2_{max}, and R^2_{avg} are all calculated by performing the associated operation for all the dependencies represented by the glyph.

6.2.2.3 Filtering

We provide an upper and lower threshold filtering mechanism for R^2 of glyphs, such that users can reduce the volume of data visualized and find meaningful relationship glyphs. When users want to identify dependency glyphs that have many stronger dependencies, they raise the lower threshold. Similarly, when users want to identify independent variables, they decrease the upper threshold.

6.2.2.4 Navigation

Users can zoom and translate the projection space to navigate and explore variable dependencies. The size of the glyphs changes based upon the number of glyphs visible. When more than $1,000,000$ glyphs are on the screen, each appears as a point. When more than 100 glyphs are visible, they appear as the basic composite glyph. Finally, when fewer than 100 glyphs are visible, the systems show the qualitative glyphs.

6.2.2.5 Selection

We provide three selection mechanisms. The first allows users to create a selection box around a region of interest, and the associated zoom and translate are updated. The second mechanism allows users to select the interested variable and all related variables with that selected variable. The final mechanism selects an individual glyph. Once selected, that set of relationship is highlighted in the detailed view, discussed in the next section.

6.3 Focus View Design

6.3.1 Visual Encoding Design

UnTangle Map represents two- and three-way dependencies with easily understood relationship information. Therefore, we use UnTangle Map to represent two- and three-way dependencies and extend this version for four-way dependencies. UnTangle Map is based on a ternary plot as shown

in Fig. 6.4a. This is a barycentric plot of three dimensions in which each vertex is one dimension. The three variables sum to one, and any given point on the triangle indicates the probability ratios of three variables.

For examples, three dimensions $D0$, $D1$, and $D2$ (Fig. 6.4a) are vertices of the triangle. The item i (blue point) is associated with $D0$, $D1$, and $D2$ with probabilities 0.25, 0.5, and 0.25. This probability data item is strongly associated with dimension $D1$ because its probability is greater than the others. This feature is presented as a point on the perpendicular direction of edge $D0$, $D2$ and it is close to $D1$. This plot makes the position of the probability data items presented as the relative probability with the three dimensions.

We extend this idea to represent the relationship of the four dimensions as shown in Fig. 6.4b. The blue square represents the four-way dependencies with the names of the four dependent variables at the corners of the square. Four lines in the square have different colors, and four points have the same colors with these four lines representing the relationship between two points in each line. For example, the probability of a data point i has a probability for D0, D1, D2, and D3 of 0.7, 0.1, 0.1, 0.1, respectively.

Lines (D0, D1), (D1, D2), (D2, D3), and (D3, D0) are red, blue, green, and purple, respectively. The position of a red point will be defined by triangle mapping coordinates between three points D0, D1, and A. A is the middle point of the opposite line with (D0, D1) (line (D2, D3)). The red point is near D0 and far from D1 as the mapping shown in Fig. 6.4b. In this case, D0 is dominant in the relationship with the other three dimensions.

6.3.2 Visual Patterns

The new visual encoding designs bring insight to help users perform multiple statistical tasks efficiently. Some meaningful visual patterns can be seen in the four examples in Fig. 6.5.

Firstly, when the dimensions are nondominant, the probability data points are in the center of all four triangles as shown in Fig. 6.5a and they also are in the center of a square. The square of the four-way dependencies can more easily point out the nondominant feature than a set of four triangles. In Fig. 6.5b, dimension S0 is more dominant than other dimensions because the probability points are focused on the vertex S0. Additionally, in bidominant relationship, the probability points are the focus of the middle between two vertices as in Fig. 6.5c. These can be seen in both the triangle mesh and relationship square. Finally, the balance relationship between the four dimensions is shown in

Fig. 6.5d.

6.4 Evaluation

To evaluate our approach, we apply our method to four datasets: a product marketing dataset with 47 variables, a particle physics dataset with 66 variables, the National Health and Aging Trends Study (NHATS) dataset with 60 variables, and the Hurricane Isabel dataset with 13 variables over 48 timesteps.

6.4.1 Performance

We build our software using Processing. We have run our experiments on a variety of desktop and laptop systems running Linux, MAC OSX, and Windows.

The visualization rendering itself is interactive. Assume that the dataset has n variables and each variable has k data points. For the overview, our visualization represents $\binom{n}{2} + 3 * \binom{n}{3} + 4 * \binom{n}{4}$ multiway dependencies through $\binom{n}{4}$ glyphs. We have tested our approach up to $n = 624$, and the system has maintained its interactivity. Rendering the detailed view is dependent upon the number of data points. Each point needs to be rendered eight times: four times for the three-way UnTangle map and four times for our four-way UnTangle map extension. Therefore, the total number of points rendered is $8k$.

The main computational challenge is the precomputation needed for determining dependencies, in particular, the pairwise correlation coefficients. Computing Pearson correlation coefficients and Spearman Rank correlation coefficients takes $\mathrm{O}\left(n^2k\right)$ and $\mathrm{O}\left(n^2klog(k)\right)$, respectively. Computing the Coefficient of Determination for Multiple Correlation, R^2 for 2-, 3-, and 4-way dependency takes $\mathrm{O}\left(n^2\right)$, $\mathrm{O}\left(n^3\right)$, and $\mathrm{O}\left(n^4\right)$, respectively. Therefore, this approach has an aggregate computing time of $\mathrm{O}\left(n^2k+n^4\right)$ or $\mathrm{O}\left(n^2klogk+n^4\right)$. In general, $k \gg n$, leading to the pairwise computation being the bottleneck. Fortunately, much of the computation is embarrassingly parallel, and is parallelized in our implementation.

The precomputation of the Hurricane data (624 dimensions and 10 millions data items of each dimension) was 5 hours on a single workstation. This precomputation only needs to be done once, as interactions only require updating the drawing. Our method requires greater precomputation time than UnTangle Map, but it is parallelizable to reduce the total wall-time. The rendering time of our approach is less than that of UnTangle Map in the overview and is a similar rendering time in the

detailed view.

6.4.2 Marketing Research Case Study

Marketing research data, often collected via surveys, is used to identify groups of individuals who might best be served by a particular product design. In this case, we use the Pacific Brands/Berlei Bras case study data, which is commonly used in business school marketing courses. Marketing researchers divide their questions into two types. First, segmentation variables, such as age, sex, income, etc., are used to differentiate groups of people (i.e., independent variables). Second are discriminant (i.e., dependent) variables, which are qualitative, such as feelings about color, texture, etc.

This dataset contains 21 segmentation variables and 26 discrimination variables, that is, a total of 47 variables with $1,081$ 2-way dependencies, $48,645$ 3-way dependencies, and $713,460$ 4-way dependencies. This requires $178,365$ glyphs to represent all multiway dependencies.

First, after loading the data into the system, the overview is shown (Fig. 6.1) with the Pearson correlation coefficient. Optionally, the Spearman Rank correlation coefficient can be selected (Fig. 6.6a) when a nonparametric view of dependency is more appropriate. To understand two- and three- and four-way dependencies separately, these options are selected and animations are used to highlight their transitions into new positions (Fig. 6.6b). The order of points can be modified along the x-axis (Fig. 6.6c), y-axis (Fig. 6.6d), or both (Fig. 6.1c). In these cases, the x-axis is switched to R^2_{max} and the y-axis is switched to R^2_{avg}, with animation connecting the transitions. It is clear from many of these views that most dependencies are weak. A filter on $R^2 \in [0.6, 1.0]$ significantly reduces the number of relationships to explore (Fig. 6.1d).

After some navigation and exploration, a smaller number of glyphs occupy the screen to highlight their corresponding relationships (Fig. 6.1e), which helps to quickly identify the strengths and directions among the relationships.

In the detailed view in Fig. 6.6f, the selected glyph contains variables $S0, D9, D10$, and $D11$:

- *S0*: I am very conscious of bras as fashion objects.
- *D9*: I like to shop in the same lingerie stores as my friends.
- *D10*: I use other people as a source of information for purchase decisions.
- *D11*: I use magazines or newspapers as a source of information for purchase decisions.

Fig. 6.6f shows that some data points move toward *S0* but most of the data points are in the middle. There is no point towards *D10*, which indicates that *S0* is weakly dominant in the four-way relationship. Fig. 6.6f also shows that there are no points around *D10* in three-way and four-way dependencies. *D10* is less dependent upon other variables (*S0*, *D9*, *D11*). This shows that to design bras as fashion objects, information of friends' shopping destination, and magazines or newspapers are a good source of information, since *S0*, *D9*, *D11* are highly correlated. This previously unknown combination of opinions helps to quickly identify groups of individuals who are best served by a particular product design. The result might lead marketers to choose a particular design or advertising campaign.

6.4.3 Particle Physics Case Study

The physics dataset represents a parameter space search in simulations that model subatomic particles under the supersymmetric extension of the Standard Model. The data has 25 input and 41 output variables with $4,000$ items for each variable, which leads to $2.8M$ 4-way dependencies, $137k$ 3-way dependencies, and $2,145$ pairwise dependencies, for a total of $3M$ dependencies. We require $720k$ glyphs to represent all these relationships.

Determining dependency can be valuable in reducing the size of a parameter search space by linking input and output variables together. Many glyphs visible near the top of the overview coordinates in Fig. 6.7a show that the variables of the physics data have strong dependencies. Users can confirm that this is a combination of two- and three- and four-way dependencies by separating the glyphs in Fig. 6.7b. The overview of composite glyphs and 2-, 3-, and 4-way separated glyphs help us understand that these data have many dominant and strong relationships, since many glyphs are on the top of the plot.

These variables are input and output variables of the simulation. The expert would like to understand which inputs are correlated with which outputs. The expert is also interested in which inputs most strongly reflect linear correlation with a given output. With our tool, the expert can easily interact with various dependencies and perform the analysis tasks efficiently.

The expert can quickly select an interesting input/output variable, and the layout will automatically show variables that are correlated to the selected variable. For example, the detailed view of the selected glyph in Fig. 6.7c enables the expert to quickly identify the dependencies from selected variables (including input *I5*, *I6*, *I7*, and output *O12*). This shows that variable *O12* is highly

correlated with others, and it is dominant.

Using UnTangle Map alone to answer the above questions would have required adding many dimensions to the layout and exploring one by one which inputs and outputs are correlated. However, by using our proposed visualization approach, the expert can quickly select the interesting input/output in the data, filter the layout, and show only correlated dimensions.

6.4.4 National Health and Aging Trends Study (NHATS)

The National Health and Aging Trends Study (NHATS) includes data collection research being conducted by Johns Hopkins Bloomberg School of Public Health. The goal is to "foster research that will guide efforts to reduce disability, maximize health and independent functioning, and enhance quality of life at older ages". NHATS collects detailed information on activities and quality of life for a sample of Medicare beneficiaries over 65.

We explore a subset of the NHATS data that has 60 variables with $38k$ items. These data have $2M$ 4-way dependencies, $100k$ 3-way dependencies, and $1,770$ pairwise dependencies, for a total of $2.15M$ dependencies. We require $487k$ glyphs to represent these relationships.

Fig. 6.8 shows an example analysis of the NHATS data. Fig. 6.8a and 6.8b show the overview with composite and split glyphs for the data. It is immediately apparent that many relationships have low R^2 values, while a few have high max R^2. Using the lasso tool (Fig. 6.8c) filters data down to a subset (Fig. 6.8d).

After exploration, a specific relationship is investigated. Fig. 6.8e shows the detailed view of variables *d45*, *d46*, *d47*, and *d48*, the four possible cases of the question "Is [Caretaker Name] paid by you (*d45*), your/his/her family, by a government program (*d46*), or by your/his/her insurance (*d47*) or other (*d48*)?". The centrality of these points in the square shows that these four variables have a nondominant (uncorrelated) relationship. This makes sense, as the four options should be mutually exclusive cases of payment.

6.4.5 Hurricane Data Case Study

Finally, we explore the IEEE Visualization 2004 Hurricane Isabel contest dataset. It consists of 48 timesteps, measuring 13 variables with a spatial resolution of $500 \times 500 \times 100$ ($25M$ points per timestep). Combining all variables over all timesteps leads to an exploration of 624 total variables (i.e., 13 variables x 48 timesteps). These data have $25B$ four-way dependencies, $250M$ three-way dependencies, and $194k$ 2-way dependencies, requiring $6B$ glyphs.

Figure 6.9 shows an example analysis. Figure 6.9a shows an overview of all dependency features of the 624 variables. The many points at the bottom of the chart show weak dependencies, yet patterns of strong dependencies are still visible. For example, a repeated pattern between *QRAIN* and *QSNOW* variables is seen in the middle of chart. To investigate further, the view is filtered by selecting the *QRAIN*/*QSNOW* variables (Figure 6.9b). Further zooming onto *QRAIN*/*QSNOW* in Figure 6.9c shows more detailed glyphs that can be individually inspected.

The relationships can also be sorted by time horizontally, by variable name vertically, and filtered by R^2 (Figure 6.9d). Figure 6.9e shows the relationships sorted by time horizontally and R^2 average vertically. Noticing inconsistency in the *CLOUD* variable in Figure 6.9d warrants further investigation. Using a selection box, the *CLOUD* glyphs are isolated in Figure 6.9f. In this view, a number of conclusions can be drawn. For example, this confirms that the relationship between *CLOUD* and *Pressure* (*P*) is not consistent over time. Similarly, the relationship between *CLOUD* with *QGRAUP* is not consistent from timesteps 10 to 20.

With over $25B$ two- and three- and four-way dependencies, the Hurricane Isabel data are large and impractical to explore completely. Our approach enables quickly reducing the variables of interest. Without our approach, the relationship between *CLOUD* and *Pressure* might not be isolated for analysis, but it is clear through our visualization that they are not consistent over time.

6.5 Applications on Large-Scale Code Performance Data

As high performance computing (HPC) codes and systems grow more complex, optimizing the performance of large-scale parallel programs becomes increasingly challenging. Many factors can contribute to less-than-optimal scaling in HPC codes, though one common root cause is imbalance, such as when a small percentage of MPI ranks require more time to complete than the rest.

We propose a new visual data exploration approach that enables rapid identification of imbalance in large-scale HPC code performance data. Our approach centers around the use of combinations of multivariate and multidimensional statistical projections to overcome the challenge of large volumes of performance data. We present a case study that shows the use of these methods to find the imbalance in performance data generated by a complex combustion modeling code, Low Mach number Combustion (LMC).

6.5.1 Statistics Projection Methodology

Within the context of displaying performance data, we are focusing on the idea of *projections* to reduce the amount of data being displayed to the user. By projection, we mean that we are "collapsing" or "consolidating" data from one or more dimensions to reduce the amount of data being displayed. The idea is to visually present several of these high-level projections to a user to provide perspective using multiple views of the data. This approach reflects design principles that have guided efforts like HPCToolkit [68], which advocate for hierarchical, top-down aggregation of performance information.

Since the performance problem we are focusing on is discovering imbalance, then variation in data, as opposed to absolute data value, is of more importance. In that regard, the statistical projection methods that can preserve the meaningful variation feature in data are standard deviation, (σ), Coefficient of Variation (or C_v).

The standard deviation is a measure that indicates how tightly samples are clustered around the mean ($\bar{x}$) in a collection of data. Intuitively, a small σ means the data are clustered tightly around the mean, whereas when large, the data are more dispersed away from the mean.

The Coefficient of Variation (Eq. 6.2) represents the ratio of the standard deviation (Eq. 6.1) to the mean; it is effectively a normalized measure of variation. As such, it is useful for comparing the degree of variation from one data series to another, even if the means and data ranges are drastically different from each other.

$$\sigma = \sqrt{\frac{\sum (x - \bar{x})^2}{n-1}} \tag{6.1}$$

$$C_v = \frac{\sigma}{\bar{x}} \tag{6.2}$$

When doing aggregation, or in other words, creating a reduced resolution version of data, which is a form of projection, use of $\bar{x}$ as the projection operator can have the undesirable side effect of "washing out" the signal we are interested in, namely variation. Instead, C_v is a much better projection operator for preserving the underlying variational signal in the data. Whereas this dissertation focuses on working with code performance data, this variation preserving property applies in a general way to diverse data, such as when looking at long-term changes in precipitation patterns in climate modeling data and in performing data aggregation (coarsening) operations [91].

Let's consider a simple example where we have two independent variables: MPI ranks along one axis and "timesteps" along the other, and the dependent variable is runtime. Most performance visualization tools would consider this type of dataset as a two-dimensional array of values, and then provide visualization techniques so a user can explore this potentially very large array of data, such as using *timelines* (see Borgo et al. 2014 [92] or a heat map for a discussion of such techniques).

An alternative approach would be to project these data in two ways: one projection collapses time, the other collapses MPI rank. When time is collapsed, we have remaining a single value for each MPI rank that reflects some $F(runtime)$ across all timesteps for each MPI rank. When MPI ranks are collapsed, we have remaining a single value at each timestep that reflects some $F(runtime)$ across all MPI ranks at that timestep.

The primary challenge with this type of approach is the potential for loss of meaningful information during the projection process. Our approach to this fundamental challenge is twofold. The first is to rely on the property of C_v as a projection operator to preserve variational signal when doing data aggregation. Use of an alternate unary function, like *maximum*, would produce different results than averaging, but would fail to represent the signal of variation inherent in the data. Similarly, boolean range queries, such as those forming the basis of query-driven visualization methods [93] would be ineffective in finding variation in data, as they focus on finding data that matches a given set of (compound) boolean criteria. The signal presented by variation is something that must be computed: it is not directly identifiable through unary operators or compound range queries.

The second element of our strategy is to present the user with multiple projection views. In other words, building on the example above, where one axis is MPI rank and another is time, if we add a third axis, "function call", then we have the ability to more accurately resolve in high-dimensional performance data those areas where imbalance occurs. The addition of projections from more data dimensions will aid in more finely resolving the source of imbalance.

6.5.2 Design of Performance Data Projections Visualization

For this study, our framework for conducting experiments consists of several discrete processing stages that are as follows and illustrated in Fig. 6.10

- Data source: generate performance data, either from a sample code that produces data having known characteristics or from an actual solver code.

- Data subset selection, summarization, statistics: use a parallel tool to ingest performance data

and perform projections/summaries.

- Data visualization: perform visual display, exploration, of both the summarization and detailed performance data.

Of these, the *data source* and *data subset selection* operations are applications built on top of BoxLib [94, 95], Both sets of applications, *data source* and *data subset selection*, are written in C++ and are capable of running in distributed memory parallel fashion using MPI. For our work here, we are leveraging BoxLib's ability to perform performance data collection and applied this methodology to a Low Mach number Combustion (LMC) code data source as shown in Fig. 6.11.

Our data display component is written in JavaScript and runs in a web browser. It ingests data products produced by *APP*, and presents a GUI to the user for the purpose of interacting with these data products. The data products produced by *APP* that are consumed by our display component are stored in JSON format.

The GUI, shown in Fig. 6.12, consists of four primary display areas.

6.5.2.1 Menu Control View

The user has the ability to selectively enable the display of four different performance metrics (running time, barrier wait time, reduction wait time, number of sent/received messages) and to select from one of several different projections for each performance metric: *mean*, *std*, *coefVar*, and *skew*. Other performance metrics and projections are possible, though they must be computed by *APP*, stored in the resulting JSON file, and then will be detected by the viewer and presented to the user.

6.5.2.2 Context+Focus View

The second major element of the GUI, appearing in the center and the bottom of Fig. 6.12, is the context+focus view. The center of the figure is the focus view, and it contains the charts for the user-selected data created from user-selected parameters.

We provide four different graphing/charting techniques (visual encodings): area graph (Fig. 6.13a), line graph (Fig. 6.13b), stream graph (Fig. 6.13c), and bar graph (Fig. 6.12).

The stream graph, as described by Byron and Wattenberg, 2008 [96], is a generalization of stacked area graphs where the baseline is not fixed. By shifting the baseline, it is possible to

minimize the change in slope (or wiggle) in individual series, thereby making it easier to perceive the thickness of any given layer across the data.

In the bottom of that figure, we see a compact version of the data display (context view). This GUI element allows a user to perform subset selection (zooming technique) from the dataset, and have the subset displayed in the focus display area.

6.5.2.3 Communication View

This view located on the top-right side of Fig. 6.12 shows a heatmap style display to indicate the number of sent/received messages across processors (MPI ranks) and regions.

6.5.2.4 Call Graph View

The Call Graph view is located on the bottom-right side of Fig. 6.12. The call graph shows the nested view of regions and provides contextual insight for the data display that appears in the context+focus view. Specifically, this view helps a user to understand the nested, hierarchical relationships of code regions and performance data.

This study focuses on a methodology and its use for finding imbalance in a large-scale HPC code having nonuniform behavior. The methodology combines statistical methods for multivariate and multidimensional projection with interaction and multiple views of data. Among its strengths is the ability to preserve variational signal in data, across projections and when doing data aggregation/coarsening. As such, this methodology appears promising in terms of scalability to even larger and more complex data. We believe these principles will be of use to other researchers and software applications that focus on studying large-scale HPC code performance data.

6.5.3 Case Study Results

These results are a case study where we show application of a visual data exploration methodology that uses multidimensional and multivariate projections to identify the source(s) of imbalance in a nontrivial code. To be successful, the methods need to lead the developer to the point in the code where imbalance occurs, and to provide some idea of why the imbalance is occurring. Such information is required in order to know where in the code to focus attention, and to know what in the code needs attention and give insight into how to fix it. For example, it could be a communication bottleneck, a computation imbalance, or other factors. This case study will unfold in a way that shows the process of investigation to find the source and nature of imbalance by using

our approach.

The code in this case study, LMC, is nontrivial because it is an AMR-based code that focuses calculations in areas where there is interesting physics happening. In the case of an evolving flame front, what may start out as a small, compact flame and relatively uniform computation will quickly become quite complex as the flame expands and the refined computations move about in the domain. This dynamic behavior will be reflected in terms of varying computational load, and varying communication load and patterns. The result of this nonuniform computation and communication leads to poor balancing characteristics.

For this case study, we are working with a collection of LMC code performance data that is about 1 TB in size, collected from a 1200-way run. The performance data contains a wealth of information at varying levels of granularity, including both function level and at the level of user-defined code regions, as well as information about the 138 million reductions, 26 thousand barriers, and 2.3 billions of data sends between ranks during the run. Identifying imbalance in this large performance data is challenging, but our approach will help a developer quickly go through these data and find the imbalance efficiently.

The purpose of the methodology we present here is to aid the developer in understanding how to make the code more efficient and balanced, which could include grid size, placement, and different grid movement strategies.

6.5.3.1 Context View: Code Performance Data

A developer typically starts with an overview of code performance data. Fig. 6.14b shows a typical overview, where the horizontal axis is time, and the vertical axis is MPI rank. Cells in this chart are colored in a way that indicates which MPI function is executing at some point in time, along with an indication of non-MPI code execution.

In Fig. 6.14b, what a developer wants to see is that all MPI ranks are executing the same thing at about the same time. That condition would appear as uniformity along the vertical axis for some point in time. Instead, this image shows a good deal of vertical heterogeneity. That image, which was produced by BoxLib's *amrvis* tool, reflects a subsampling strategy that displays data from every *N*'th MPI rank as way to generate a coarse resolution view. This view is a subset in time, as there are many more samples in time than can be displayed at full temporal resolution on a monitor.

6.5.3.2 Context View: Code Region Projection

Using our visual exploration approach to find imbalance in the code, the developer will begin with multiple context views that convey imbalance, then formulate a hypothesis about the source and nature of the imbalance. These context views provide the multiple views design that helps developers quickly see and explore vast amounts of information, and facilitate deeper drill down into finer resolution views of performance data.

We begin with an image that conveys information about the code to aid in situational awareness. The nested nature of the regions is shown in call graph form (tree view and nested view) in Fig. 6.15. This figure shows the nested relationship between the code regions, along with some indication of elapsed time consumed by each code region.

One might be tempted to simply look at a function call projection that shows per function runtime across all MPI ranks and across all timesteps. Fig. 6.14a shows such a chart. There, functions are arranged alphabetically along the horizontal axis, and the vertical axis is $\bar{x}$ execution time for each of these functions, where $\bar{x}$ is the projection operator across all dimensions. From this image, one might inclined to focus on the function *CollectData Alltoall* since it has the highest average execution time.

However, $\bar{x}$ as a projection and data reduction operator by itself is somewhat misleading in terms of identifying balance. Fig. 6.14b shows a function call projection that uses C_v, which shows variation in runtime for each function across all MPI ranks and timesteps. Contrast the three longest executing functions in Fig. 6.14a with those showing the most variation in runtime in Fig. 6.14b. We see that the function with the highest average runtime, *CollectData Alltoall*, has very low variation. Therefore, this function is likely not the source of imbalance.

Further complicating matters is the fact that functions may be invoked from multiple places in the code. For example, a reduce operator may be invoked from multiple locations. When invoked from some code regions, it may exhibit reasonable balance, while when invoked from others, it may be acutely out of balance. The reason for this variation is likely problem specific. While a function projection, such as Fig. 6.14 is informative, it is insufficient by itself to provide the developer with sufficient information to optimize the code.

6.5.3.3 Context View: Rank Projections

An alternative way to do a high-level view of "all" the code's performance data is to perform rank projections. In a rank projection, we present information about each MPI rank's performance across all code regions, all timesteps, and so forth. The intention here is to determine, at a glance, the degree to which there is imbalance across MPI ranks. Figs. 6.16 and 6.17 show per rank projections using the $\bar{x}$ and C_v operators, respectively, of execution time (red), reduction wait time (blue), and barrier wait time (green). There are three primary points to consider about these figures.

The point is what these images convey in terms of understanding about the code's performance. Clearly, both Figs. 6.16 and 6.17 show imbalance in performance, but they each provide a different view about the imbalance. Fig. 6.16, particularly Fig. 6.16b, shows MPI rank average execution time to range between about 70 and 120. In contrast, Fig. 6.17, particularly Fig. 6.17b, shows what appears to be an even higher degree of variation. Arguably, Fig. 6.17 presents a more pronounced view of the imbalance in the code.

The second point concerns the use of $\bar{x}$ and C_v as alternative projection operators. The difference between Figs. 6.16b and 6.17b called out in the previous paragraph reflects this point. Use of $\bar{x}$ as a projection operator can hide variation, as low and high values are averaged. In contrast, C_v reflects this variation in a direct way.

The third point concerns effective presentation of large amounts of data. To create Fig. 6.16b, which is a coarse resolution view of the data in Fig. 6.16a, we combine 25 MPI ranks' worth of data into one data value using averaging, effectively creating a *collapsed* or coarsened view. In contrast, we create Fig. 6.17b by *collapsing* 25 MPI ranks' worth of data using the C_v operator to create Fig. 6.17b. As in the previous paragraph, coarsening large data using $\bar{x}$ can hide variation, whereas coarsening using C_v preserves the variation. This characteristic is consistent with results from applying this operator to other types of data [91].

If we are looking for a particular MPI rank as being the source of trouble, we would look for a correlation between high execution time and high variation. However, in these images, we see no such correlation. In Fig. 6.16, we see a few ranks that have execution time higher than the others, but a different set of MPI ranks shows up in Fig. 6.17 as exhibiting high variation in running time. While useful as an overview, this projection doesn't seem to yield a positive result in our search for imbalance in this code. However, this rank projection is a valuable view to have early in an investigation, and could prove useful in other circumstances.

6.5.3.4 Context View: Region Projections

Next, we do a per-region projection, shown in Fig. 6.18, which shows per-region performance—running time (red), reduction time (blue), and time spent waiting in barriers (green) — across all MPI ranks and all timesteps. Figs. 6.18a and Fig. 6.18b use $\bar{x}$ and C_v projection operators, respectively. As in the case of Fig. 6.14, we are interested in understanding the relationship between high average runtime and variation.

Fig. 6.18a shows that `TS0`, `TS1`, and `Aladv` take longer to execute than other regions. Fig. 6.18b shows the variation of running time of those regions. A developer would look back to the call graph in Fig. 6.15 to understand the nested region relationships. Because `TS0` and `TS1` contain `Aladv`, and all have high $\bar{x}$ and C_v, the developer will focus their attention on `Aladv`. The next question is what is the source of imbalance in that particular region.

Next, we look more closely at the performance variable *waiting time* (barrier waiting time and reduction time). Since LMC has 138 million reductions, but has only 26 thousand barriers, reduction time will be considered. Fig. 6.19 shows a per-region projection, across all MPI ranks, of reduction time using $\bar{x}$ (Fig. 6.19a) and C_v (Fig. 6.19b) as the projection operators, respectively. This figure confirms that the code region `Aladv` has high reduction time in terms of both $\bar{x}$ and C_v, and provides a clue as to the source and location of imbalance in this code.

Interestingly, we also see in Fig. 6.19 that the region `Lsync` has both high $\bar{x}$ and C_v barrier wait time. Referring back to Fig. 6.15, we see that `Lsync` is not part of `Aladv`. However, per Fig. 6.18, `Lsync` has relatively low runtime $\bar{x}$ and C_v, so it is likely of less interest as a major source of imbalance.

6.5.3.5 Detailed View: Function Projection in the Region

At this point, we are focusing our attention on the `Aladv` region as the most likely source of imbalance. Our next step is to do a function projection, but only within that region. By focusing on one region, we are excluding things we're not interested in, for example initialization, file I/O, etc. And, more importantly, we are limiting the scope of view of performance data, particularly per-function information, only to the region of interest, since it may be possible that a single function that exhibits imbalanced performance in one place exhibits balanced behavior elsewhere.

Fig. 6.20 is a function projection showing $\bar{x}$ execution time (Fig. 6.20a) and variation in execution time (Fig. 6.20b) for all function calls within the `Aladv` region. We see that functions

`DoNodalProjection()` and `MLsyncProject()` have higher $\bar{x}$ and C_v execution time than others. The hypothesis here is one of these two functions is the source of imbalance.

In Fig. 6.21b, we see more imbalance in terms of the reduction waiting time (taller blue bars), compared to the execution time. Looking at Fig. 6.22b, we see greater variation in execution time (red) than in waiting time (blue and green), which indicates the computation is out of balance in that function.

The next step is to examine the performance of each of those two functions in more detail. Fig. 6.21 shows per-rank execution time of `MLsyncProject()` in terms of $\bar{x}$ (Fig. 6.21a) and C_v (Fig. 6.21b), while Fig. 6.22 shows the same information but for `DoNodalProjection()`. The developer would probably begin to focus attention on `DoNodalProjection()`, since it has both high $\bar{x}$ execution time as well as a high degree of variation. `DoNodalProjection()` average runtime is about $8\times$ that of `MLsyncProject()`.

6.6 Applications on Large-Scale Climate Data

This case study focuses on exploring how the C_v can reveal features and characteristics that would otherwise not be visible using only $\bar{x}$ or σ in large-scale, complex climate model output. The case study consists of two parts, one focusing on precipitation (§6.6.1) and one focusing on winds (§6.6.2).

We use precipitation and wind speed data generated by the CAM5.1 global atmospheric climate model [97] run at approximately $1^\circ \times 1^\circ$ longitude-latitude resolution under observed boundary conditions from the period 1959-2014 [98]. Output from this run consists of multivariate, four-dimensional data: latitude, longitude, time, ensemble member. The size of precipitation data over 50 runs is 7.4 GBytes and the size of wind speed data is 122 GBytes.

The model was run 50 times with different initial states, thus producing an ensemble of 50 realizations of how the weather might have evolved. While the large number of simulations is unusual, the generation of multiple simulations in this manner is a standard approach for characterizing uncertainty in the climate system. Here we examine monthly mean precipitation output on the model's longitude-latitude grid.

For both precipitation and wind studies, we are using the same general approach: produce different types of projections (spatial, temporal) using different projection operators ($\bar{x}$, σ, C_v), and make observations about the differences in science that emerge from each type of projection.

In both cases, it turns out that C_v is able to reveal specific scientific features that are not present in the other two types of images, suggesting that C_v is quite useful in helping facilitate scientific knowledge discovery.

6.6.1 Precipitation

Precipitation is one of the more visible and influential aspects of the climate system for society and ecological systems, and thus is a frequent topic of analysis. It represents one branch of the planet's hydrological cycle, wherein moisture evaporates over the ocean, is transported over ocean and land, precipitates out of the air, and then (if over land) returns to the ocean through rivers and groundwater.

Because precipitation amounts vary strongly across space (e.g., deserts versus rainforests) and, in some places, across seasons, comparisons often require some form of normalization. A common way of doing this is by dividing by the mean, usually multiplying by 100 to get a percentage deviation from the historical mean. For instance, when generating gridded observational products of precipitation variations, point measurements at weather stations are converted to fractional anomalies, which are then interpolated; after the interpolation, the fractional anomalies are multiplied with a spatially interpolated field of mean precipitation [99]. The C_v is closely related to the calculation of these fractional values.

This case study focusing on precipitation has two lines of exploration: space and time. The key idea in both investigations is that C_v reveals information that is not apparent using either $\bar{x}$ or σ.

6.6.1.1 Spatial Projections

To begin, we compare spatial projections of $\bar{x}$, σ, and C_v, shown in Fig. 6.23. Here, we are projecting climate data from a 4D space (latitude, longitude, time, ensemble member) to a 2D space (latitude, longitude). For each lat/lon point, we compute the projected value as $p = f(T,E)$ across all times T and ensemble members E, where $f \in [\bar{x}, \sigma, C_v]$.

The images of $\bar{x}$ and σ precipitation (Figs 6.23a and 6.23b) show the band of rainfall that straddles the equator, known as the Intertropical Convergence Zone (ITCZ), along with the mid-latitude storm tracks that branch off from the ITCZ from the western sides of the major ocean basins; much less precipitation falls in higher latitude areas where the air is too cold to hold much water. The σ simply shows that areas with large precipitation amounts have freedom for large variability.

The image of C_v (Fig. 6.23c) looks rather different. Generally it is highlighting the deserts in

the subtropical areas to the north and south of the ITCZ. The air that has dried through precipitation while rising in the ITCZ moves poleward and descends here, leading to hot and dry conditions. The low mean precipitation means that the denominator of C_v is small, and the infrequent but substantial storms lead to a comparatively high numerator. The exception to this subtropical focus is the area of higher C_v over the eastern tropical Pacific (i.e., against South America). Because the trade winds blowing from the east pull up cool water from the deep ocean here, the water at the surface is usually quite cool, does not evaporate much, and thus does not provide much moisture for subsequent rainfall. However, during El Niño years, the winds reverse and temperature rises markedly, driving major thunderstorms.

6.6.1.2 Temporal Projections

The primary focus of this part of the case study is to facilitate visual comparison of the variability in climate model precipitation calculations with an observed measure of climate variability, the Oceanic Niño Index (ONI). The ONI is a metric of the shift between El Niño (warm) and La Niña (cool) events in the tropical Pacific [100]. This phenomenon is a well-documented driver of year-to-year variability in climate worldwide, representing a major shift of winds around the globe and providing the primary basis for forecasting on seasonal time scales.

There were major El Niño events during the years 1983 and 1998. Those major weather events resulted in substantial increases in precipitation in parts of the world, and are represented through exceptionally high ONI values during those years. We use this information in the examples that follow to look for visual correlation between precipitation variability as represented in different types of temporal projections and known major weather events.

We begin with temporal projections from a 4D space (latitude, longitude, time, ensemble) to a 1D space (time), shown in Fig 6.24. For each time value T, we compute $p = f(S_{lat}, S_{lon}, E)$ across all spatial locations (S_{lat}, S_{lon}) and ensemble members E, where $f \in [\bar{x}, \sigma, C_v]$. Since the data are computed and stored at monthly temporal resolution, our computations produce a yearly value from monthly values.

In the plot of $\bar{x}$ (blue bars in Fig. 6.24a), there is relatively little variation visible in the mean from year-to-year, with the main feature being a gradual long-term trend of increasing precipitation levels. Comparing these mean yearly values with the ONI and the the major El Niño events of 1983 and 1998, which are reflected with exceptionally high ONI values during those years, there is no

visible correlation between yearly $\bar{x}$ and those high ONI values. Similar to the $\bar{x}$ plot, the σ plot (purple bars Fig. 6.24b) shows the little variation in the σ from year-to-year, and there is nothing remarkable about the σ during the major events of 1983 and 1998.

In contrast, looking at the C_v projection in Fig. 6.24c, these two major events correspond to the two years with the highest C_v values. The correspondence does not seem to hold for more moderate El Niño events, however (e.g., 1972).

For the sake of completeness, we present a boxplot presentation of yearly precipitation values in Fig. 6.24d, along with annotation showing the major El Niño events of 1983 and 1998. Here, the box attributes are computed as yearly mean, min, max, and quartiles from the monthly precipitation model data, across all ensemble members. From this image, there is no visible evidence of anything remarkable happening in terms of precipitation variability associated with the major events of 1983 and 1998. The conclusion here is that visualization method, i.e., bar chart vs. boxplot, is not the key issue. The key issue is that C_v reflects data characteristics in a way not possible with either $\bar{x}$ or σ.

The properties of the C_v map in Fig. 6.23c help to explain the behavior of the yearly bar plots of C_v in Fig. 6.24c. In essence, C_v is acting as a combined index of the occurrence of El Niño events and of anomalous rainfall over subtropical regions. If data were only retained over the ocean, the C_v projection onto time would likely improve as an index of El Niño variability. On the other hand, if data were only retained over land (to mask out the El Niño aspect), then it would provide a metric of variations in subtropical deserts, without any parametric definition of what constitutes a subtropical desert. In contrast, the yearly bar plots of the mean and standard deviation are mostly reflecting activity in the ITCZ.

6.6.2 Winds

We now explore these projections for wind speed data from the climate model simulations. The data are the monthly average wind speed on the 500 hPa surface, the pressure surface that is about half the pressure at sea level and which lies approximately 5.5 km above sea level. The images in Fig. 6.25 are spatial projections, from a four-dimensional space—two spatial dimensions, time, and ensemble members—down to a two-dimensional latitude/longitude projection.

The most prominent features in the map of the mean winds are the mid-latitude jet streams. These winds are strongest over the ocean, flow from west to east in the 40°–50° latitude range, and extend down to the surface (hence named the “Roaring 40s” in the Southern Hemisphere). These

appear as horizontally oriented regions of red in Fig. 6.25a, the projection of mean wind speed, $\bar{x}$.

The map of σ (Fig. 6.25b) appears to show that the jets over the North Pacific and North Atlantic are variable, while the southern jet is instead quite steady except in the South Pacific. However, the C_v map (Fig. 6.25c) indicates that the spatial alignment of features is not perfect. In fact, the jet cores over the North Pacific, North Atlantic, and Antarctic Oceans are all very steady. The variation instead comes from a tendency of the winds to expand toward the equator: there is little or no power on the poleward side of the jets.

It is well known that the jets vary in their north-south position, with those variations so prominent that they form the leading Principal Components of extratropical variability, often termed the "Southern Annual Mode" (Antarctic), the "Pacific/North America Pattern" (North Pacific), and the "North Atlantic Oscillation" (North Atlantic); the northern two PCs are sometimes merged into the "Northern Annular Mode" [101, 102]. However, the point that these variations are manifest through an equatorward expansion of the winds, and not through a poleward expansion or through north-south shifts of the jet core, is not something that is readily apparent in the patterns associated with the PCs, which are themselves ignorant of the context of the mean base flow.

Comparison of the σ and $\bar{x}$ maps can reveal the equatorward tendency, but it requires careful scrutiny. On the other hand, by stressing the differences between the $\bar{x}$ and σ maps, the high-value regions in the C_v map are more clearly displaced from those in $\bar{x}$ map, meaning that the asymmetry in the north-south movement of the jets is apparent even in just a casual comparison.

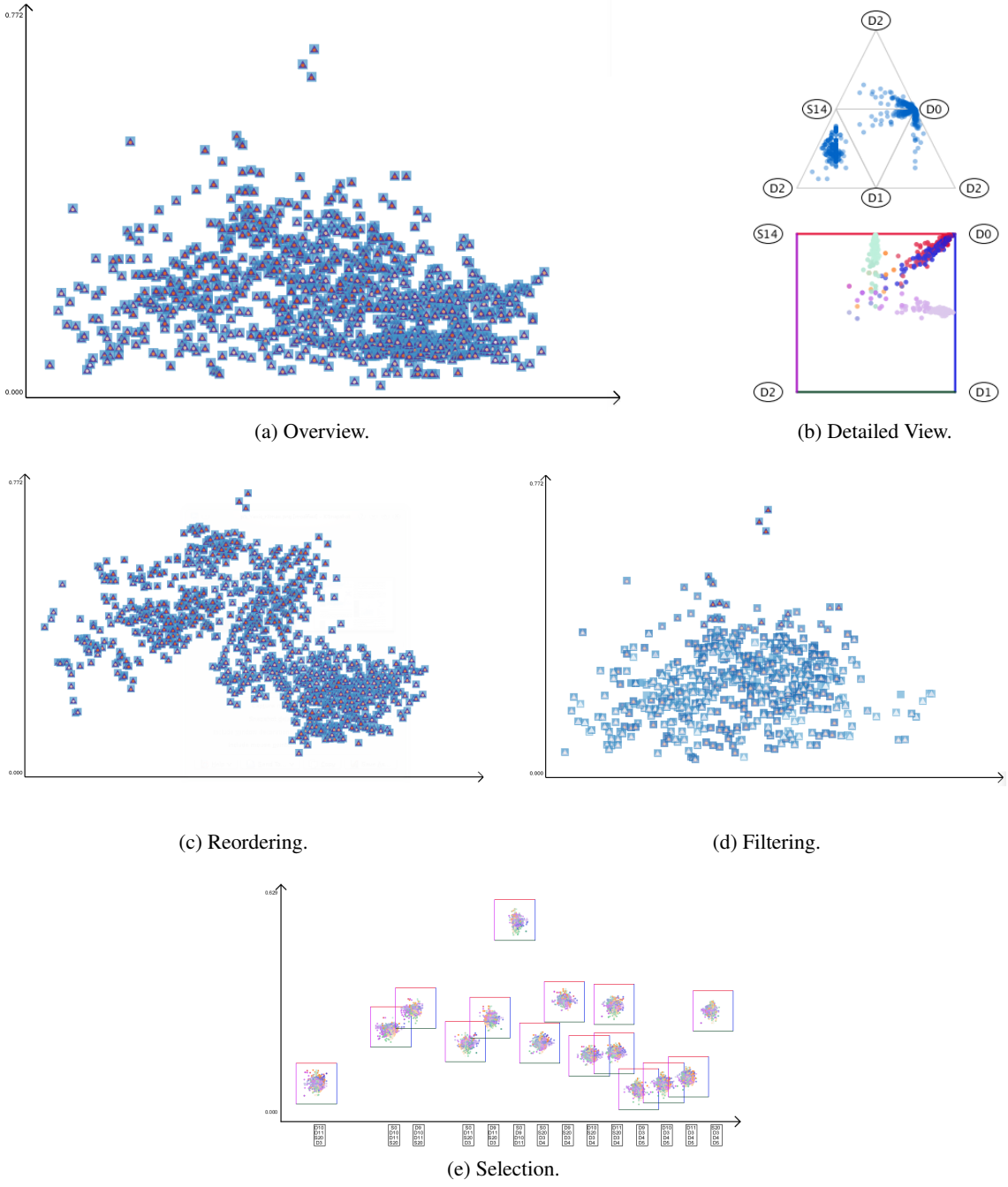
6.7 Discussion

We have proposed a method that visualizes multiway dependencies from multivariate data. Previous work has focused on two-way or three-way correlations. UnTangle Map can represent only two-way or three-way dependencies. We propose a new glyph-based visualization for high-dimensional data that includes an extension to UnTangle for four-way dependencies. The combination of these designs and filtering/selection interactions provides a powerful visual exploration mechanism that is intuitive and effective.

Our approach is scalable to both the number of variables and the size of data, as demonstrated by the Hurricane Isabel dataset, which contains hundreds of variables and millions of points. Few other approaches have attempted to analyze this number of variable dependencies. The upper limit for the number of dimensions tested was 624 for the Hurricane data.

We chose to limit our approach to four-way dependencies for a number of reasons. First, the number of 5-way relationships is huge, e.g., $\binom{624}{5} = 775$ billion. Second, as the number of independent variables grows, there is a naturally increasing coefficient of determination (i.e., more input variables are more likely to explain an output variable). Nevertheless, most of our visual encodings could be extended to 5-way dependencies.

Finally, our approach uses the R^2 coefficient of determination for multiple correlation with the Pearson correlation coefficient and Spearman Rank correlation coefficient. Many other statistical models could be used in place of the coefficient of determination, depending on the requirements of the analysis.



(a) Overview.

(b) Detailed View.

(c) Reordering.

(d) Filtering.

(e) Selection.

Figure 6.1: Overview+detail of the marketing research dataset. Our approach for representing multiway dependencies in multivariate data begins with (a) an overview supported by a glyph representation of pairwise, three-way, and four-way relationships for all sets of four variables. The overview can be (c) reordered, (d) filtered, and (e) zoomed until a relationship of interest is identified. A selected glyph then populates the detailed view (b).

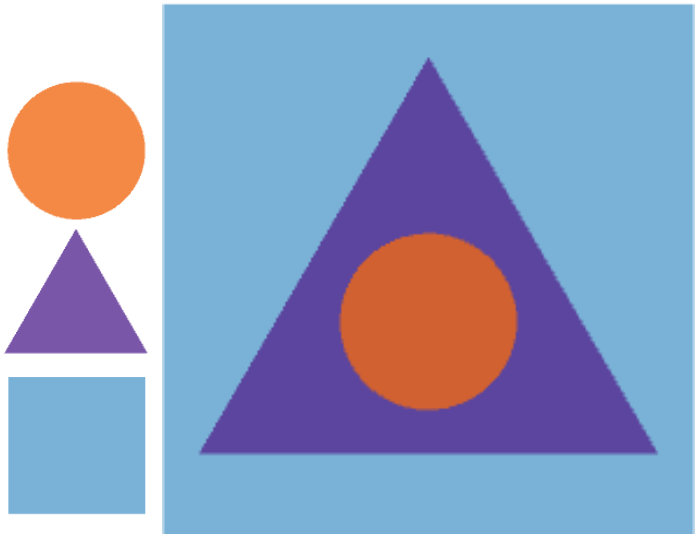
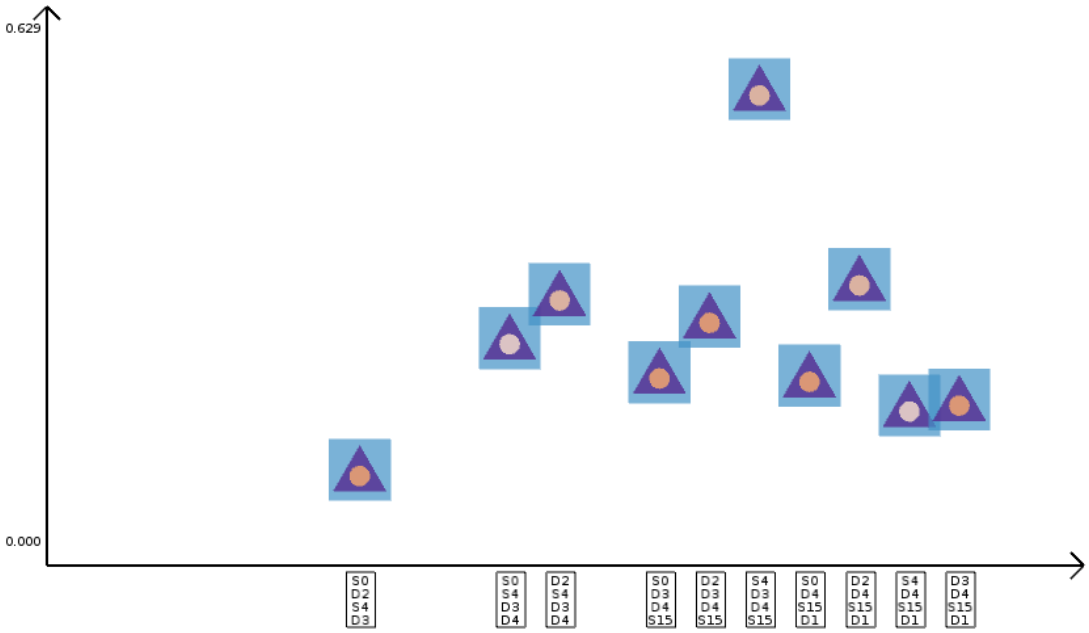
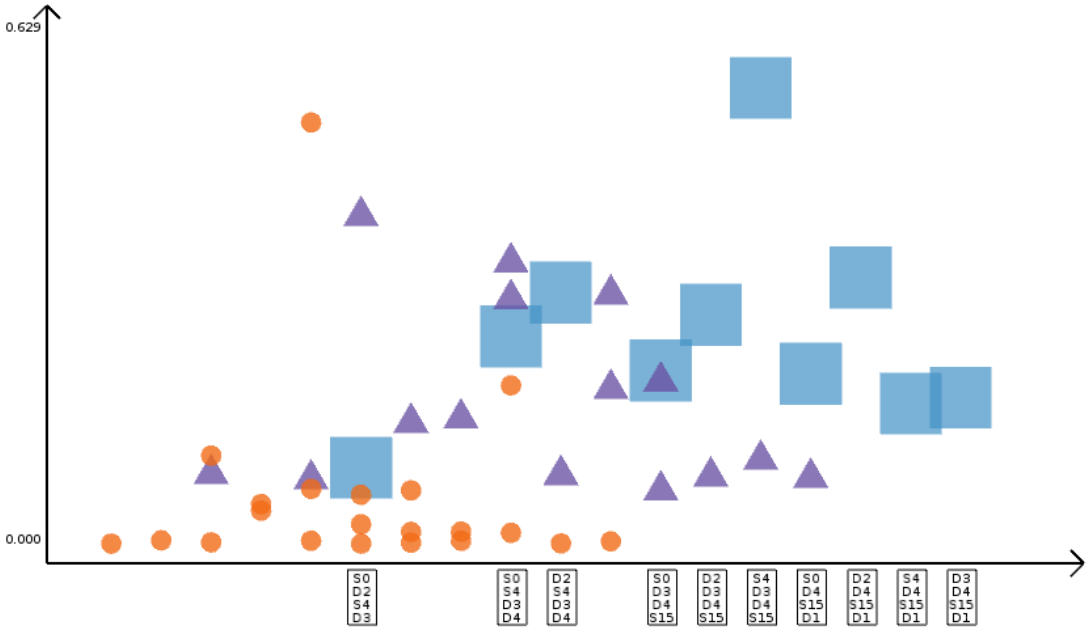


Figure 6.2: Visual encodings for multiway dependencies in overview

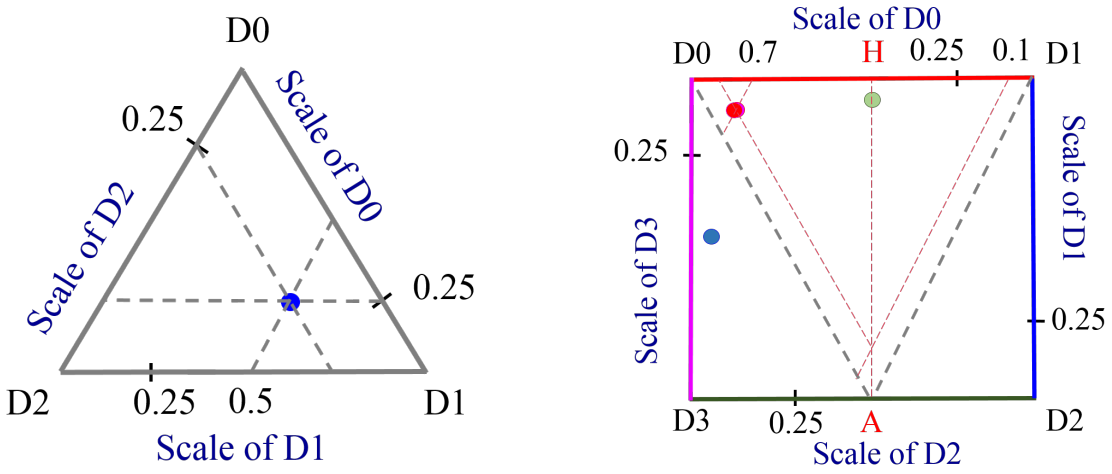


(a) Composite glyph overview.



(b) Individual circles, triangles, and square glyph overview.

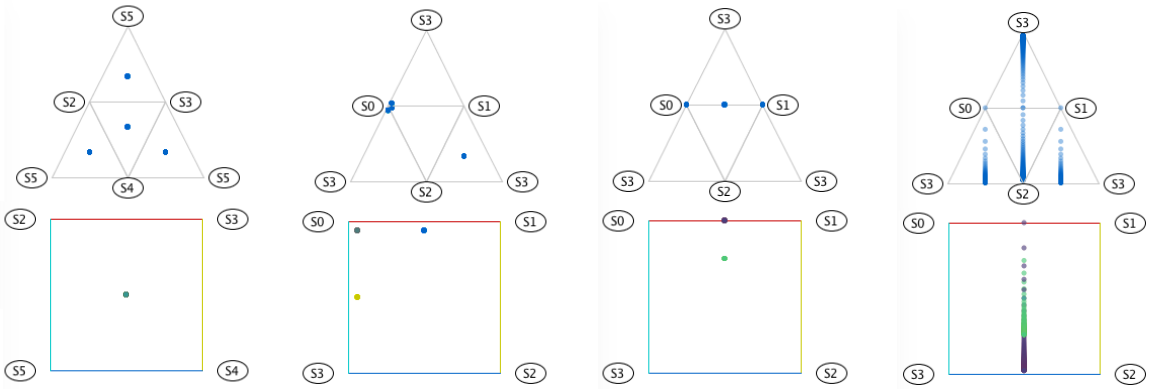
Figure 6.3: Example overviews for multiway dependencies.



(a) two- and three-way mapping

(b) four-way mapping

Figure 6.4: Mapping for two- and three-, and four-way dependencies.



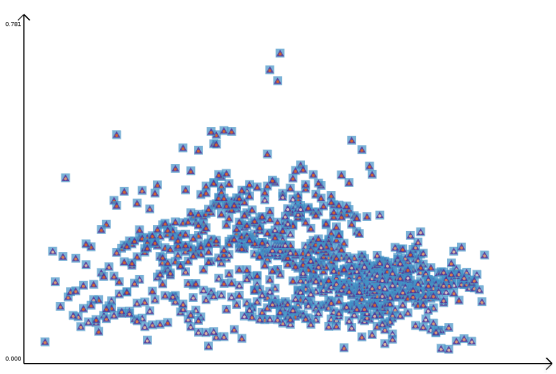
(a) nondominant

(b) unidominant

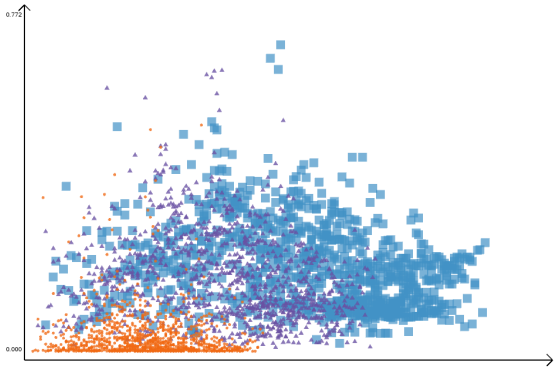
(c) bidominant

(d) balance

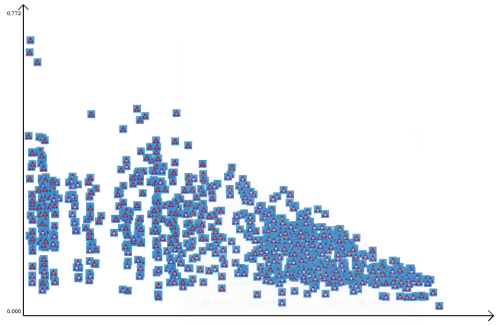
Figure 6.5: Visual patterns for (a) nondominant, (b) unidominant, (c) bidominant, and (d) balance relationship.



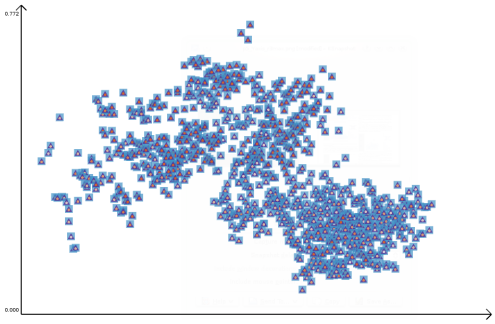
(a) Spearman Rank Correlation



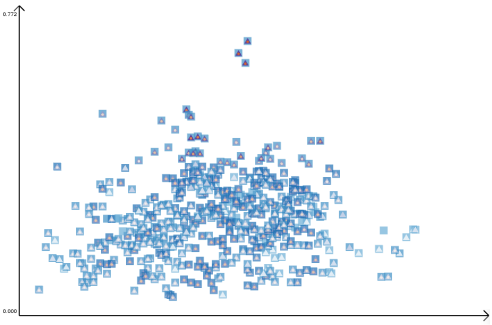
(b) two- and three- and four-way dependency



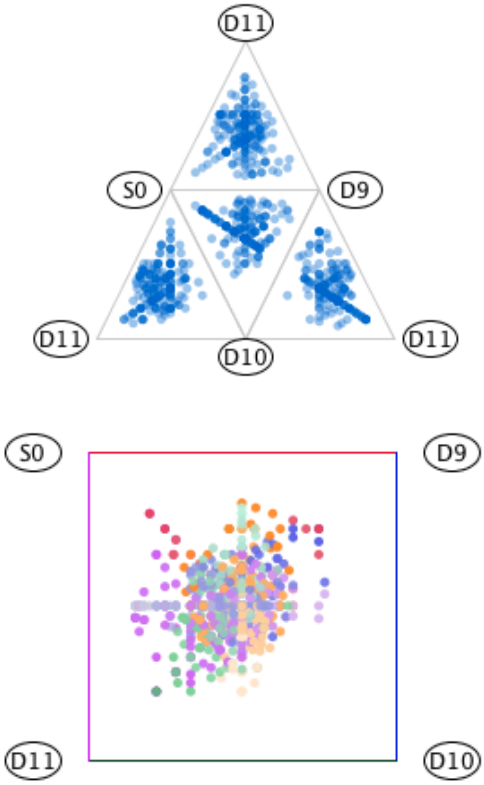
(c) Switching x-axis ordering



(d) Switching y-axis ordering

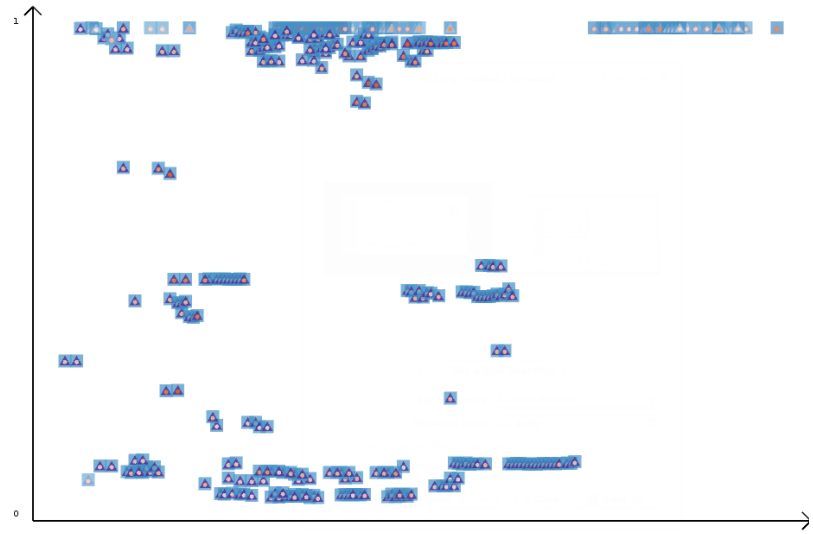


(e) Overview after filtering

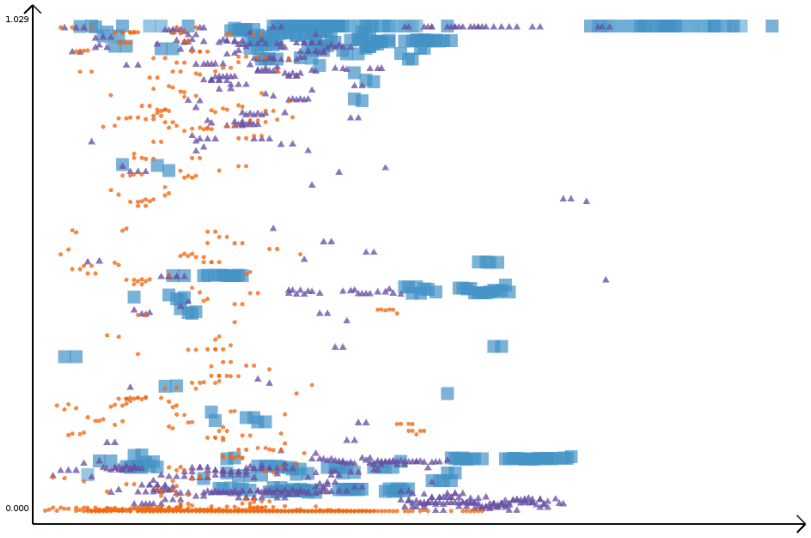


(f) Selection View

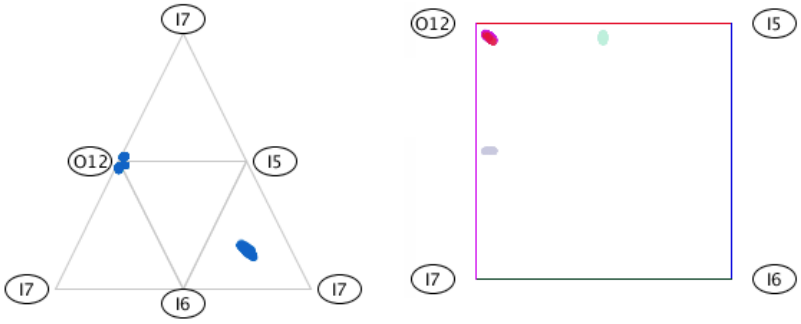
Figure 6.6: A variety of overviews and one detailed view of the marketing research dataset.



(a) Composite glyph overview

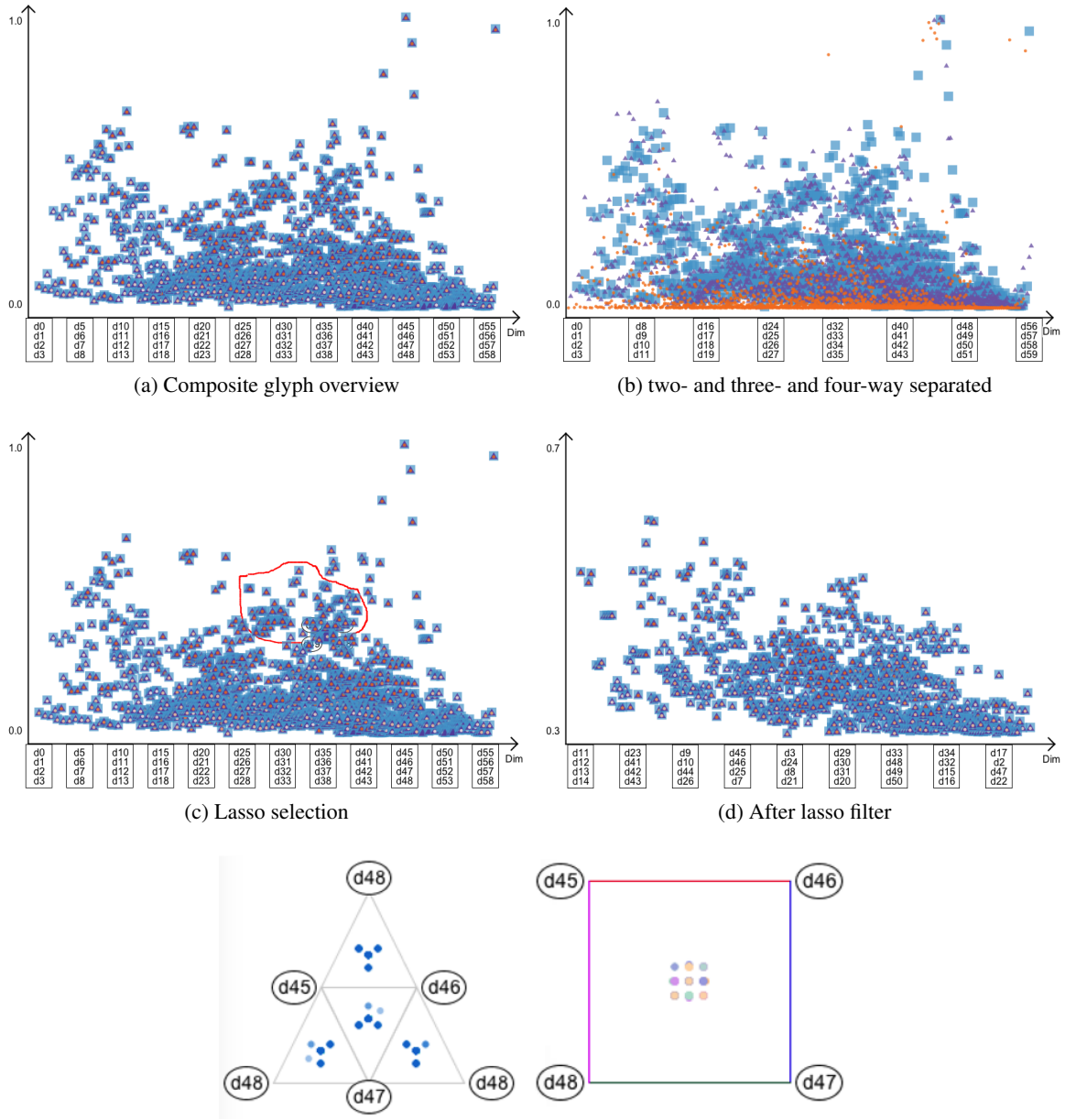


(b) two- and three- and four-way glyphs overview



(c) detailed view

Figure 6.7: Two overviews and one detailed view of the physics data.



(a) Composite glyph overview

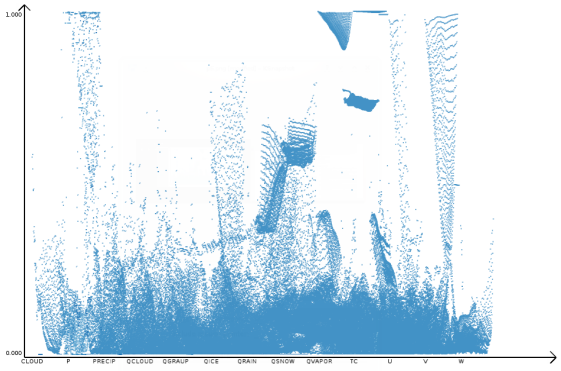
(b) two- and three- and four-way separated

(c) Lasso selection

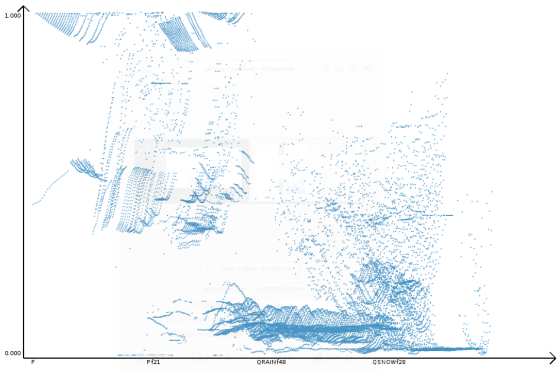
(d) After lasso filter

(e) detailed view of data

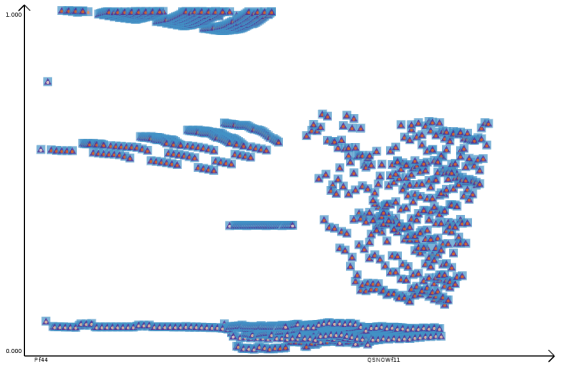
Figure 6.8: A variety of overviews and a single detailed view of the NHATS data.



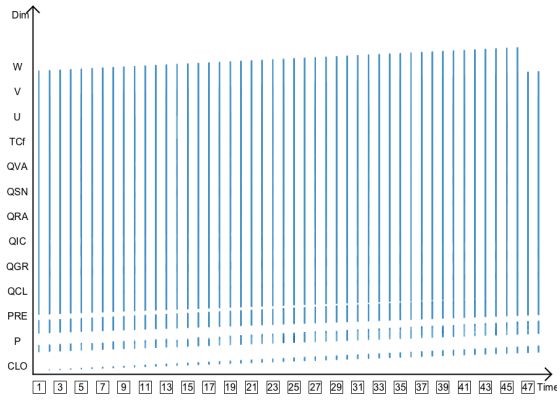
(a) Overview of all variables



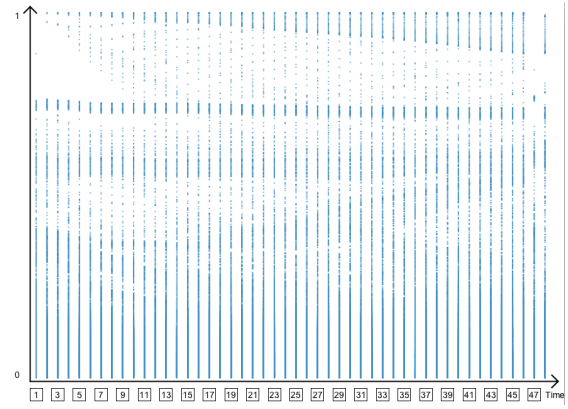
(b) Selected *QRAIN* and *QSNOW*



(c) Zoomed *QRAIN* and *QSNOW*



(d) Overview sorted by time



(e) Overview sorted by time & R^2

(f) Selection of *CLOUD*

Figure 6.9: Hurricane Isabel data by variable series (a-c) and timeline series (d-f).

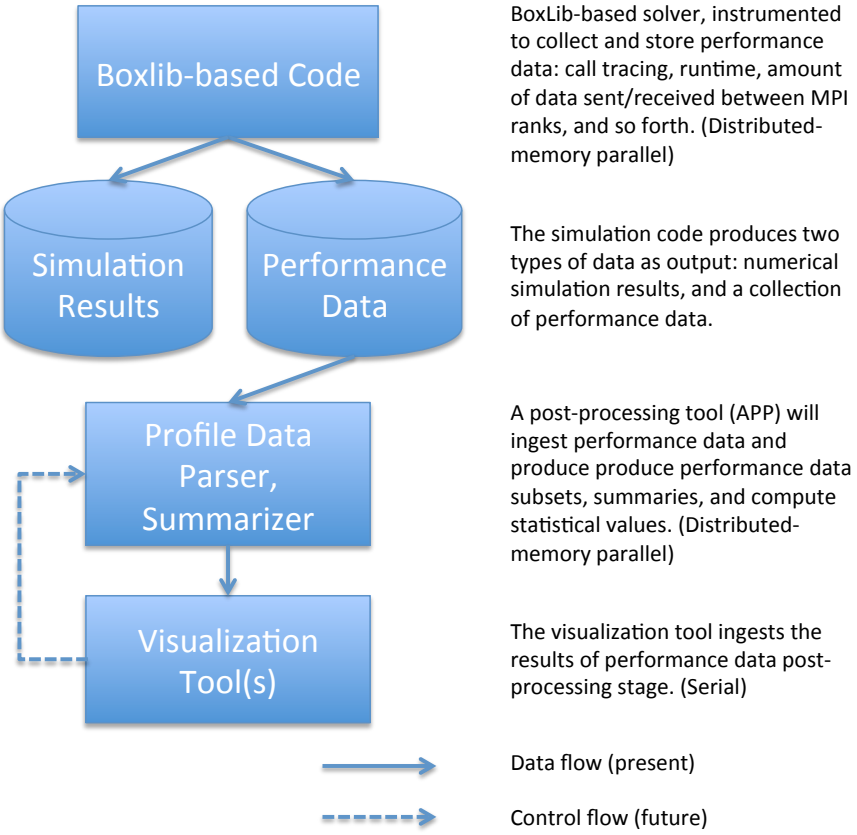


Figure 6.10: This illustration shows the work flow, where a parallel BoxLib-based simulation code produces simulation results and performance data, which is then postprocessed by a parallel tool that extracts performance data subsets and computes performance data projections. Finally, a serial visualization process displays the projection information. Future work will focus on directly linking the visualization process back to the projection process to better enable interactive coupling between these two processes.

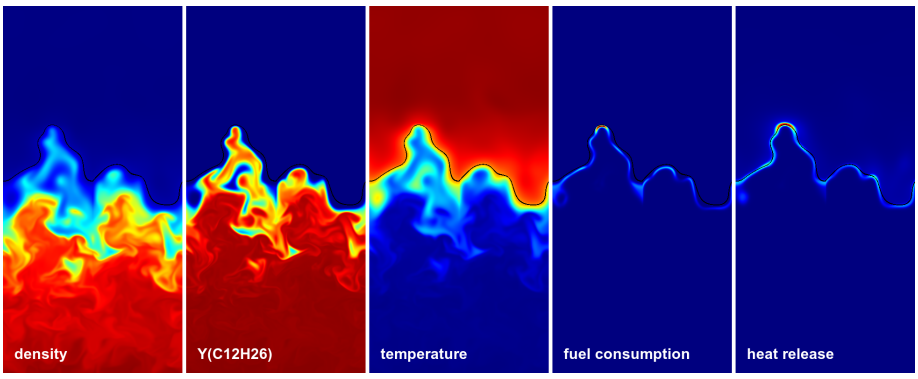


Figure 6.11: Low Mach number Combustion (LMC).

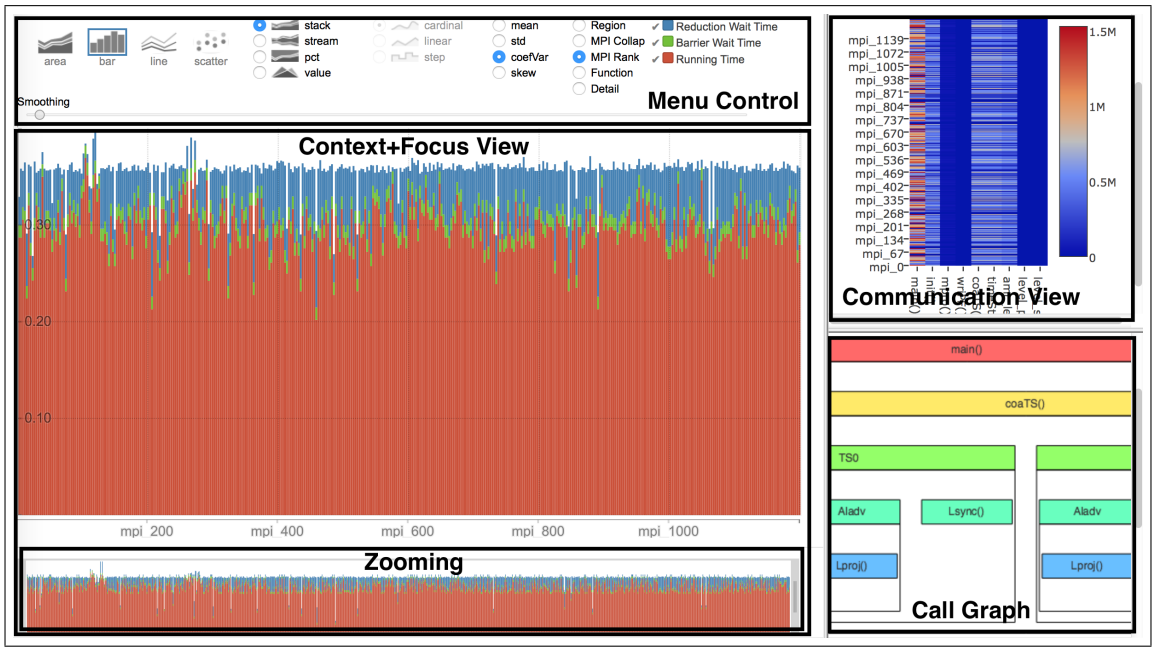
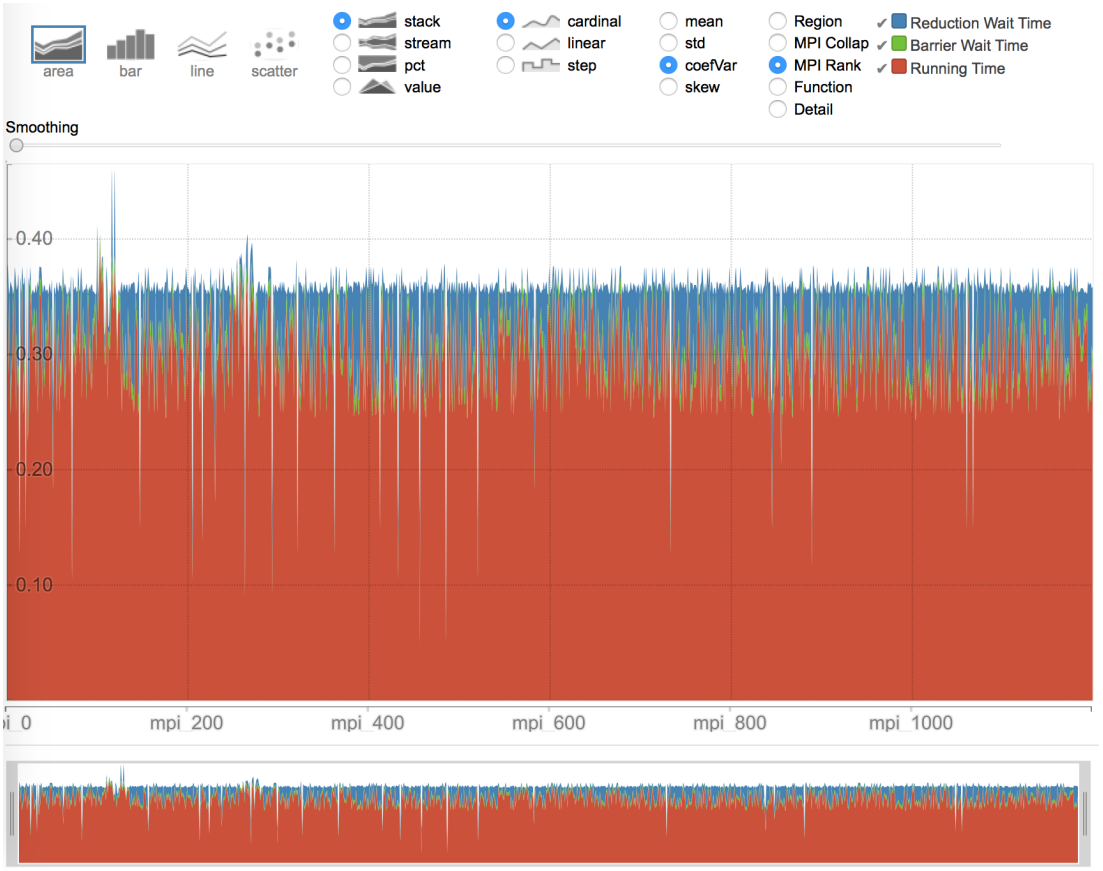
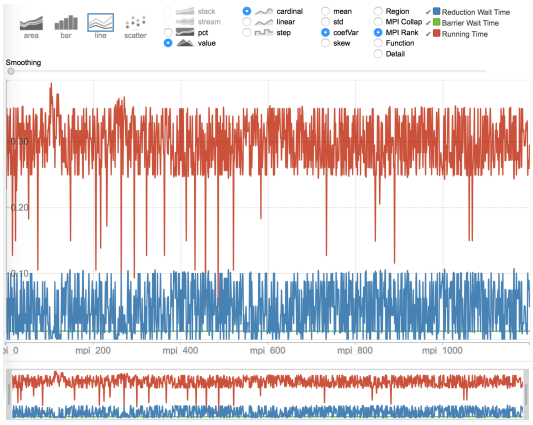


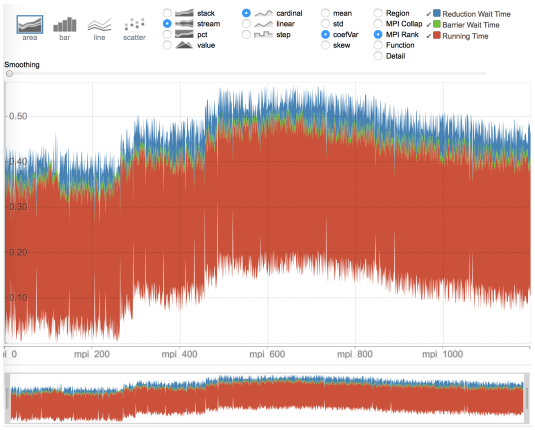
Figure 6.12: When viewing performance data projections, a user has the ability to select between different performance metrics (runtime, barrier wait time, etc.) and projections (mean, skew, etc.) using different charting methods. Interactions with a widget along the bottom, which shows a context data view, perform subset selection of the projection dataset. The data focus view, which is the main data display, and the call graph on the right, reflect the subset selection performed in the context view.



(a) Area Graph

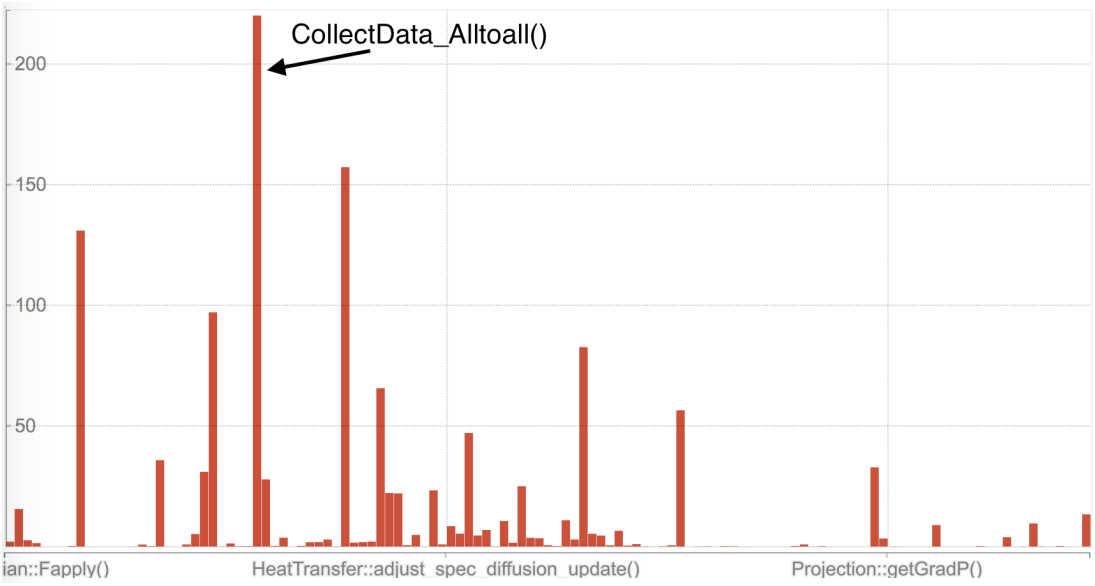


(b) Line Graph

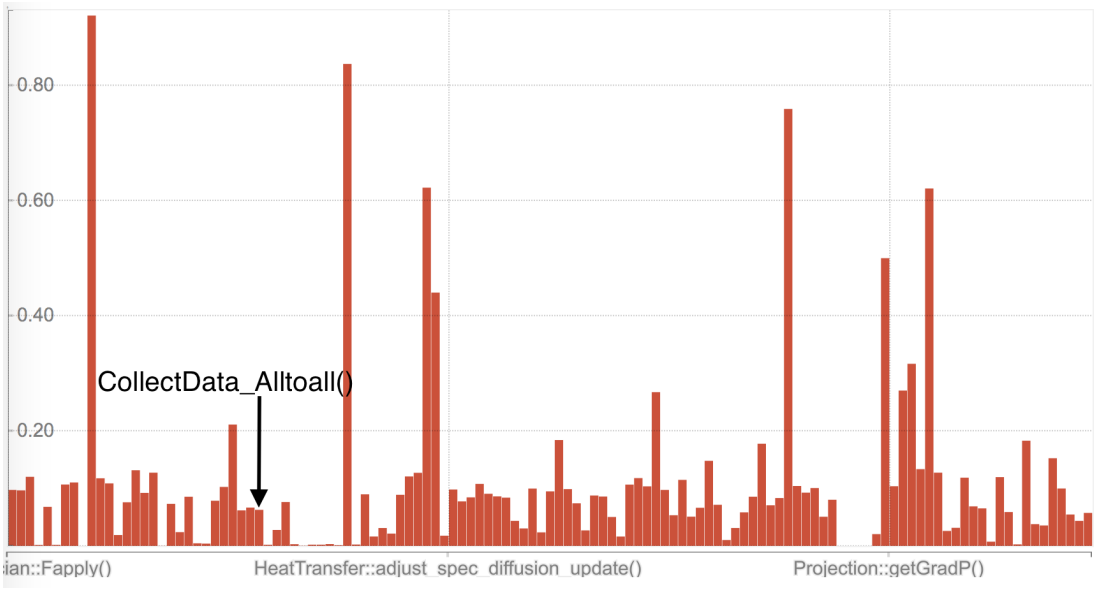


(c) Stream Graph

Figure 6.13: Visual encodings: area(a), line(b), stream(c) and bar graph(Fig. 6.12).

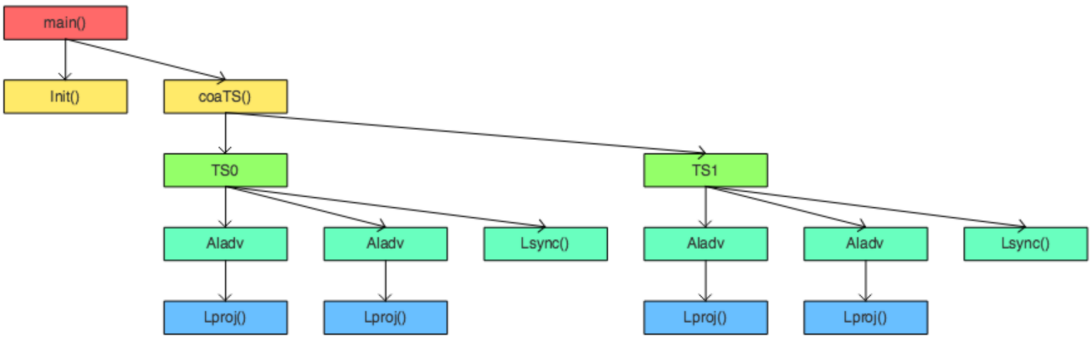


(a) $\bar{x}$ runtime for all function calls

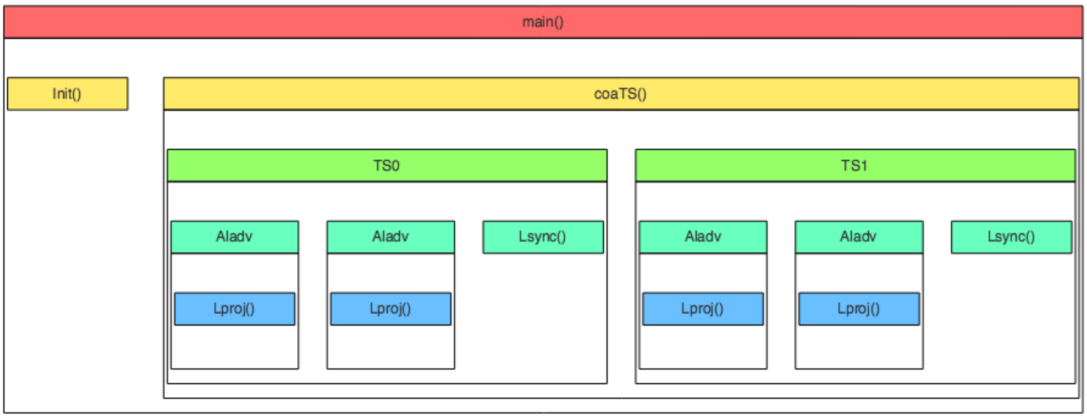


(b) C_v runtime for all function calls

Figure 6.14: Function call projection: average and Coefficient of Variation of each function across all MPI ranks, all regions, and all timesteps.

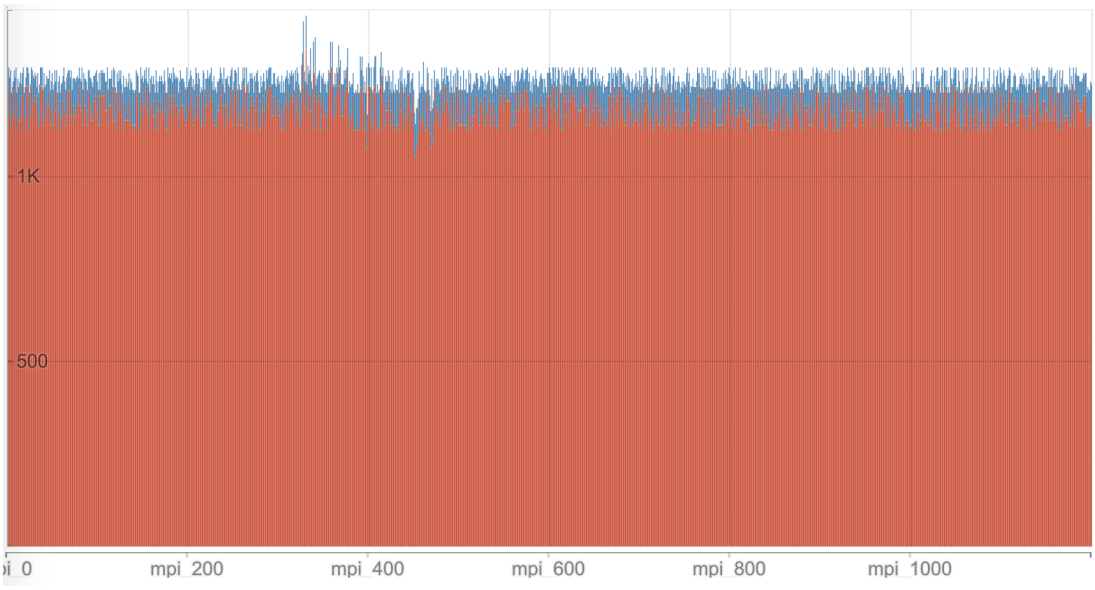


(a) Tree View

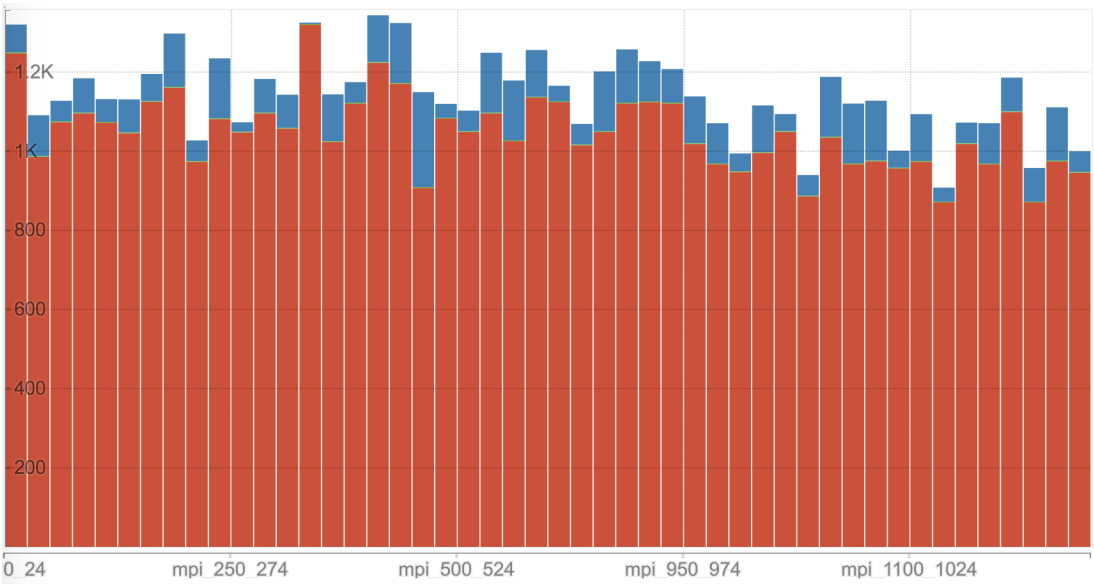


(b) Nested View

Figure 6.15: Code regions graph.

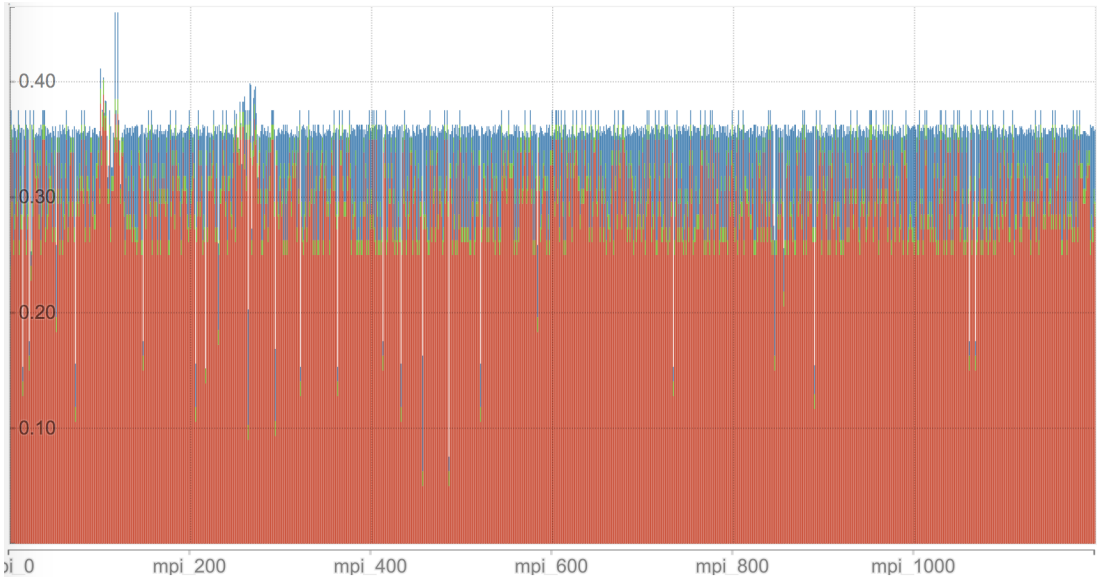


(a) Full resolution: $\bar{x}$ execution time for all MPI ranks

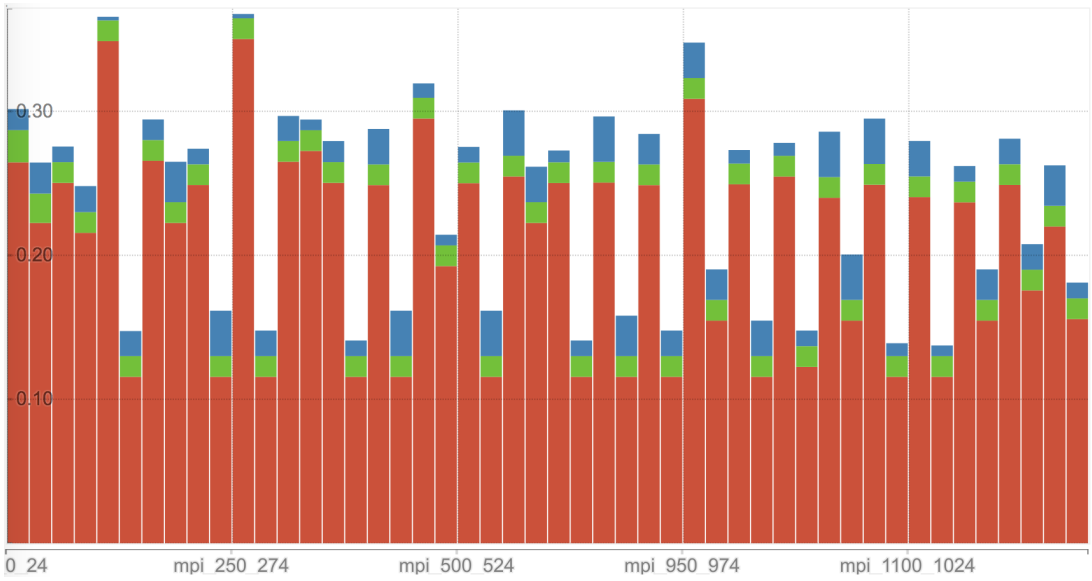


(b) Coarse resolution (25:1): $\bar{x}$ execution time for all MPI ranks

Figure 6.16: Rank projection: average execution time (red), barrier wait time (green), and reduction wait time (blue) for each MPI ranks across all timesteps.

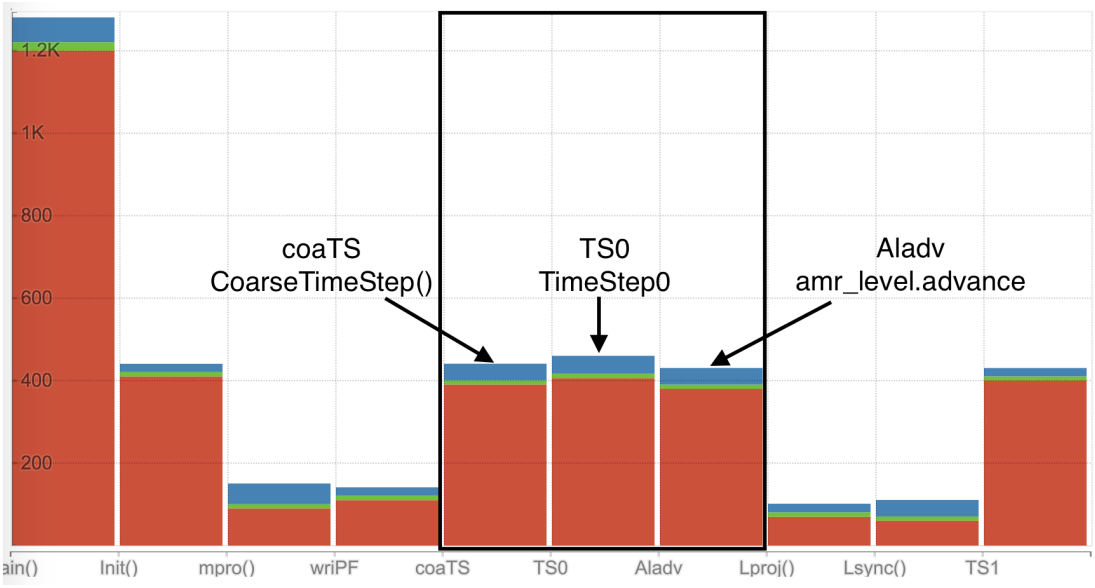


(a) Full resolution: C_v for all MPI ranks

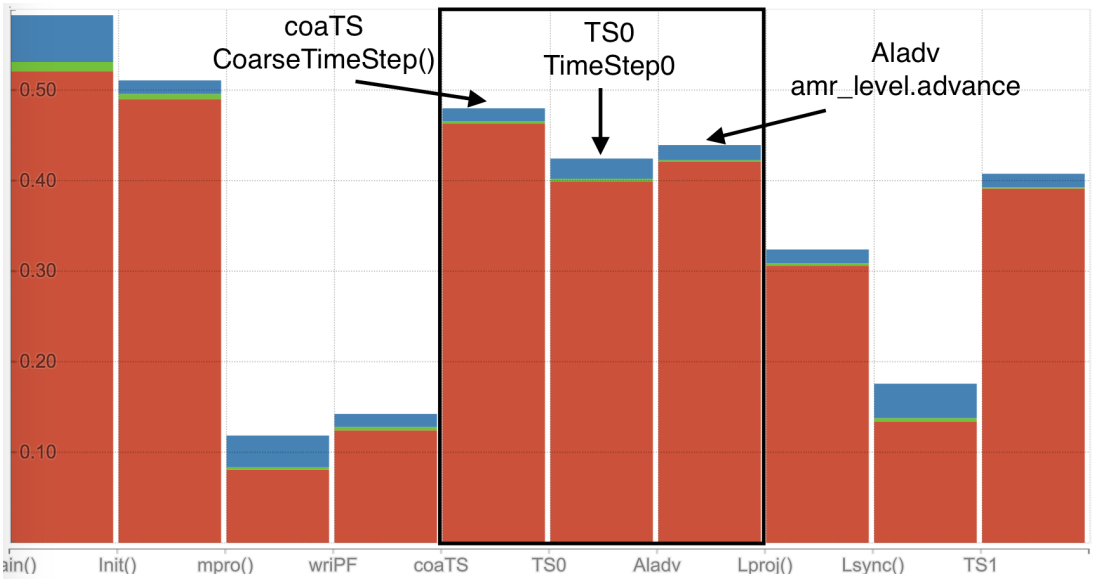


(b) Coarse resolution (25:1): C_v for all MPI ranks)

Figure 6.17: Rank projection: variation (C_v) in execution time (red), barrier wait time (green), and reduction wait time (blue) for each MPI ranks across all timesteps.

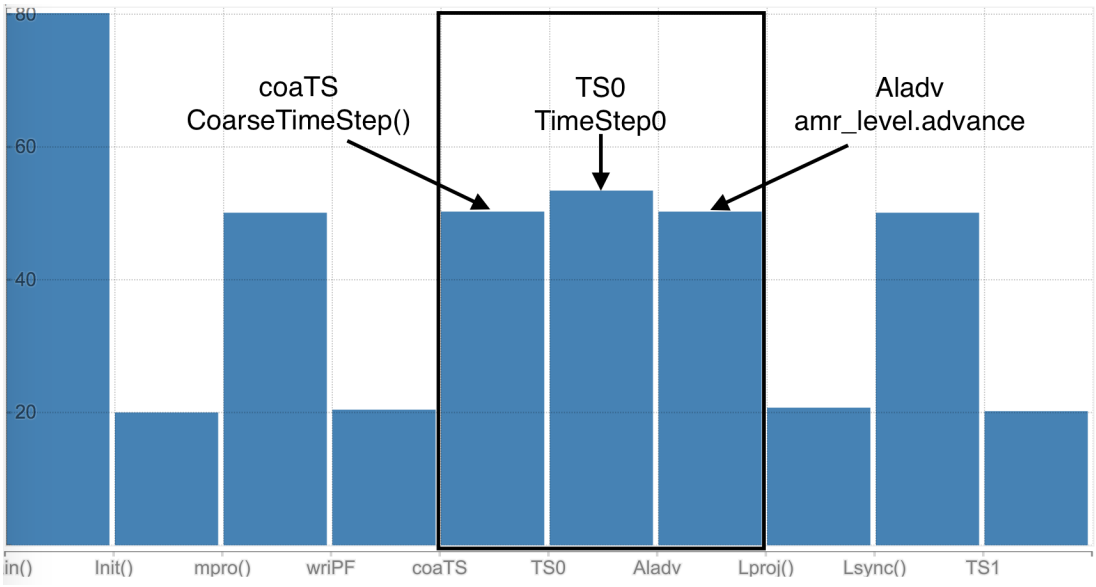


(a) $\bar{x}$ of regions

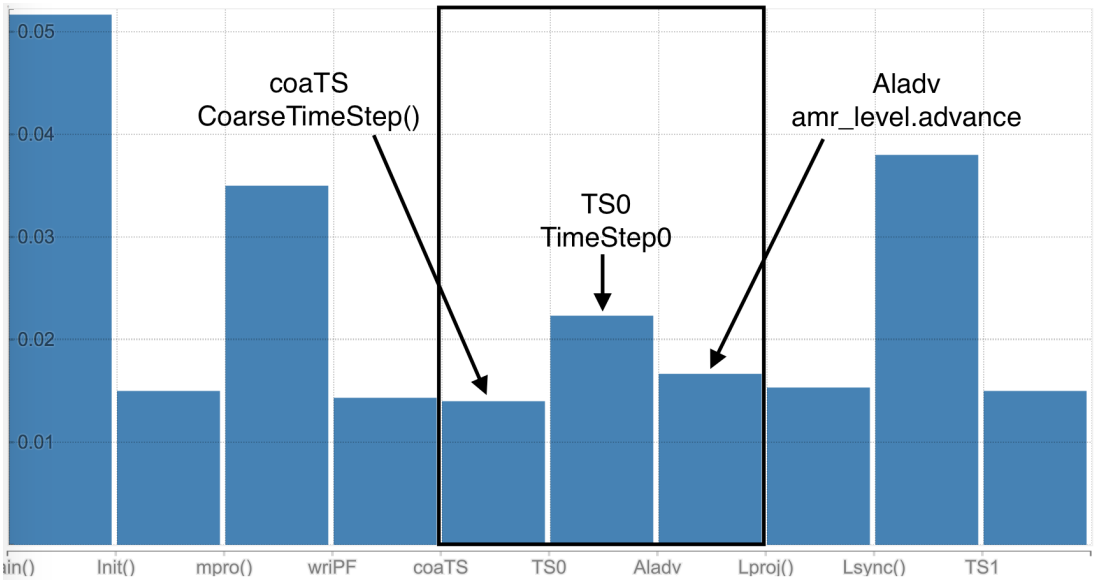


(b) C_v of regions

Figure 6.18: Region projection of runtime (red), reduction wait time (blue), and barrier wait time (green) across all MPI ranks. The image shows that the `Aladv` region has higher $\bar{x}$ and C_v levels than the others: it takes longer to run, and exhibits higher variation than the others.

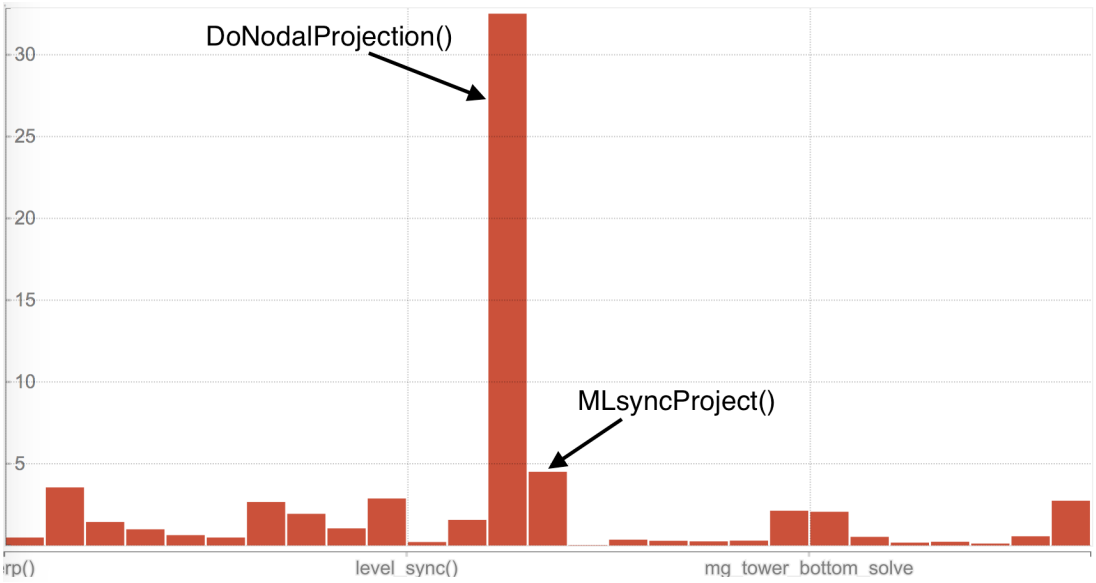


(a) $\bar{x}$ waiting time of different regions over all MPI ranks

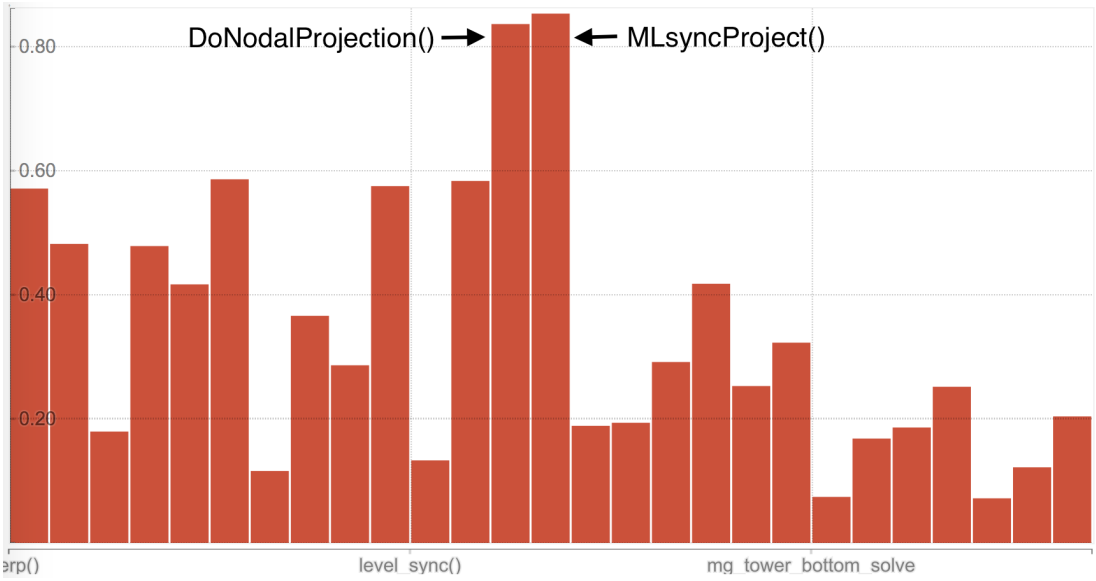


(b) C_v waiting time of different regions over all MPI ranks

Figure 6.19: Per-region projection of waiting time. This figure shows that `Aladv` region has higher $\bar{x}$ and C_v waiting time than others, providing a clue as to why there is imbalance in the code, and where.

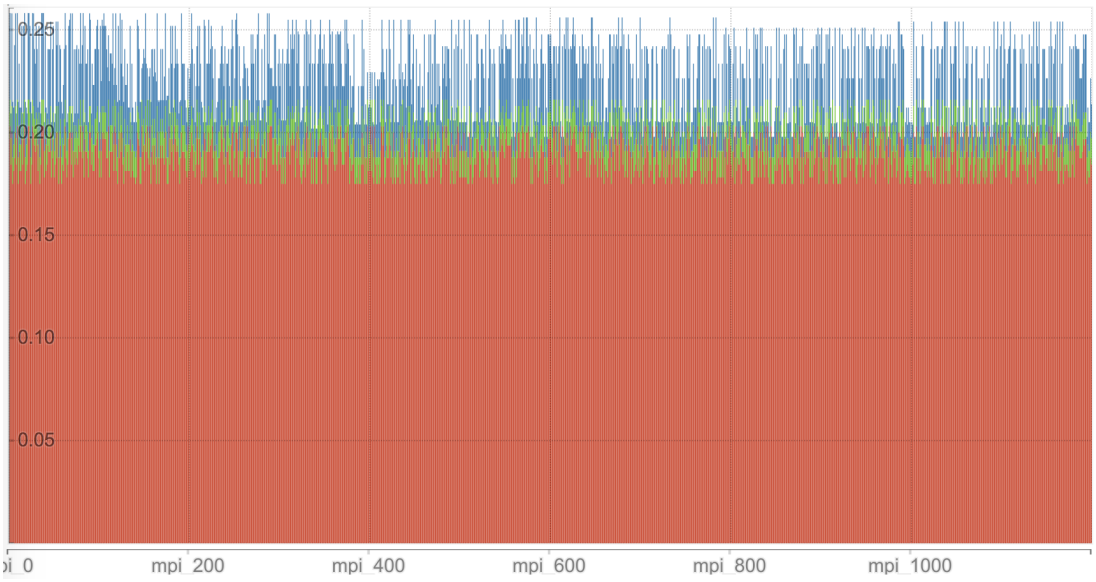


(a) $\bar{x}$ of functions in `Aladv` region across MPI ranks

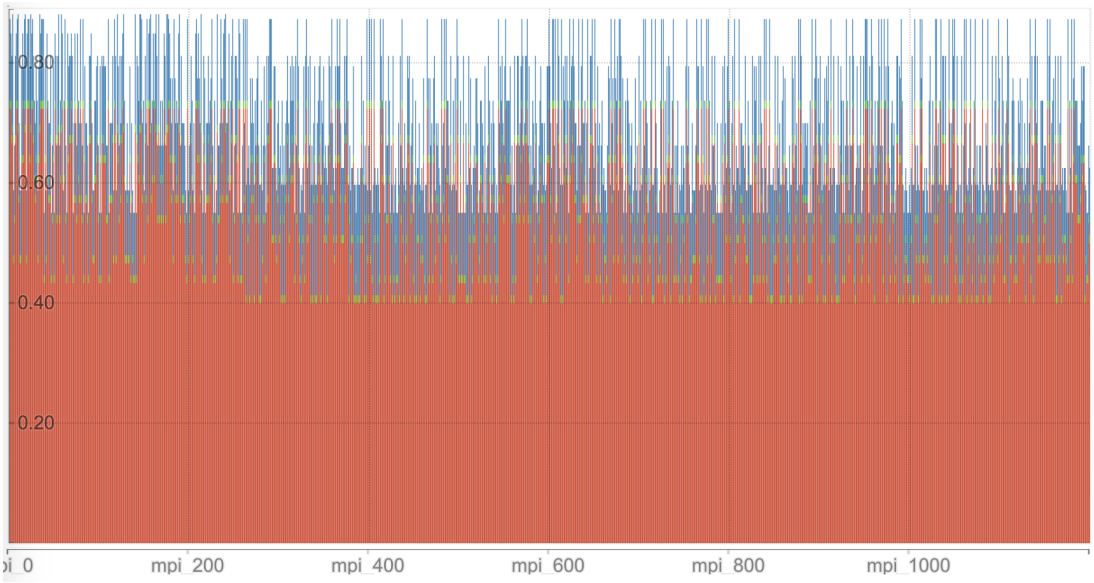


(b) C_v of functions in `Aladv` region across MPI ranks

Figure 6.20: Function projection of execution time within the `Aladv` region. level. We notice that functions `DoNodalProjection()` and `MLsyncProject()` have higher average execution time, and higher variation in execution time than the others. The hypothesis is that one of these two is the source of imbalance in the code.

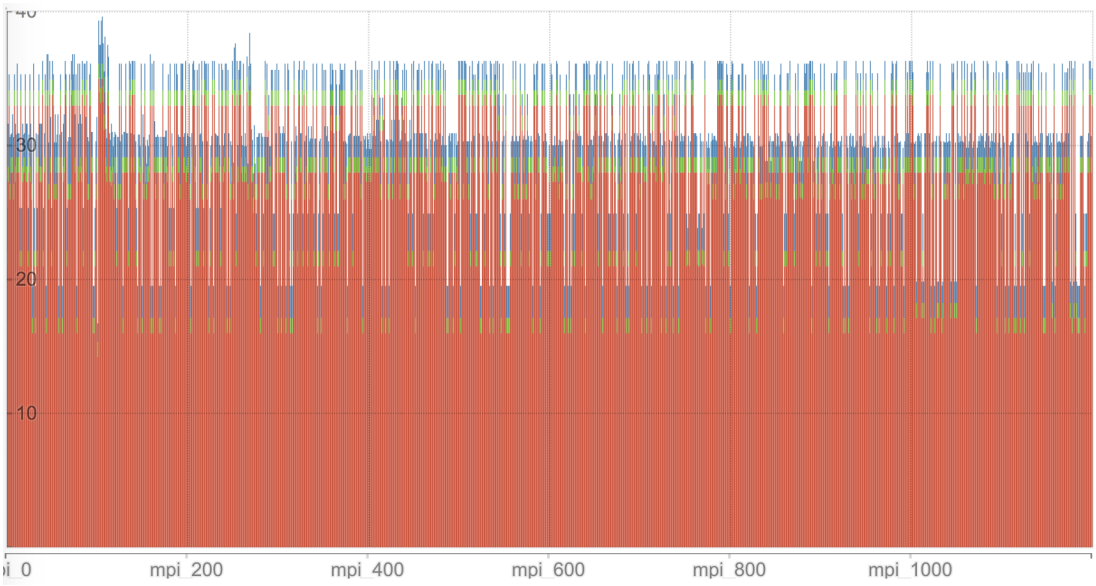


(a) `MLsyncProject()`, $\bar{x}$

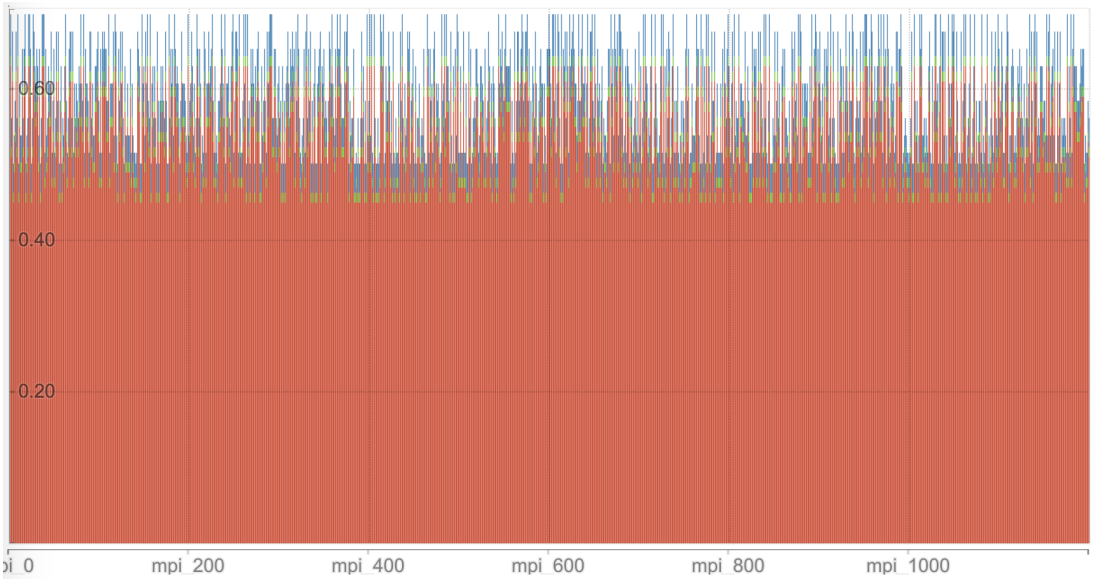


(b) `MLsyncProject()`, C_v

Figure 6.21: Function projection: `MLsyncProject()` performance across all MPI ranks within the `Aladv` code region. Execution time (red), barrier wait time (green), reduction wait time (blue).

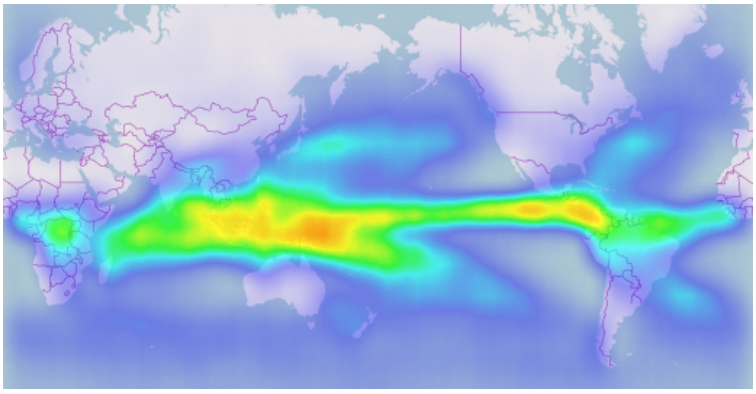


(a) `DoNodalProjection()`, $\bar{x}$

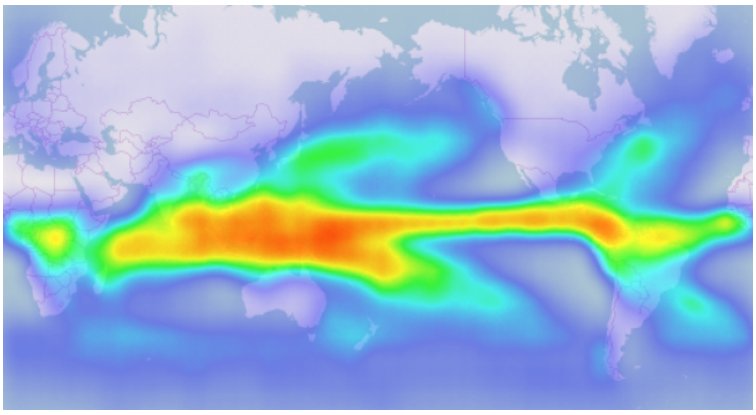


(b) `DoNodalProjection()`, C_v

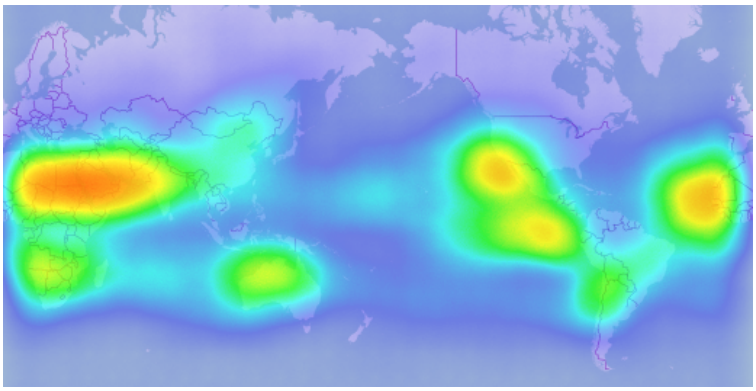
Figure 6.22: Function projection: `DoNodalProjection()` performance across all MPI ranks within the `Aladv` code region. Execution time (red), barrier wait time (green), reduction wait time (blue).



(a) $\bar{x}$, lat/lon projection through all ensemble members across all years.

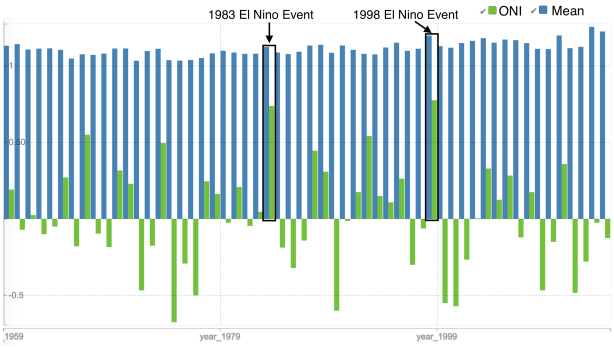


(b) σ, lat/lon projection through all ensemble members across all years.

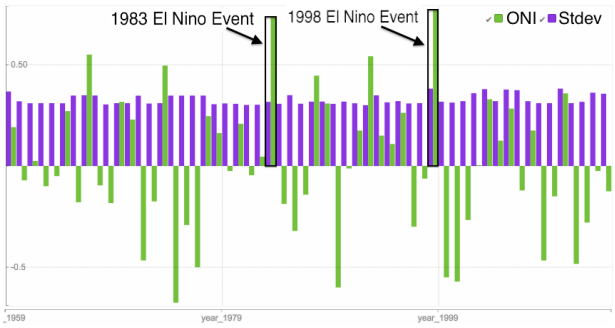


(c) C_v, lat/lon projection through all ensemble members across all years.

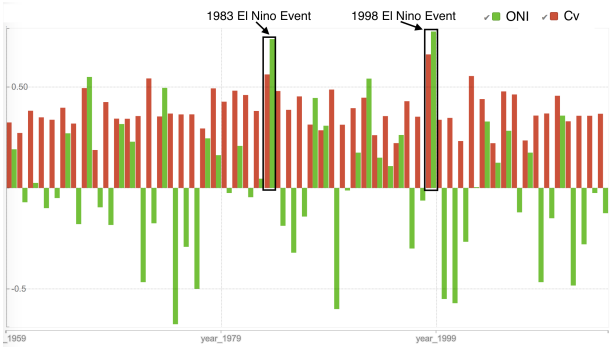
Figure 6.23: Comparison of $\bar{x}$, σ, and C_v as the basis of a spatial projection of climate model output, where we go from 4D to 2D. The C_v projection shows specific features not visible in either the $\bar{x}$ or σ projections, which are both similar in appearance.



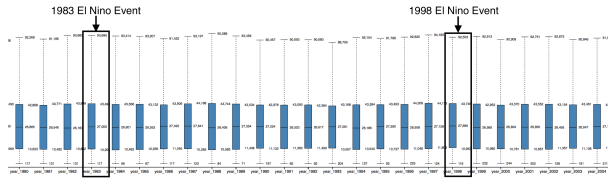
(a) $\bar{x}$.



(b) σ.

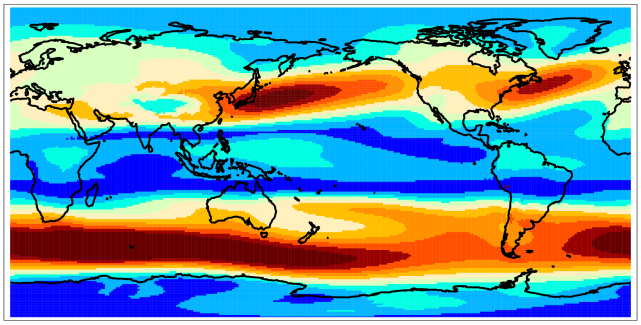


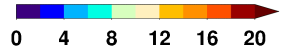
(c) C_v.



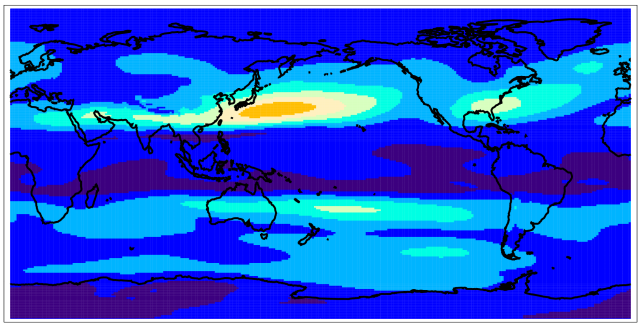
(d) Box plots for precipitation from year 1980 to 1994

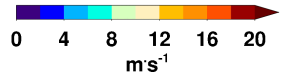
Figure 6.24: Comparison of $\bar{x}$, σ, and C_v as the basis for temporal projection operators, where we project from all spatial locations and ensemble members to yearly values. We show these temporal projections in comparison with the ONI. Of these projections, the C_v projection shows the strongest correlation with ONI, which is a known measure of climate variability.



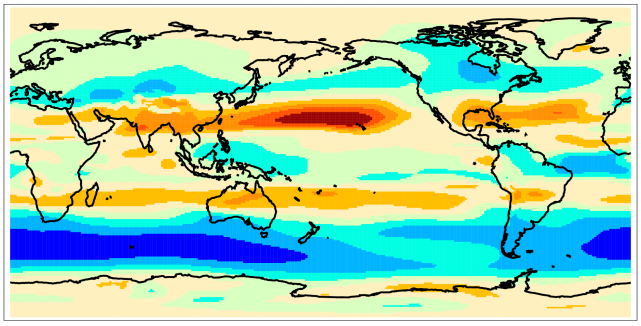


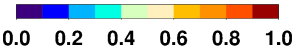
(a) $\bar{x}$





(b) σ





(c) C_v

Figure 6.25: 2D spatial projections of 500 hPa wind speed from a 4D space. Comparison of the C_v map against the $\bar{x}$ map reveals that the strong mid-latitude winds have a tendency to expand equatorward but not poleward, some thing that is harder to distinguish in the σ map.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Contribution

In this dissertation, we proposed four techniques of a new methodology that provides new visual designs to show details when possible and aggregates when necessary, along with robust interactive mechanisms. These techniques enable users to quickly identify and investigate meaningful relationships in large and multidimensional data. Depending on the data size and the number of dimensions, the most appropriate technique can be chosen to optimize the correlation identification performance.

First, correlation coordinate plots (CCPs) have been developed for the task-specific visualization. They have distinct advantages when compared to general task visualizations such as SCP and PCP. The advantages, as confirmed by our user study, are providing simple visual cues, improving the estimation of correlation strength by focusing the coordinate system on model fit, and improving correlation identification performance by reducing ambiguity in the visualization.

The second correlation visualization technique, the snowflake visualization, showed significant performance improvements over SPLOMs and PCPs. The snowflake visualization is an efficient focus+context style layout representing a fair compromise among space-efficient design, comprehensive visualization, and reduced user interaction for showing all pairwise correlations in multidimensional data. In addition, the snowflake visualization showed significant performance improvements over SPLOMs and PCPs. We believe that the CCP and snowflake visualization represent complementary approaches to existing techniques, replacing existing approaches only where correlation is the major feature of focus in data. We believe that more of these task-specific approaches are on the horizon and will provide data analysts better, faster access to relevant information in their data.

Our third proposed technique is data scalable parallel coordinates (DSPCP). In DSPCP, we propose a new model for mapping data from the attribute domain into the parallel coordinates domain. DSPCP has two major advantages. First, our approach scales well with increases in the size of data (up to 10 millions data points) and avoids the overdraw problem. Second, using thoughtful

encodings, data clustering, and interactions helps to identify relationships previously difficult to find in PCPs. Our approach supports identification of linear patterns, nonlinear patterns, clusters, and hidden patterns, and even enables detecting some outliers. The results of our experiments for simulated and real-world data demonstrate that DSPCP is practical for high-performance analysis of large and complex data. Finally, we compared and evaluated our method with the conventional PCPs through different tasks to emphasize the benefits of our approach.

Finally, we proposed a visualization technique to explore multiway dependencies in multivariate data, and we demonstrated the effectiveness of our approach through exploration of large simulation and real-world datasets. Our main contributions are in the new approach to represent these data and a set of visual encodings and interactions for exploring multiway dependencies with a large number of variables (up to 624). Although the number of potential dependencies is large, our visual encoding design helps to reduce a significant number of glyphs required to represent the information, and our detailed view helps clarify the exact nature of such relationships. Our approach is scalable to both the number of variables and the size of data, as demonstrated by the large and multidimensional real-world datasets.

We compared and evaluated these methods with conventional methods through different visualization tasks in real-world, large, and multidimensional data to emphasize the benefits of our methodology.

7.2 Future Work

In the future, we anticipate our methods will continued to be applied to the larger and higher-dimensional real-world datasets that further demonstrate that our approaches are practical for high-performance analysis in large and multidimensional data.

Confirming this to be true will require future user studies that measure how easily our methods enable performing multiple visualization tasks. These user studies would ideally be formal evaluations of our methods compared with some existing visualization methods in terms of learning time and visualization tasks performance such as accuracy and speed. The user studies we conducted for our first and second approaches showed that our method outperforms SCP and PCP. Conducting user studies for our third and fourth approaches (DSPCP and MultiDepViz) will make their evaluations even stronger.

Our methods do not naturally support many data types, such as categorical, text data, etc. Sup-

port for these data types will require improvements and modifications to our designs. Beyond our current designs, new designs should also be investigated to better optimize correlation identification for the specific users and correlation tasks.

Improving design is just one half of optimizing performance. Some of these approaches may not be naturally intuitive to many users. First, understanding the learning curves for both non-visualization and visualization familiar users learning our methodology will be valuable. Following that, additional research into the best methods for training users on the use of these approaches would further benefit their utility.

Finally, we believed that our methodology can be applied to additional statistical methods beyond just Pearson Correlation Coefficient, Rank Correlation Coefficient, and Multiway Dependency. Other statistical methods, such as t-tests, ANOVA, or nonlinear regressions may benefit as well.

REFERENCES

[1] X. Hong, C.-X. Wang, J. S. Thompson, B. Allen, W. Q. Malik, and X. Ge, "On Space-Frequency Correlation of UWB MIMO Channels," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4201–4213, sep 2010.

[2] R. K. Sharma and J. W. Wallace, "Correlation-Based Sensing for Cognitive Radio Networks: Bounds and Experimental Assessment," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 657–666, sep 2011.

[3] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang, "Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient," *IEEE Transactions on Parallel Distribution System*, vol. 23, no. 6, pp. 1073–1080, jun 2012.

[4] F. J. Anscombe, "Graphs in Statistical Analysis," in *The American Statistician*, vol. 27, no. 1. London, UK: Springer Berlin Heidelberg, Feb 1973, pp. 17–21.

[5] S. B. Jarrell, *Basic Statistics*. Pennsylvania Plaza, New York, US: McGraw-Hill Education, Nov 1994, vol. 23, no. 1.

[6] A. Inselberg, "The Plane with Parallel Coordinates," *The Visual Computer*, vol. 1, no. 2, pp. 69–91, aug 1985.

[7] A. Aris and B. Shneiderman, "Designing Semantic Substrates for Visual Network Exploration," in *Information Visualization 2007*, vol. 6, no. 4. New York, NY, USA: Palgrave Macmillan, Dec 2007, pp. 281–300.

[8] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J.-D. Fekete, "GraphDice: A System for Exploring Multivariate Social Networks," *Computer Graphics Forum*, vol. 29, no. 3, pp. 863–872, dec 2010.

[9] M. Wattenberg, "Visual Exploration of Multivariate Graphs," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '06, vol. 12, no. 1. New York, NY, USA: ACM, sep 2006, pp. 811–819.

[10] J. Heinrich and D. Weiskopf, "State of the Art of Parallel Coordinates," in *Eurographics 2013 - State of the Art Reports*, vol. 2, no. 1. Germany: The Eurographics Association, sep 2013, pp. 95–116.

[11] H. N. R. M. K. Tiago Etiene and C. T. Silva, "'Flow Visualization' Juxtaposed with 'Visualization of Flow': Synergistic Opportunities between Two Communities," in *51st AIAA Aerospace Meeting*, vol. 3, no. 5. New York, NY, USA: AIAA, dec 2013, pp. 51–59.

[12] D. Lehmann, F. Kemmler, T. Zhyhalava, M. Kirschke, and H. Theisel, "Visualnostics: Visual Guidance Pictograms for Analyzing Projections of High-dimensional Data," *Computer Graphics Forum*, vol. 34, no. 3, pp. 291–300, jun 2015.

[13] A. Dasgupta, M. Chen, and R. Kosara, "Conceptualizing Visual Uncertainty in Parallel Coordinates," *Computer Graphics Forum*, vol. 31, no. 3, pp. 1015–1024, jun 2012.

[14] G. Albuquerque, T. Löwe, and M. Magnor, "Synthetic Generation of High-dimensional Datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2317–2324, Dec 2011.

[15] M. Friendly, "Corrgrams: Exploratory Displays for Correlation Matrices," *AMSTAT*, vol. 1, no. 3, sep 2002.

[16] Y.-H. Chan, C. D. Correa, and K.-L. Ma, "Flow-based Scatterplots for Sensitivity Analysis," in *IEEE Visual Analytics Science and Technology 2010*, vol. 24, no. 3. New York, NY, USA: IEEE, October 2010, pp. 43–50.

[17] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in Visual Data Analysis," in *Information Visualization*, ser. IV '06, vol. 34, no. 2. IEEE, sep 2006, pp. 9–16.

[18] C.-K. Chen, C. Wang, K.-L. Ma, and A. T. Wittenberg, "Static Correlation Visualization for Large Time-Varying Volume Data," *PacificViz*, vol. 2, no. 3, pp. 27–34, dec 2011.

[19] M. Beham, W. Herzner, M. E. Gröller, and J. Kehrer, "Cupid: Cluster-Based Exploration of Geometry Generators with Parallel Coordinates and Radial Trees," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1693–1702, sep 2014.

[20] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang, "iPCA: An Interactive System for PCA-based Visual Analytics," in *Eurographics*, ser. EuroVis'09, vol. 34, no. 3. Chichester, UK: The Eurographics Association, jun 2009, pp. 767–774.

[21] S. Teoh and K.-L. Ma, *Hifocon: Object and Dimensional Coherence and Correlation in Multidimensional Visualization*, ser. Lecture Notes in Computer Science. Springer-Verlag, sep 2005, vol. 3804, no. 3.

[22] N. Cao, Y.-R. Lin, and D. Gotz, "UnTangle Map: Visual Analysis of Probabilistic Multi-Label Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 99, pp. 1–9, sep 2015.

[23] H. Nguyen and P. Rosen, "Improved Identification of Data Correlations through Correlation Coordinate Plots," in *International Conference on Information Visualization Theory and Application*, vol. 2, no. 3. Springer, 2016, pp. 34–42.

[24] ——, "Correlation Coordinate Plots: Efficient Layouts for Correlation Tasks," in *Springer Lecture Notes, Communications in Computer and Information Science, Springer*, vol. 2, no. 3. Springer, dec 2016.

[25] ——, "DSPCP: A Data Scalable Approach for Identifying Relationships in Parallel Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 99, pp. 1–9, 2017.

[26] ——, "Data Scalable Approach for Identifying Correlation in Large and Muti-Dimensional Data," in *International Conference on Information Visualization Theory and Application*, vol. 2, no. 3. New York, NY, USA: Springer, apr 2016, pp. 23–31.

[27] H. Nguyen, B. Wang, and P. Rosen, "Visual Exploration of Multiway Dependencies for Multivariate Data," in *SIGGRAPH ASIA 2016 Symposium on Visualization*, vol. 2, no. 1. New York, NY, USA: ACM, dec 2016, pp. 21–28.

[28] Y. A. Chen, J. S. Almeida, A. J. Richards, P. Muller, R. J. Carroll, and B. Rohrer, "A Nonparametric Approach to Detect Nonlinear Correlation in Gene Expression," *Journal of Computational and Graphical Statistics*, vol. 19, no. 3, pp. 552–568, sep 2010.

[29] W. Xu, C. Chang, Y. S. Hung, and P. C. W. Fung, "Asymptotic Properties of Order Statistics Correlation Coefficient in the Normal Cases." *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2239–2248, sep 2008.

[30] J. Wang and N. Zheng, "A Novel Fractal Image Compression Scheme with Block Classification and Sorting Based on Pearson's Correlation Coefficient," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 32–39, sep 2013.

[31] E. Magnello and B. Vanloon, *Introducing Statistics: A Graphic Guide*. Icon Books, 2009, vol. 78, no. 3.

[32] J. Benesty, J. Chen, and Y. Huang, "On the Importance of the Pearson Correlation Coefficient in Noise Reduction." *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 757–765, sep 2008.

[33] Y. Ke, J. Cheng, and W. Ng, "Efficient Correlation Search from Graph Databases." *IEEE Transactions on Knowledge Data Engineering*, vol. 20, no. 12, pp. 1601–1615, sep 2008.

[34] P. D. Allison, *Multiple Regression: A Primer*. London: London: Sage Publications, jan 1998, vol. 1, no. 2.

[35] T. Keith, *Multiple Regression and Beyond*. Boston, USA: Boston: Pearson Education, sep 2006, vol. 2, no. 4.

[36] T. Buering, J. Gerken, and H. Reiterer, "User Interaction with Scatterplots on Small Screens - A Comparative Evaluation of Geometric-Semantic Zoom and Fisheye Distortion," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 829–836, sep 2006.

[37] J. A. Hartigan, "Printer Graphics for Clustering," *Journal of Statistical Computation and Simulation*, vol. 4, no. 3, pp. 187–213, jun 1975.

[38] T.-H. Huang, M. L. Huang, and K. Zhang, "An Interactive Scatter Plot Metrics Visualization for Decision Trend Analysis," in *Conference on Machine Learning, Applications*, vol. 23, no. 1. New York, NY, USA: IEEE, jun 2012, pp. 258–264.

[39] N. Elmqvist, P. Dragicevic, and J.-D. Fekete, "Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1141–1148, sep 2008.

[40] H. Janicke, M. Bottinger, and G. Scheuermann, "Brushing of Attribute Clouds for the Visualization of Multivariate Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1459–1466, Nov 2008.

[41] E. Fanea, M. S. T. Carpendale, and T. Isenberg, "An Interactive 3D Integration of Parallel Coordinates and Star Glyphs." in *Information Visualization 2005*, vol. 24, no. 3. New York, NY, USA: IEEE, sep 2005, pp. 149–156.

[42] J. Li, J.-B. Martens, and J. J. van Wijk, "Judging Correlation from Scatterplots and Parallel Coordinate Plots," *Information Visualization 2008*, vol. 9, no. 1, pp. 13–30, Nov 2010.

[43] H. Zhou, W. Cui, H. Qu, Y. Wu, X. Yuan, and W. Zhuo, "Splatting Lines in Parallel Coordinates." *Computer Graphics Forum*, vol. 28, no. 3, pp. 759–766, sep 2009.

[44] 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*, vol. 3, no. 12, pp. 2572–2580, sep 2011.

[45] C. Viau, M. J. McGuffin, Y. Chiricota, and I. Jurisica, "The FlowVizMenu and Parallel Scatterplot Matrix: Hybrid Multidimensional Visualizations for Network Exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1100–1108, Nov 2010.

[46] D. Holten and J. J. van Wijk, "Evaluation of Cluster Identification Performance for Different PCP Variants," *EuroVis*, vol. 29, no. 3, pp. 21–28, jun 2010.

[47] H. Qu, W.-Y. Chan, A. Xu, K.-L. Chung, K.-H. Lau, and P. Guo, "Visual Analysis of the Air Pollution Problem in Hong Kong," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1408–1415, Dec 2007.

[48] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu, "Scattering Points in Parallel Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1001–1008, sep 2009.

[49] J. Heinrich, S. Bachthaler, and D. Weiskopf, "Progressive Splatting of Continuous Scatterplots and Parallel Coordinates," in *Eurographics*, ser. EuroVis 2011, vol. 3, no. 2. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, jun 2011, pp. 653–662. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2011.01914.x

[50] P. L. M. J. Jimmy Johansson and M. Cooper, "Revealing Structure in Visualizations of Dense 2D and 3D Parallel Coordinates," in *Information Visualization*, vol. 3, no. 2. New York, NY, USA: IEEE, sep 2006, pp. 125–136.

[51] K. T. M. Donnell and K. Muellers, "Illustrative Parallel Coordinates," *EUROVIS*, vol. 27, no. 3, pp. 221–239, jun 2008.

[52] J. Heinrich, J. Stasko, and D. Weiskopf, "The Parallel Coordinates Matrix," in *EuroVis - Short Papers*, vol. 2, no. 4. Chichester, UK: The Eurographics Association, jun 2012, pp. 37–41.

[53] J. Heinrich and D. Weiskopf, "State of the Art of Parallel Coordinates," in *EuroVis STAR*, vol. 2, no. 1. Chichester, UK: The Eurographics Association, sep 2013, pp. 95–116.

[54] ——, "Continuous Parallel Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1531–1538, sep 2009.

[55] H. Xiao, H. Guo, and X. Yuan, "Scalable Multivariate Volume Visualization and Analysis Based on Dimension Projection and Parallel Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1397–1410, dec 2012.

[56] P. Muigg, M. Hadwiger, H. Doleisch, and E. Groller, "Visual Coherence for Large-Scale Line-Plot Visualizations," *Computer Graphics Forum*, vol. 43, no. 2, pp. 643–652, jun 2011.

[57] H. Chen, "Compound Brushing Dynamic Data Visualization," in *Information Visualization 2003*, vol. 24, no. 3. New York, NY, USA: IEEE, sep 2003, pp. 181–188.

[58] M. O. Ward, "Linking and Brushing," in *Encyclopedia of Database Systems*. Springer, dec 2009, vol. 13, no. 2, pp. 1623–1626.

[59] P. C. Wong and R. D. Bergeron, "Multiresolution Multidimensional Wavelet Brushing," in *Information Visualization 1996*, vol. 3, no. 2. New York, NY, USA: IEEE, sep 1996, pp. 141–148.

[60] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner, "Structure-Based Brushes: A Mechanism for Navigating Hierarchically Organized Data and Information Spaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 2, pp. 150–159, dec 2000.

[61] L. Gosink, J. C. Anderson, E. W. Bethel, and K. I. Joy, "Variable Interactions in Query-Driven Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1400–1407, November/December 2007, lBNL-63524.

[62] M. Glatter, J. Huang, S. Ahern, J. Daniel, and A. Lu, "Visualizing Temporal Patterns in Large Multivariate Data using Modified Globbing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1467–1474, Nov 2008.

[63] J. Sukharev, C. Wang, K. Ma, and A. T. Wittenberg, "Correlation Study of Time-Varying Multivariate Climate Data Sets," in *PacificVis*, vol. 3, no. 4. New York, NY, USA: IEEE, Dec 2009, pp. 161–168.

[64] Pfaffelmoser, Tobias, and Westermann, "Correlation Visualization for Structural Uncertainty Analysis," *International Journal for Uncertainty Quantification*, vol. 3, no. 2, jun 2013.

[65] D. Jen, P. Parente, J. Robbins, C. Weigle, R. Taylor, A. Burette, and R. Weinberg, "ImageSurfer: A Tool for Visualizing Correlations between Two Volume Scalar Fields," in *IEEE Information Visualization*, vol. 2, no. 3. New York, NY, USA: IEEE, Oct 2004, pp. 529–536.

[66] C. W. Yi Gu, "A Study of Hierarchical Correlation Clustering for Scientific Volume Data," in *ISVC*, vol. 32, no. 1. Berlin, Heidelberg: Springer, jun 2010, pp. 437–446.

[67] N. Sauber, H. Theisel, and H. Seidel, "Multifield-Graphs: An Approach to Visualizing Correlations in Multifield Scalar Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 917–924, dec 2006.

[68] L. Adhianto, S. Banerjee, M. Fagan, M. Krentel, G. Marin, J. Mellor-Crummey, and N. R. Tallent, "HPCToolkit: Tools for Performance Analysis of Optimized Parallel Programs," *Concurrency and Computation: Practice and Experience*, vol. 22, no. 6, pp. 685–701, apr 2010.

[69] A. M. Wissink, R. D. Hornung, S. R.Kohn., S. S. Smith, and N. S. Elliott, "Large Scale Structured AMR Calculations Using the SAMRAI Framework," in *ACM/IEEE Supercomputing Conference*, vol. 20, no. 2. New York, NY, USA: IEEE, apr 2001, pp. 32–39.

[70] R. Gupta, P. Beckman, H. Park, E. Lusk, P. Hargrove, A. Geist, D. K. Panda, A. Lumsdaine, and J. Dongarra, "CIFTS: A Coordinated Infrastructure for Fault-Tolerant Systems," in *International Conference on Parallel Processing (ICPP)*, vol. 1, no. 2. New York, NY, USA: IEEE, dec 2009, pp. 21–28.

[71] H. Nguyen and G. Bronevetsky, "Visualizing the Behavior of Large Programs," in *IEEE Super Computing 2014*, vol. 2, no. 1. New York, NY, USA: IEEE, aug 2014, pp. 34–34.

[72] L. Harrison, F. Yang, S. Franconeri, and R. Chang, "Ranking Visualizations of Correlation Using Weber's Law," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1943–1952, dec 2014.

[73] M. Kay and J. Heer, "Beyond Weber's Law: A Second Look at Ranking Visualizations of Correlation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 469–478, dec 2016.

[74] E. Kandogan, "Visualizing Multi-Dimensional Clusters, Trends, and Outliers Using Star Coordinates," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 23, no. 1. New York, NY, USA: ACM, jun 2001, pp. 107–116.

[75] J. Li, J.-B. Martens, and J. J. van Wijk, "Judging Correlation from Scatterplots and Parallel Coordinate Plots," *Information Visualization 2010*, vol. 9, no. 1, pp. 13–30, Mar. 2010.

[76] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley, "DNA Visual and Analytic Data Mining," in *Information Visualization 1997*, vol. 3, no. 2. New York, NY, USA: IEEE, dec 1997, pp. 437–ff.

[77] D. J. Lehmann and H. Theisel, "General Projective Maps for Multidimensional Data Projection," *Computer Graphics Forum*, vol. 35, no. 2, 2016.

[78] E. Kandogan, "Star coordinates: A multi-dimensional Visualization Technique with Uniform Treatment of Dimensions," in *Information Visualization 2000*, vol. 650, 2000, p. 22.

[79] Y.-H. Chan, C. D. Correa, and K.-L. Ma, "Flow-based Scatterplots for Sensitivity Analysis," in *IEEE Visual Analytics Science and Technology*, vol. 24, no. 3. New York, NY, USA: IEEE, dec 2010, pp. 43–50.

[80] I. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer, jun 1986, vol. 23, no. 12.

[81] H. Sanftmann and D. Weiskopf, "Illuminated 3D Scatterplots," *Computer Graphics Forum*, vol. 32, no. 1, pp. 24–32, jun 2009.

[82] D. Matthew, R. L. S. Drysdale, and S. Jorg-Rudiger, "Simple Algorithms for Enumerating Interpoint Distances and Finding k Nearest Neighbors," *International Journal of Computational Geometry and Applications*, vol. 2, no. 3, pp. 221–239, sep 1992.

[83] M. Fournier, "Surface Reconstruction: An Improved Marching Triangle Algorithm for Scalar and Vector Implicit Field Representations," in *Computer Graphics and Image Processing (SIBGRAPI)*, vol. 2, no. 3. New York, NY, USA: IEEE, Nov 2009, pp. 72–79.

[84] L. Novakova and O. Stepankova, "Radviz and Identification of Clusters in Multidimensional Data," in *Information Visualization 2009*, vol. 2, no. 3. New York, NY, USA: IEEE, sep 2009, pp. 104–109.

[85] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An Efficient k-means Clustering Algorithm: Analysis and Implementation," *IEEE PAMI*, vol. 23, no. 3, pp. 881–892, sep 2002.

[86] M. Novotny and H. Hauser, "Outlier-Preserving Focus+Context Visualization in Parallel Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 893–900, Sep. 2006. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2006.170

[87] D. Lehmann and H. Theisel, "Discontinuities in Continuous Scatter Plots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1291–1300, dec 2010.

[88] ——, "Features in Continuous Parallel Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1912–1921, dec 2011.

[89] S. Rados, R. Splechtna, K. Matkovic, M. Duras, E. Groller, and H. Hauser, "Towards Quantitative Visual Analytics with Structured Brushing and Linked Statistics," in *Computer Graphics Forum*, vol. 35, no. 3, Wiley Online Library. Chichester, UK: The Eurographics Association, jun 2016, pp. 251–260.

[90] L. Harrison, F. Yang, S. Franconeri, and R. Chang, "Ranking Visualizations of Correlation Using Weber's Law," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1943–1952, dec 2014.

[91] H. T. Nguyen, D. Stone, and E. W. Bethel, "Statistical Projections for Multi-resolution, Multi-dimensional Visual Data Exploration and Analysis," Lawrence Berkeley National Laboratory, Berkeley, CA, USA, 94720, Tech. Rep. 1, Jan. 2016, lBNL-1003958.

[92] K. E. Isaacs, A. Giménez, I. Jusufi, T. Gamblin, A. Bhatele, M. Schulz, B. Hamann, and P.-T. Bremer, "State of the Art of Performance Visualization," in *EuroVis - STARs*, R. Borgo, R. Maciejewski, and I. Viola, Eds., vol. 2, no. 3. Chichester, UK: The Eurographics Association, jun 2014, pp. 21–28.

[93] K. Stockinger, J. Shalf, K. Wu, and E. W. Bethel, "Query-Driven Visualization of Large Data Sets," in *Proceedings of IEEE Visualization 2005*, vol. 32, no. 3. Minneapolis, MN, USA: IEEE Computer Society Press, October 2005, pp. 167–174, lBNL-57511. [Online]. Available: http://vis.lbl.gov/Publications/2005/QueryDrivenVis-LBNL-57511.pdf

[94] "Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory," Berkeley, CA, USA, pp. 1–220, apr 2015, last accessed: July 2015; available at http://ccse.lbl.gov.

[95] C. A. Rendleman, V. E. Beckner, M. Lijewski, W. Y. Crutchfield, and J. B. Bell, "Parallelization of Structured, Hierarchical Adaptive Mesh Refinement Algorithms," *Computer and Visualization in Science*, vol. 3, no. 1, pp. 147–157, jun 2000.

[96] B. Lee and W. Martin, "Stacked Graphs, Geometry and Aesthetics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1245–1252, Nov 2008.

[97] R. B. Neale, C. Chen, A. Gettelman, P. H. Lauritzen, S. Park, D. L. Williamson, A. J. Conley, R. Garcia, D. Kinnison, J. Lamarque *et al.*, "Description of the NCAR community atmosphere model (CAM 5.0)," *NCAR Tech. Note NCAR/TN-486+ STR*, vol. 1, no. 1, pp. 1–12, jan 2010.

[98] C. Folland, D. Stone, C. Frederiksen, D. Karoly, and J. Kinter, "The International CLIVAR Climate of the 20th Century Plus (C20C+)," *CLIVAR Exchanges*, vol. 19, no. 2, pp. 57–59, jan 2014.

[99] M. New, M. Hulme, and P. Jones, "Representing Twentieth-Century Space-Time Climate Variability. Part II: Development of 1901-96 Monthly Grids of Terrestrial Surface Climate," *Jounal Climate*, vol. 13, no. 1, pp. 2217–2238, jan 2000.

[100] G. G. W. Services, "El Niño and La Niña Years and Intensities," New York, NY, USA, pp. 1–4, jan 2015, http://ggweather.com/enso/oni.htm, last accessed December 2015.

[101] J. M. Wallace and D. S. Gutzler, "Teleconnections in the Geopotential Height Field during the Northern Hemisphere Winter," *Mon. Wea. Rev.*, vol. 109, no. 2, pp. 784–812, apr 1981.

[102] D. W. J. Thompson and J. M. Wallace, "Annular Modes in the Extratropical Circulation. Part I: Month-to-Month Variability," *Journal Climate*, vol. 13, no. 3, pp. 1000–1016, apr 2000.