VISUAL ANALYTICS APPROACHES TO EXPLORING MULTIVARIATE TIME SERIES DATA

by

Tinghao Feng

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computer Science

Charlotte

2021

Approved by:

____

Dr. Jing Yang

____

Dr. Aidong Lu

____

Dr. Xi Niu

____

Dr. Zachary Justin Wartell

____

Dr. Martha-Cary Eppes

ABSTRACT

TINGHAO FENG. Visual Analytics Approaches to Exploring Multivariate Time Series Data. (Under the direction of DR. JING YANG)

Time-oriented data analysis has attracted the attention of researchers for decades, across many research domains, including but not limited to medical records, business, science, engineering, biographies, history, planning, and project management. However, the complexities of time-oriented data with a large number of variables and varying time scales hinder scientists from completing more than the most basic analyses. In this dissertation, I present two design studies where multivariate time series data are involved. In the first design study, I developed an interactive interface, *t*-RadViz, for a manufacturing company to visually monitor and analyze real-time streaming multivariate testbench data with continuous numeric values. In the second design study, I developed a visual analytics prototype named EVis for analyzing and exploring how recurring environmentally driven events are related to high dimensional time series of continuous numeric environmental variables. In both design studies, I closely collaborated with domain users in the whole process of requirement analysis, design, and evaluation. Besides a rich set of fundamental graphic charts for supporting basic analysis functions, new visual analytics techniques were developed in the design studies for addressing challenging tasks, such as a novel trajectory-based multivariate time series visual analytics approach in EVis for exploring temporally lagging relationships between events and antecedent conditions. The effectiveness and efficiency of the prototypes are illustrated by case studies conducted with real users and feedback from domain experts.

ACKNOWLEDGEMENTS

Throughout the writing of this dissertation I have received a great deal of support and assistance.

I would first like to thank my supervisor, Professor Jing Yang, whose expertise was invaluable in formulating the research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I would also like to thank my tutor, Dr. Martha Cary Eppes, for her valuable guidance throughout my studies. You provided me with the data and basic knowledge of Earth Sciences that I needed to successfully complete my dissertation.

I would like to acknowledge my colleagues from my internship at TRD (Toyota Racing Development) for their wonderful collaboration. I want to thank you for your patient support and for all the opportunities I was given to further my research.

I would like to thank my fellow colleague Yueqi Hu and Zhaocong Yang, who provided stimulating discussions and important support during my research.

Finally, I could not have completed this dissertation without the support of my family for their wise counsel and sympathetic ear. You are always there for me. I would particularly like to single out my son Zhiyu Feng, who provided happy distractions to rest my mind outside of my research.

TABLE OF CONTENTS

LIST OF FIGURES viii

LIST OF ABBREVIATIONS xii

CHAPTER 1: Introduction 1

CHAPTER 2: Related Work 5

2.1. Multivariate Time Series Visualization 5

2.2. RadViz 10

CHAPTER 3: *t*-RadViz: A Real-Time Visual Interface for Monitoring Multivariate Streaming Testbench Data 14

3.1. Introduction 14

3.2. Development Process, Data, and Tasks 18

3.2.1. Development Process 18

3.2.2. Data 18

3.2.3. Tasks and Requirement Analysis 19

3.3. *t*-RadViz Prototype 21

3.3.1. Prototype Implementation and Data Processing 21

3.3.2. socket.io 22

3.3.3. RadViz panel 23

3.3.4. Line Charts 24

3.3.5. Bar Charts 25

3.3.6. Scatterplots 29

3.3.7. Time Series Alignment 29

3.4. Expert Feedback 30

3.5. Conclusion 33

CHAPTER 4: EVis: Visually Analyzing Environmentally Driven Events 35

4.1. Introduction 35

4.2. Background - Rock Mechanical Weathering 38

4.3. Development Process and Requirement Analysis 40

4.3.1. Development Process 40

4.3.2. Requirement Analysis 41

4.4. EVis 42

4.4.1. Data Preprocessing and Concepts 42

4.4.2. Visualization Overview 42

4.4.3. Basic Graphical Charts 44

4.4.4. RadViz 47

4.4.5. RadViz-Leash 51

4.5. Conclusion 59

CHAPTER 5: Evaluation of EVis 60

5.1. Case Studies 60

5.1.1. Case 1: Temperature and Rock Cracking 60

5.1.2. Case 2: Exploratory Analyses and Unexpected Patterns about Wind Speed 63

5.2. Expert Feedback 65

5.3. Discussions 68

5.4. Conclusion 69

CHAPTER 6: EVis 2.0 and a Case Study on Rockfalls at Yosemite National Park 70

6.1. Introduction 70

6.2. Data Description 71

6.3. System Improvement 72

6.3.1. SCanvas, a JS Library of Scalable Canvas 72

6.3.2. Missing data processing 73

6.3.3. Categorical attributes of events 75

6.3.4. New data import interface 76

6.4. Case Study 76

6.5. Conclusion 80

CHAPTER 7: Summary and Future Direction 82

7.1. Summary 82

7.2. Future Direction 84

REFERENCES 86

LIST OF FIGURES

FIGURE 1.1: An abstract overview of the different stages and their transitions in Visual Analytics [1]. The figure is used without the permission of its authors. 2

FIGURE 2.1: A screenshot of TimeSearcher2 [2]. It arranged multiple line charts vertically for visualizing multivariate time series. Segments similar to the sample segment (highlighted in red in the top line chart) were highlighted in the bottom view in red. The figure is used without the permission of the authors. 6

FIGURE 2.2: Animations and trajectories were used to show multivariate time series [3]. The figure is used without a permission of its authors. 7

FIGURE 2.3: The interface of StreamVis [4]. The StreamGraph display showed the whole time series in Theme River [5]. It allowed users to select a certain time range for further analyses. The Time Slice Similarity Plot showed the MDS plot for time slices in the selected time range. The time order was represented by color density with lighter points for earlier time. Users can select a rectangular range in the MDS plot. The selected time slices were highlighted in the StreamGraph plot. The figure is used without a permission of its authors. 8

FIGURE 2.4: Time Curves [6]. Data cases were displayed along a curved timeline. The timeline was folded so that similar data cases were close to each other. Similar data cases were also displayed in similar colors. The right figure showed a long time series displayed in Time Curves. It could be considered as a MDS projection of the data cases, where the projections were connected by a curved timeline according to the temporal order of the data cases. The figure is used without a permission of its authors. 9

FIGURE 2.5: Elzen et al.'s approach to visualizing dynamic networks [7]. In the top figure, snapshots of a dynamic network were projected to points in a 2D space and connected in their temporal order. Projections of similar snapshots were closer to each other in the 2D space. Through the projection, seven stable states (A-G) were detected. The bottom figure showed a representative snapshot for each state and their transitions. The figure is used without a permission of the authors. 10

FIGURE 2.6: Definition of RadViz mapping.[8] 12

FIGURE 3.1: The previous visualization system consisted of traditional line charts and scatterplots. (This figure is from the collaborators with variable names covered for confidentiality.) 16

FIGURE 3.2: The prototype of t-RadViz. A). Control panel; B). Monitor table; C). RadViz panel; D). Scatterplots; E). Line charts. (All dimension names and comments are covered for confidentiality.) 22

FIGURE 3.3: RadViz with four input variables. The right figure only displays points. The left figure displays points and polylines connecting the points, which makes it easier to detect outliers 24

FIGURE 3.4: Left: RadViz with four input variables. Right: RadViz with ten output variables. 25

FIGURE 3.5: RadViz is showing the current test run in live. 26

FIGURE 3.6: The top line chart contains all four input variables. Each of the bottom four line charts and its delta bars are related to a preselected output variable. In the line charts, the current/reference run is displayed in orange/blue. In the bar charts, the delta bars in red/green indicate the current run has a worse/better performance than the reference run. For confidentiality, the names of input are covered and the names of variables of the bottom four line charts are removed. 28

FIGURE 3.7: The line chart for the current test running in live. 30

FIGURE 3.8: The scatterplot for the current test running in live. 32

FIGURE 4.1: The New Mexico Rock Dataset [9] in EVis. (a) The value-time scatterplot of Ambient Temperature Change (ATC). A group of discrete-time-points with negative ATCs are selected as POIs. (b) RadViz with two top surface temperature variables placed on the top and two bottom surface temperature variables placed at the bottom. (c) Histograms showing event hours and event rates of the whole dataset (in blue) and the POIs (in orange). (d) The cluster timeline view where three-hour leashes of the POIs are grouped into eight clusters and displayed over timelines. The clusters are sorted by event rate in descending order. Leashes of non-event hours are dimmed. 36

FIGURE 4.2: Studying POIs with high wind speeds. (a) Heatmaps with Vapor Pressure (outer dimension), ATC (Y-axis), and Wind Speed (X-axis). (b) RadViz with a few well-known triggers as anchor dimensions. Highlighted POIs are well-distributed in this view. (c) Histograms. Bars of high wind speeds are selected. 44

FIGURE 4.3: Studying a leash of interest using similarity search. Red: the focus POI. Orange: POIs with similar leashes to the focus POI. Green: POIs with dissimilar leashes but similar POI positions to the focus POI. (a) The date-diurnal time scatterplot shows orange POIs are distributed around 15:00 pm. (b) RadViz shows different distributions of orange POIs and green POIs. (c) The histograms show orange POIs and green POIs have different event rates in many value ranges. (d) Leashes of orange POIs are displayed in a timeline view and a detail view similar to Fig. 4.7 (not fully displayed due to the page limit). 46

FIGURE 4.4: Rescale the projections. (a) An illustration of how rescaling works. (b) A cluttered RadViz view. (c) (b) rescaled with $s = 5$. 48

FIGURE 4.5: Adjust the weight of a dimension. (a) An illustration of how the distortion works when the weight of the dimension on the top is increased. (b) A skewed RadViz view. (c) (b) after the weight of the dimension on the top is increased to $w = 5$. 50

FIGURE 4.6: A portion of the cluster sequential view (sorted by leash similarity) of POIs at freezing temperatures (leash length = 24 hours). (a) and (b) are zoomed-in views of leashes from different clusters. The POI in (a) is an event hour and the POI in (b) is a non-event hour. 52

FIGURE 4.7: A portion of the detail view of a cluster. The first column is selected for comparison. (a)-(e) are zoomed-in views of several line charts of the Wind Speed variable. (a) is the chart of the selected POI. (b)-(e) compare (a) (gray lines) with charts of other POIs. They reveal high wind speeds might be related to events. 56

FIGURE 6.1: A value-time scatterplot for vapor pressure at Yosemite village (upper) and average daily temperature at Yosemite park headquarter (lower). There are a large number of days with missing values in the upper chart for quite a long period of time. They are displayed on the line above the scatterplot range as hollow bubbles so that analysts can still examine their time and the event information. 74

FIGURE 6.2: An example of the timeline matrix. Rows with a light background are anchor dimensions. The red frame highlights timelines of non-anchor dimensions with missing data. 75

FIGURE 6.3: The triggering condition histogram (orange frame) for selecting events according to triggering conditions. It has a bar for each possible triggering condition whose height represents the count of event days with that trigger. The Y-position of the orange cross on a bar represents the event rate of that trigger, calculated with the enhanced event rate calculation method. All non-event days are represented by a gray bar. 76

FIGURE 6.4: The data import interface. Users can upload a new data set to the EVis 2.0 database and select a dataset from the database for visual exploration. All dimensions of the selected dataset are displayed on the left of the list. Users can add a dimension to the visualization by typing an alias for the dimension on the right column. Users can create new dimensions derived from original dimensions using the entries placed at the bottom of the interface. 77

FIGURE 6.5: The settings for the case study. Top: Events with a precipitation triggering condition were selected using the histogram. Bottom: Events with an unrecognized triggering condition were selected. Left of both figures: The RadViz view. Two temperature anchors were put close to each other. The weight of Vapor Pressure was set to 2 and the weight of Precipitation was set to 4 to make the whole shape balanced. 79

FIGURE 6.6: The clustering result of leashes of events with an unrecognized triggering condition. 79

FIGURE 6.7: The detailed view of cluster *C_2* with a timeline charts matrix. 80

LIST OF ABBREVIATIONS

AE Acoustic Emission

DOI Dimension Of Interest

CHAPTER 1: Introduction

Time is a peculiar data dimension that is common across many application domains, such as manufacturing, Earth science, medical records, business, biographies, history, planning, or project management. An ever-larger body of time-series datasets is generated nowadays from these fields. Following the attention of data analysts, a high interest in mining time-series data is coming along and has led to thousands of papers introducing new algorithms to index, classify, cluster, segment, and predict time series. Lots of data mining algorithms are targeted on time-series data, such as temporal association rules mining and pattern evolution discovery [10]. They are used in tasks such as extracting useful structures [11], finding interesting representations [12], measuring similarity [13], and detecting change points [14].

Besides traditional analytical methods, visual analytics for time-series data has also been introduced. One of the earliest visual representations of time series illustrated planetary orbits. It was created in the 10th or 11th century [15]. In recent decades, data visualization has been employed in analytics processes to inspire researchers, discover patterns, explain ideas, and analyze data [16]. To inspire researchers is not only to superficially wow them but also to really engage people into deeper thinking. Visual perception is always prioritized on attracting people's attention, which motivates people to find patterns from data visualized. Human beings are a kind of visual creature, and a picture is worth a thousand words [17]. Data visualization is often used to explain some complex ideas, phenomenon, or processes through graphic representations. A well-designed interactive visualization can efficiently and effectively help researchers find patterns behind numbers. The main goal of data analysis is to extract information from data with the purpose of answering questions and under-

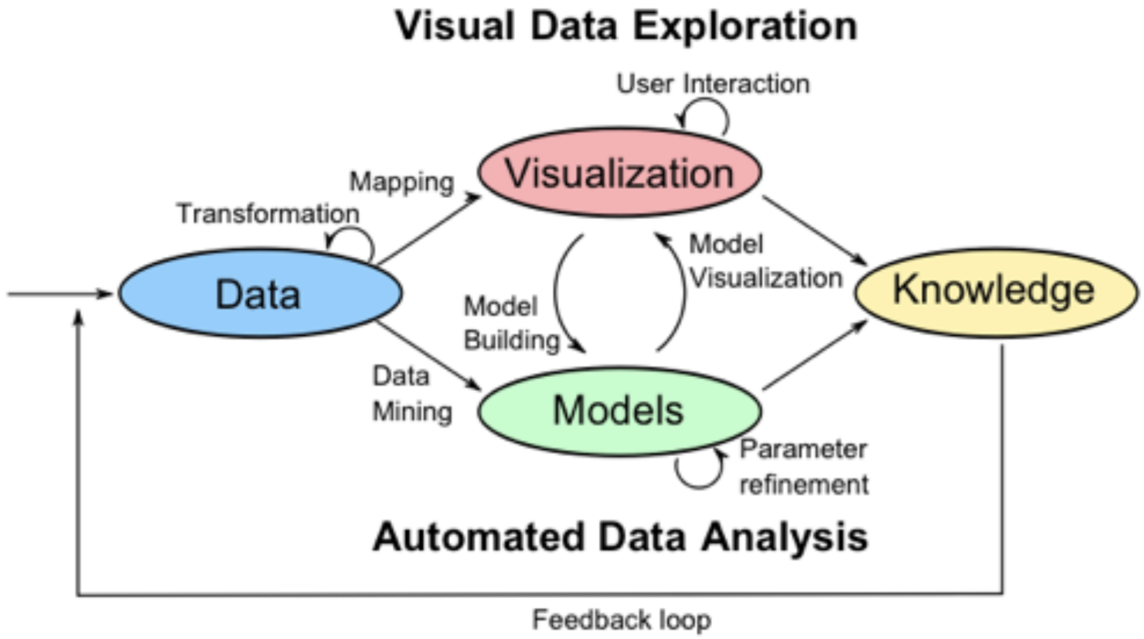


Figure 1.1: An abstract overview of the different stages and their transitions in Visual Analytics [1]. The figure is used without the permission of its authors.

standing phenomena of interest. Scientists have defined Visual Analytics as "a science of analytical reasoning supported by interactive visual interface" [18], which "combines automatic and visual analysis methods with a tight coupling through human interaction in order to gain knowledge from data" (Figure 1.1).

In my dissertation, I focus on visual analytics of multivariate time-series data, where each time primitive is associated with multiple data values [19]. Many analysis methods and algorithms have been developed for multivariate time series data [20]. Also, visualization of multivariate time series data is a flourishing research area. Many visual analytics approaches have been proposed to help analysts explore patterns of multivariate time series data. For example, TimeWheel is used to explore temporal distributions of multiple attributes [21]; ThemeRiver uses the metaphor of a river that flows through time to reveal thematic changes in a document collection [5]; LifeLines uses horizontal bars to show temporal locations and duration of health-related incidents that are related to several facets of patient information [22].

However, when encountering a specific domain application, existing techniques cannot always satisfy the analysis requirements. It means specific visual analytics approaches needed to be developed for the specific data and tasks. In this dissertation, I will present two design studies for two domain applications and show how to develop visualization methods to assist domain experts in conducting time-series data analyses.

The first design study is to develop an interactive visual interface to monitor and analyze live streaming data generated from a testbench of a vehicle manufacturer. The data is time series containing multiple input and output variables that need to be monitored and compared in live when experiments are running. In this study, I have worked closely with domain experts from the manufacturer and developed a working prototype that has been evaluated on-site in real experiments. To monitor and compare multivariate time-series data from multiple experiments, the prototype projects multivariate time series to 2D trajectories using a dimensionality reduction method named RadViz [23]. The trajectories are used to visually compare and automatically align the settings and performances of a running experiment with a baseline experiment. The prototype also integrates other interactive visualizations for experiment monitoring and comparison.

The second design study is to develop a visual analytics prototype for analyzing rock mechanical weathering data. I have collaborated with an Earth scientist on this project. She has studied rock mechanical weathering for more than 12 years. To study the impact of environmental conditions on rock's mechanical weathering, she set up an instrumentation system with sensors affixed on the surface of granite rock to collect environmental data and weathering data. Inside cracking of the rock is "listened" by the sensors as AE (Acoustic Emission) events. The information was collected per minute for about three years. It formed a large time-series dataset, including a time sequence of AE events as well as multi-dimensional time series of environmental conditions. I developed a prototype named EVis to visually analyze this data set for exploring the relationship between the environmental conditions and the AE events. It uses a set of coordinated visualizations to support the goals mentioned above.

EVis provides customized scatterplots, histograms, and heatmaps to conduct foundational analyses that are common across virtually all Earth sciences applications. They allow users to explore the relationships between AE events and one, two, or three environmental conditions. In addition, through interactions provided in these basic visualizations, users can select subsets of interesting AE events for further analyses. EVis also provides a RadViz [23] view to allow users to explore the relationships between AE events and multiple environmental conditions. To address the challenging task of exploring the temporally lagging relationships between AE events and their antecedent environmental conditions, a novel RadViz-Leash approach has been proposed and implemented in EVis. **Leashes** are RadViz projections of multivariate time series segments recording environmental conditions before the AE events happened. By calculating similarities among leashes, clustering leashes based on their similarities, and visually presenting the resulting clusters, EVis provides an effective and efficient way to explore and analyze the temporally lagging relationships between a large number of AE events and their antecedent environmental conditions.

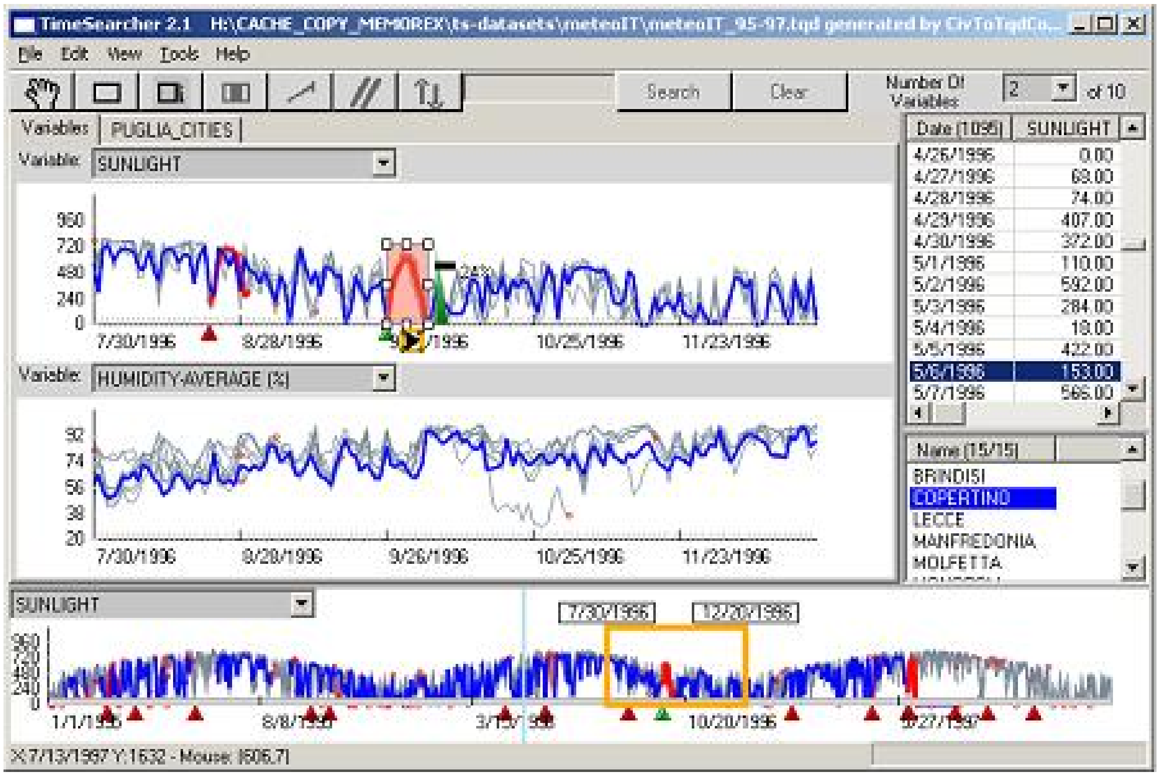
In Chapter 2, I will introduce related work on multivariate time series visualization. The design study for streaming testbench data is introduced in Chapter 3. In Chapter 4, EVis, the prototype resulting from the design study of rock mechanical weathering data, is introduced. Chapter 5 introduces the evaluation of EVis and in Chapter 6 I propose the updated version EVis 2.0 that improves the adaptability of EVis and evaluates it using the Yosemite rockfall data. Then I summarize the whole work and discuss the future direction and plan on further analysis in Chapter 7.

CHAPTER 2: Related Work

2.1 Multivariate Time Series Visualization

Many efforts have been made to visualize a collection of multivariate time series. For example, TimeSearcher2 [2] presented time series with multiple variables in a set of line charts, one for each variable, and placed them vertically side by side sharing the same horizontal timeline (see Figure 2.1). It provided a similarity search interaction where line chart segments similar to a user selected sample segment are automatically selected and highlighted (see Figure 2.1 for an example). This approach was intuitive, but it only applied similarity search on univariate line charts of single dimensions and did not explicitly reveal the temporal relationships among the dimensions.

Many researchers used dimensionality reduction (DR) [24, 25] when visualizing multivariate time series. For example, Fujiwara et al. [24] projected a group of multivariate time series to 2D points using a Two-step Dimension Reduction (TDR) process (the first DR used Principal Component Analysis (PCA) [26] along a variable mode for dimensions and the second DR used Uniform Manifold Approximation and Projection (UMAP) [27] along a time mode for time points). Users could examine feature contributions of selected clusters in the projection space. A drawback of this approach was that each time series is reduced to one point in the projection space. It was not intuitive for users to conduct tasks where temporal trends and temporal relationships need to be examined for a large number of time series. Yang et al. [28] extracted a set of features to characterize a large number of short 1D time series and projected the time series to a PCA biplot based on these features. From this biplot, users could identify clusters of time series and learn correlations among the features. Takami and Takama [3] used animations and trajectories to show how values



Number Of Variables: 2 of 10

Variable: SUNLIGHT

Variable: HUMIDITY-AVERAGE (%)

SUNLIGHT

Date (1096)	SUNLIGHT
4/26/1996	0.00
4/27/1996	69.00
4/28/1996	74.00
4/29/1996	407.00
4/30/1996	372.00
5/1/1996	110.00
5/2/1996	592.00
5/3/1996	284.00
5/4/1996	19.00
5/5/1996	422.00
5/6/1996	153.00
5/7/1996	566.00

Name (15/15)
BRINDISI
COPERTINO
LECCE
MANFREDONIA
MOLFETTA

X:7/13/1997 Y:1632 - Mouse: (606,7)

Figure 2.1: A screenshot of TimeSearcher2 [2]. It arranged multiple line charts vertically for visualizing multivariate time series. Segments similar to the sample segment (highlighted in red in the top line chart) were highlighted in the bottom view in red. The figure is used without the permission of the authors.

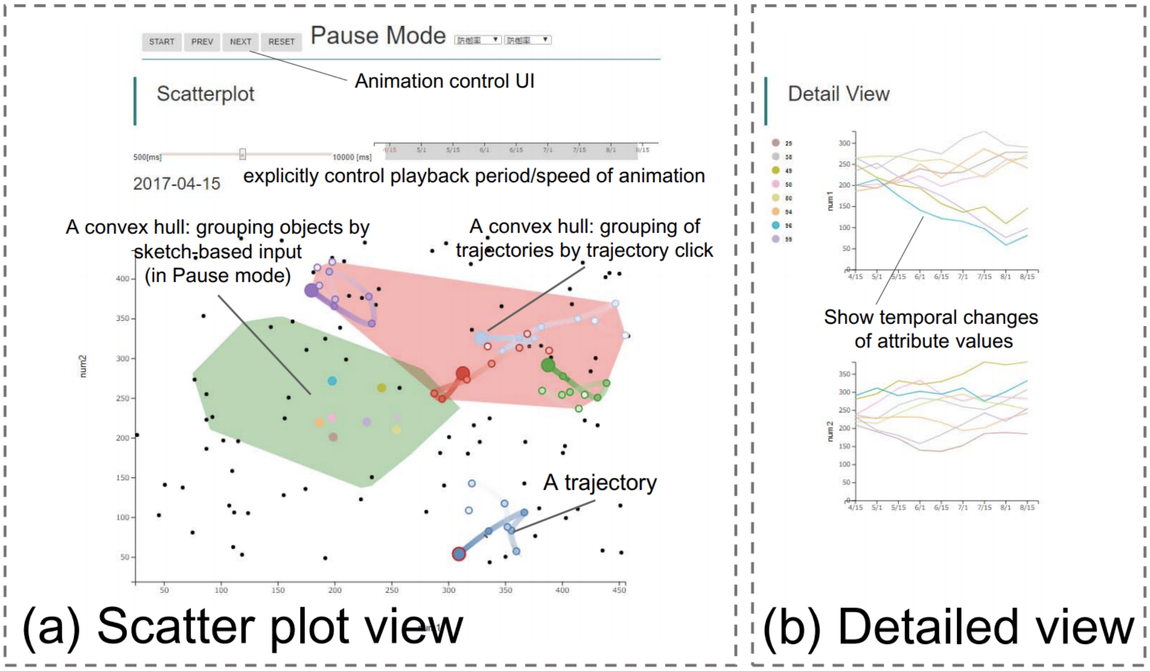
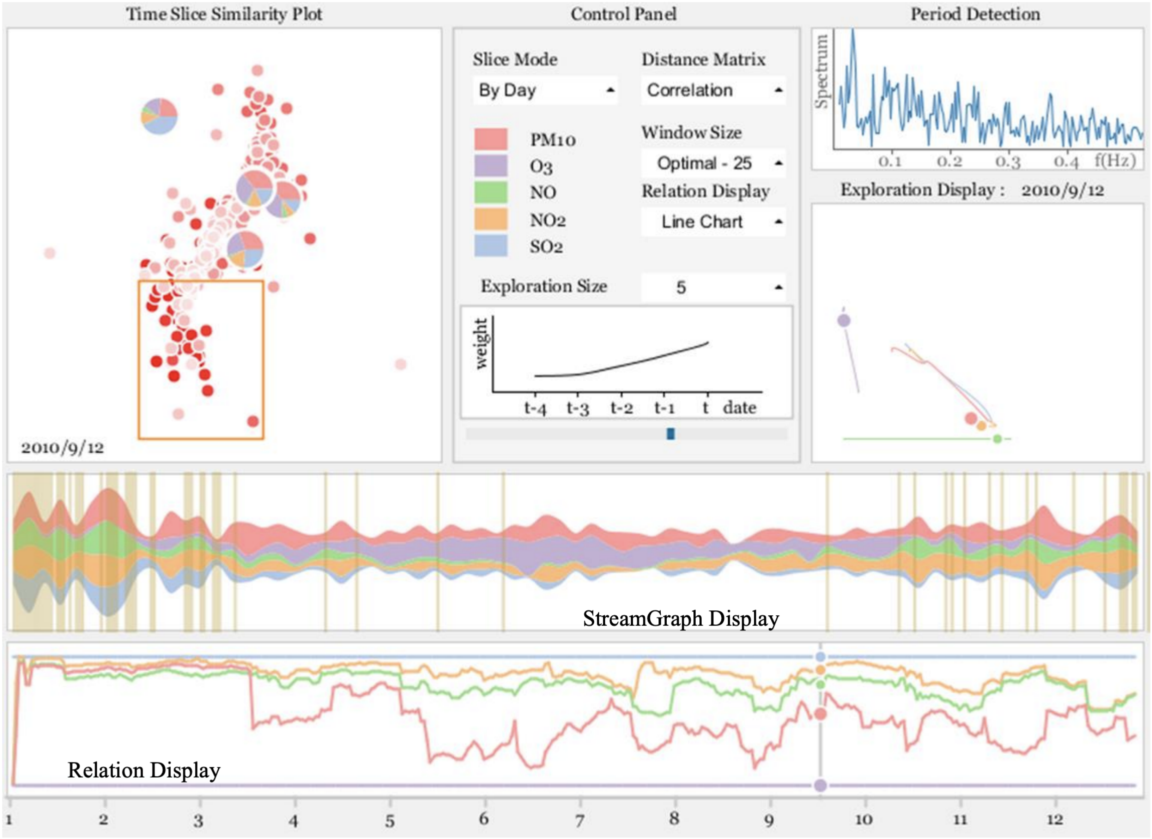

START
PREV
NEXT
RESET
Pause Mode
Animation control UI
Scatterplot
500[ms]
10000 [ms]
2017-04-15
explicitly control playback period/speed of animation
A convex hull: grouping objects by sketch-based input (in Pause mode)
A convex hull: grouping of trajectories by trajectory click
A trajectory
num2
400
350
300
250
200
150
100
50
50
100
150
200
250
300
350
400
450
Detail View
Show temporal changes of attribute values
num1
num2
(a) Scatter plot view
(b) Detailed view

Figure 2.2: Animations and trajectories were used to show multivariate time series [3]. The figure is used without a permission of its authors.

of multiple objects change over time in a shared 2D space. For example, in Figure 2.2, the scatter plot view showed how the projections of multiple objects changed over time. A trajectory of one object was displayed. The detailed view showed the line charts of the two dimensions displayed in the scatter plot view. They revealed how the individual objects changed over time on these dimensions. The drawbacks of this approach were the potential clutter and change blindness problems when many objects were visualized.

StreamVis [4] divided a multivariate time series into time slices. It then projected the time slices to a Multi-Dimensional Scaling (MDS) [29] plot based on their similarities. The temporal order of the time slices was mapped to the color density of the projections. Figure 2.3 shows a screenshot of StreamVis.

Time Curves [6] projected temporal multivariate data cases through a time line to a 2D MDS [29] space and connected them by a time curve according to their temporal order. Figure 2.4 shows an example of Time Curve. Similarly, Elzen et al.



Slice Mode: By Day

Distance Matrix: Correlation

Window Size: Optimal - 25

Relation Display: Line Chart

Exploration Size: 5

Figure 2.3: The interface of StreamVis [4]. The StreamGraph display showed the whole time series in Theme River [5]. It allowed users to select a certain time range for further analyses. The Time Slice Similarity Plot showed the MDS plot for time slices in the selected time range. The time order was represented by color density with lighter points for earlier time. Users can select a rectangular range in the MDS plot. The selected time slices were highlighted in the StreamGraph plot. The figure is used without a permission of its authors.

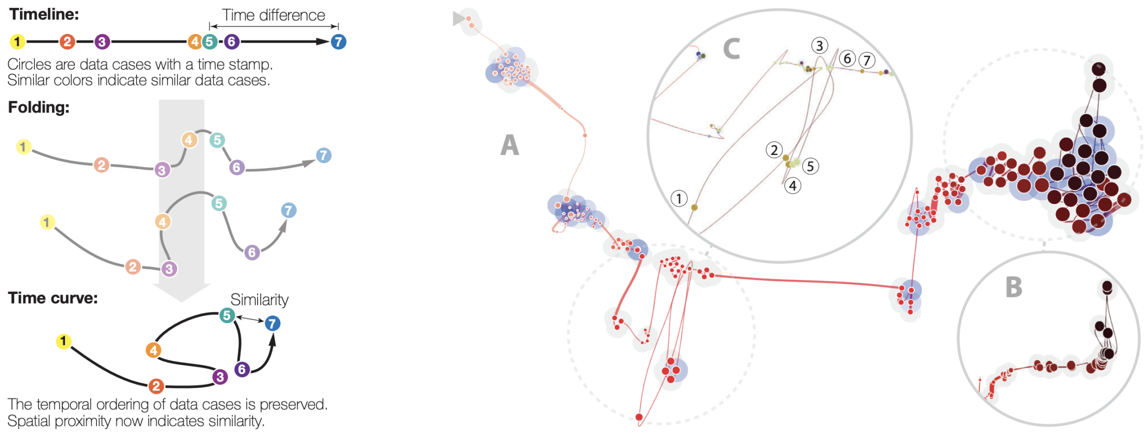

Timeline:
Time difference
1
2
3
4
5
6
7
Circles are data cases with a time stamp.
Similar colors indicate similar data cases.
Folding:
Time curve:
Similarity
The temporal ordering of data cases is preserved.
Spatial proximity now indicates similarity.
A
B
C

Figure 2.4: Time Curves [6]. Data cases were displayed along a curved timeline. The timeline was folded so that similar data cases were close to each other. Similar data cases were also displayed in similar colors. The right figure showed a long time series displayed in Time Curves. It could be considered as a MDS projection of the data cases, where the projections were connected by a curved timeline according to the temporal order of the data cases. The figure is used without a permission of its authors.

[7] discussed different DR approaches for snapshots of a dynamic network. Figure 2.5 shows an example where the snapshots of a dynamic network were projected to a 2D space and connected by a line. The dime dimension was also represented by color in this figure. A representative network of each stable state was also displayed in a temporal order.

Schreck et al. proposed a trajectory-based financial time series visualization [30]. It plotted 2D time series as trajectories in a 2D space, extracted high dimensional feature vectors from the trajectories, clustered the trajectories using a Self-Organizing Map [31] based on the feature vectors, and provided a rich set of visualizations and interactions to explore the clusters. A limitation of this approach was that it was designed for 2D time series rather than multidimensional time series. I also proposed a trajectory-based time series visualization approach. Different from Schreck et al. [30], my approach targets multivariate time series, projects them from a multidimensional space to a 2D space to construct the trajectories, and uses a trajectory similarity measure for clustering. Besides providing overview for the clusters, my work also

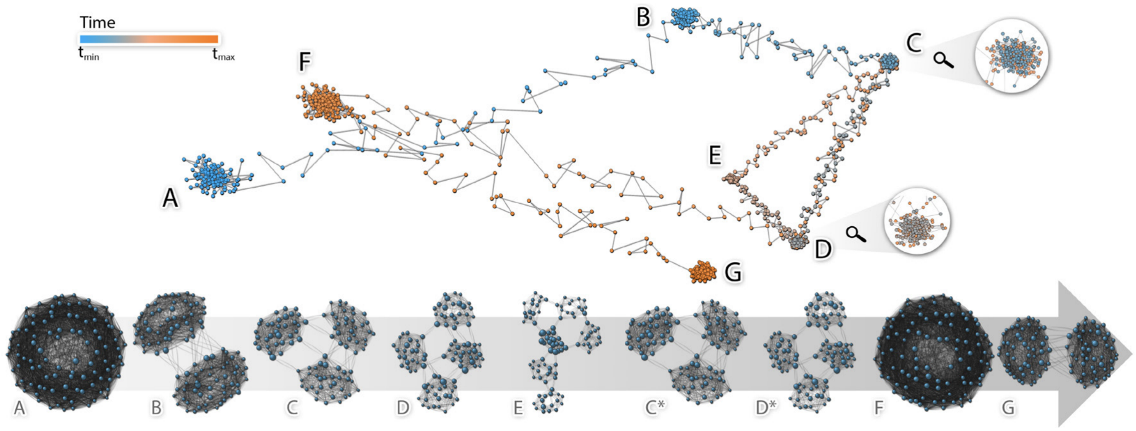


Figure 2.5: Elzen et al.'s approach to visualizing dynamic networks [7]. In the top figure, snapshots of a dynamic network were projected to points in a 2D space and connected in their temporal order. Projections of similar snapshots were closer to each other in the 2D space. Through the projection, seven stable states (A-G) were detected. The bottom figure showed a representative snapshot for each state and their transitions. The figure is used without a permission of the authors.

provides a view allowing users to examine full details of the trajectories in the original multidimensional space.

Clustering approaches have been developed for multivariate time series. Recent examples include model-based clustering [32], subspace clustering [33], and shape-based clustering [34, 35]. For example, Ghassempour et al. [32] proposed an approach to clustering time series with both categorical and continuous variables based on Hidden Markov Models[36]. Dasu and Swayne [37] clustered time series based on nonparametric statistical summaries using a k-means algorithm [38]. Different from these approaches, my approach clusters time series based on trajectory similarity in a 2D projection space so that the clustering result is consistent with what users see from the visualization and domain knowledge integrated into the dimension reduction process is reserved.

2.2 RadViz

RadViz (Radial Coordinate Visualization) is a visualization technique for analyzing multi-dimensional data [23]. After placing dimensions along the circumference of a circle as anchors, it projects a multidimensional dataset to a 2D space. The projection of a data point is solely decided by the anchor positions, the value range used for normalization of each dimension, and the values of the data point itself. This feature makes RadViz a good choice for generating trajectories of multivariate time series data, where new data points coming in will not change the layout of existing data points. This feature distinguishes it from other dimension reduction methods such as MDS [29, 39], PCA [40, 25], t-SNE [41], and UMAP [27], where the projection of a data point is also decided by the values of other data points.

RadViz maps data points from an n-dimensional space to a 2D plane, inside a circle usually. All dimensions are normally situated around a circle as anchors. It is supposed that each anchor holds its own virtual spring of variable stiffness and all the loose ends of the springs are bound together (Figure 2.6).

Consider $[\overrightarrow{S}_1, ..., \overrightarrow{S}_n]$ are vectors for n anchors around the circle. To decide the RadViz position $\overrightarrow{u}$ of a point $[d_1, ..., d_n]$ from the n-dimensional space, set the stiffness of the virtual spring of each anchor j to corresponding value of the dimension d_j, and then apply the Hooke's law of mechanics to get equation (2.1) that all the spring forces reach balance (sum to 0). Finally, the position of $\overrightarrow{u}$ is given by equation (2.3) [8].

$$\sum_{j=1}^{n}(\overrightarrow{S}_j - \overrightarrow{u})d_j = 0 \tag{2.1}$$

$$\sum_{j=1}^{n} \overrightarrow{S}_j d_j = \overrightarrow{u} \sum_{j=1}^{n} d_j \tag{2.2}$$

$$\overrightarrow{u} = \frac{\sum_{j=1}^{n} \overrightarrow{S}_j d_j}{\sum_{j=1}^{n} d_j} \tag{2.3}$$

RadViz allows users to see data points in relation to their attributes, which is a feature distinguishing it from other dimension reduction approaches. PCA [40] and

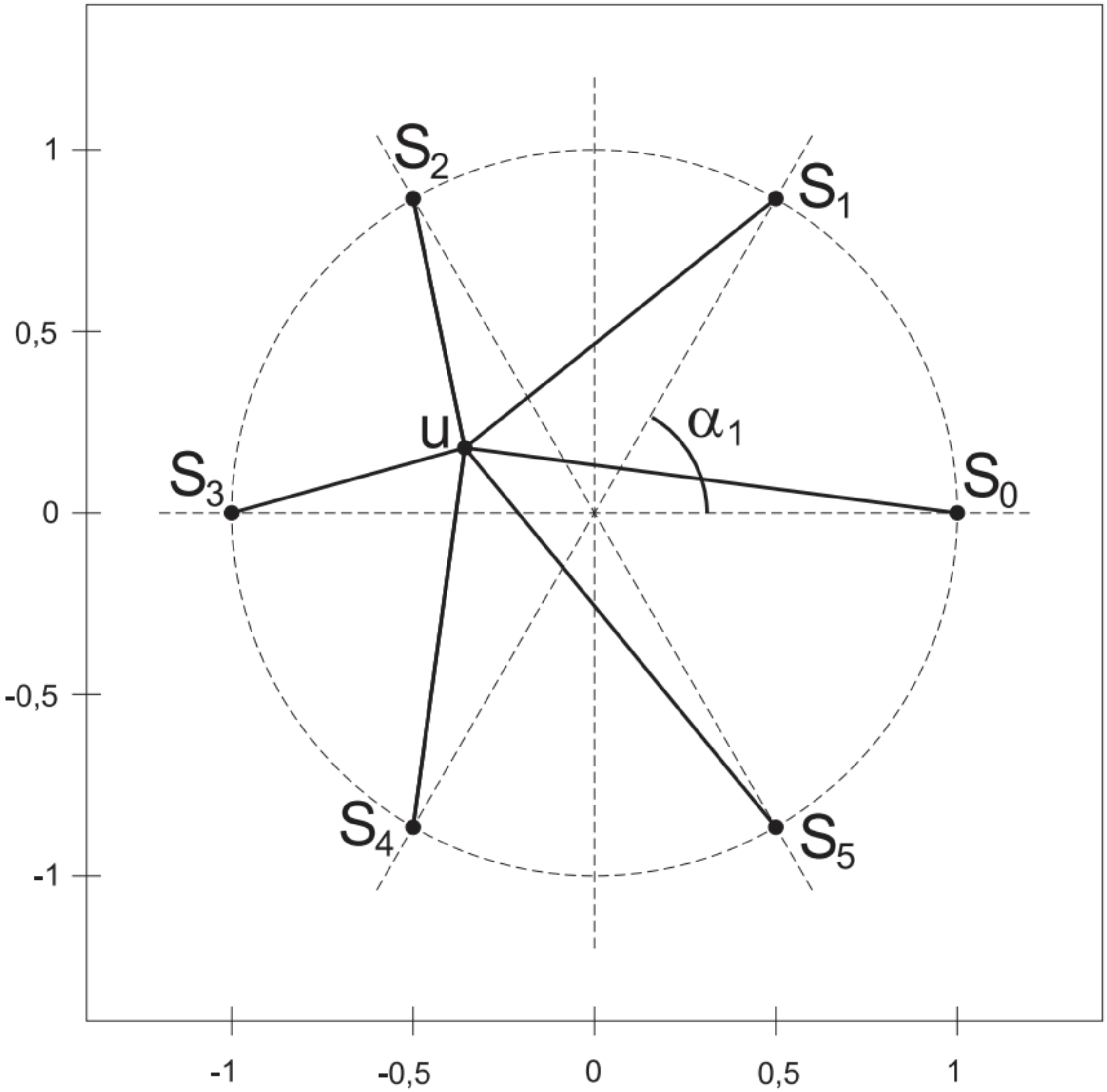


Figure 2.6: Definition of RadViz mapping.[8]

related displays, such as biplots [42], are not optimized for this type of tasks [43]. MDS [29, 39] and t-SNE [41] also cannot support this type of tasks since they lost the data-dimension relationships in the projection process [43]. This feature makes RadViz an ideal approach for visualizations designed for domain applications, where domain experts would like to see data points in relation to their attributes in a projection space.

Many efforts have been made to improve the original RadViz design to address problems such as clutters in the center of the RadViz space. Cheng et al. [43] introduced RadViz Deluxe, which automatically relocated dimension anchors for clutter reduction and insight discovery. They employed several optimization procedures to enforce a variety of distance constraints. Angelini et al. [44] proposed a method to automatically rearrange dimension anchors based on a point disposition heuristic. Their prototype also allowed dragging the anchors freely along the circumference. Zhou et al. [45] designed rich interactions in RadViz, such as adding and removing dimensions as well as selecting data from RadViz and examining them in a coordinated detail view. Their case studies revealed the benefits of providing interactions in RadViz. We provide a rich set of interactions in RadViz in our system. Rather than using automated anchor dimension relocation algorithms, we propose two interactive distortion techniques to reduce clutter in RadViz, which avoids changing the anchor layout created by scientists since it may carry their domain knowledge and hypotheses.

CHAPTER 3: t-RadViz: A Real-Time Visual Interface for Monitoring Multivariate Streaming Testbench Data

3.1 Introduction

In this design study, I have worked closely with an automobile manufacturing company to design, develop, and evaluate a visual interface to allow testbench operators to monitor results of running testing for optimizing vehicle designs. Since feedback from the company to this visual interface was highly positive, it has been employed by the company in multiple work groups.

When a vehicle is designed, a series of tests and adjustments are always necessary for the best setting[46]. For the final adjustment of a design, a testbench is usually used to simulate a real situation. A testbench usually consists of multiple runs of testing, in each of which a vehicle being tested has different design parameters. In each run of the testing, testbench operators continuously adjust the body positions of the vehicle (input variables to be explored) following the same predefined plan by engineers and measure its aerodynamics parameters (output variables to be explored). The positions and aerodynamics parameters are collected in real-time through a set of sensors during the testing. The aerodynamics parameters of the vehicle at the same position at different runs are compared to learn cons and pros of the design parameters used in each run. The adjustment of the design parameters is usually subtle and incremental during a testbench.

A critical task of testbench operators is to monitor the input and output variables in real-time and compare a running test (referred to as current run) against one or more reference runs. By monitoring the input variables and comparing them with those in the reference runs, the engineers verify that the positions of the vehicle are adjusted as planned so that the outputs of these runs are comparable. If an unexpected error that may fail the current run is detected, the operators can terminate the current run before it ends to reduce cost. By comparing the outputs of the current run with the reference runs on the fly, the operators can learn whether the new adjustments they made to the vehicle in this run is good or bad. If they observe that the performance of the vehicle in the running test is worse than the reference runs, they may also terminate the current run before it ends to reduce cost.

Before using the visual interface I developed, the company conducted this task using a visualization system consisted of line charts and scatterplots (see Fig 3.1). The line charts showed how the input variables change over time. Users could turn on a vertical line and move it along the line charts to view the status for all input variables at a specific time. Users could also add one reference run to the line charts to compare it with the run under investigation. Several scatterplots were used to check the relationships between any two selected dimensions. Users could examine those relationships in a current run or compare the relationships in a current run with one or more reference runs using the scatterplots.

The testbench operators were not satisfied with their previous visualization system for several reasons. First, it didn't provide any visual aid to help the operators compare multiple line charts in a real-time environment. It was difficult for the operators to simultaneously examine multiple line charts to compare a current run with reference runs to identify errors. Second, it didn't handle delays in the tests, which are common in testbenches. The delays caused unsynchronized line charts, where the same input settings of different runs were not displayed at the same time point (X position) for effective comparsion. Third, the operators needed to check multiple line charts to get a comprehensive picture about whether a current run was comparable to the reference runs according to the input variables, which was overwhelming.

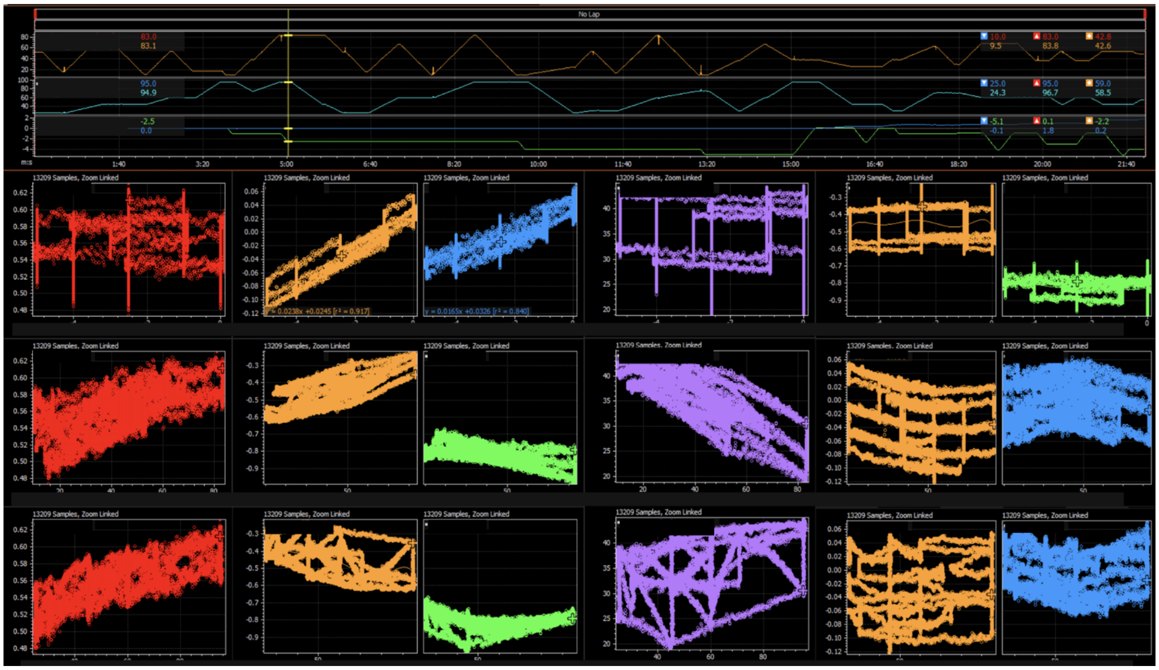


Figure 3.1: The previous visualization system consisted of traditional line charts and scatterplots. (This figure is from the collaborators with variable names covered for confidentiality.)

In addition, as shown in Figure 3.1, the scatterplots were very cluttered when all data posts in one or more runs were plotted. It was difficult to get a comprehensive view of how the input variables were related to the output variables from the cluttered scatterplots. The operators were able to filter the data by selecting line segments in the line charts and only showing the corresponding data in the scatterplots, but it was difficult for them to manually select synchronized matching segments in other runs after a segment in one run was selected. Therefore, the company had a strong need for a new visualization system that can better help the operators.

To address this needs, I have worked toward a visual interface for monitoring real-time streaming testbench data. I have designed, developed, and evaluated a fully working prototype named *t*-RadViz. It works in a streaming mode. It keeps receiving new data streaming in from the testbench. The visualization highlights the newest burst while providing a context of historical data. It allows users to analyze input path development, variations of output, and compare input and output between a current run and one or more reference runs on the fly. This prototype has been employed by the manufacturing company after it received positive feedback in evaluations with several runs of real testbench experiments.

t-RadViz adopts RadViz, an existing multivariate visualization technique, to visualize multivariate time series consisting of selected input and output variables. To the best of my knowledge, my work is among the first efforts that use RadViz to visualize multivariate time series data. When only input variables are selected, the RadViz view provides an overview of parameter change paths and allows users to compare the input parameters between multiple runs of testing. When both input and output variables are selected, the RadViz view provides an overview of the differences among multiple runs in performance at different input settings.

Besides the RadViz view, the prototype also provides line charts and scatterplots. Similar to the previous practice of the company, the line charts allow users to examine the temporal evolution of selected variables and compare the same variables in multiple runs. Each scatterplot allows users to examine the relationship between a selected input variable and a selected output variable. Different from the previous practice, *t*-RadViz only displays the most recent time series segments in a current run and its matching segments in one or more reference runs to reduce the cognitive load of engineers. A new algorithm has been developed to search matching segments on the fly using projections of input paths in the RadViz space.

In addition, the prototype provides delta histograms and table views. The delta histograms are set attached to the line charts to highlight the difference between the current run and a reference run in real-time. The table view displays detailed information in digital form in a way that engineers and operators are accustomed to. I have applied color coding on the table to enhance the reading efficiency.

3.2 Development Process, Data, and Tasks

3.2.1 Development Process

This design study was conducted from 2020 Summer to 2021 Spring. During this period of time, I conducted bi-weekly meetings with target users of this prototype, namely testbench operators and engineers from the manufacturing company. Along the process, I also participated in multiple experiments conducted by the users in the testbench lab to test the prototype with real data in a real working environment. I developed the prototype iteratively with multiple designs and evaluation cycles. Typically, in each cycle, a version of the prototype was developed based on input and feedback from the users. A few meetings were conducted where I presented the prototype with test data to the users for their feedback. After the prototype met their expectations, the prototype was tested on the spot. During and after the real-time test, I met with the users to collect input and feedback for the next cycle.

3.2.2 Data

Each testbench experiment consists of multiple runs. Before each run, the engineers physically fine-tune a vehicle being tested. The goal of the experiment is to find the best design parameters with which the vehicle has the best performance.

When a run of testing starts, the vehicle runs in the lab. The operators change its position along a pre-defined input changing path to study how the fine-tuning of the vehicle affects the driving performance under different positions. The lab collects the position and aerodynamics measures of the vehicle through a set of sensors at 10 Hz, or say, 10 multivariate data points are generated per second. Each data point consists of position measures and aerodynamics measures, as well as a timestamp. During the test, the data collected by sensors is written to a hard drive to form a streaming multivariate time series. In an example experiment, each run lasted about 15 minutes. The time series data for each run contains about 9,000 rows and 46 columns/dimensions (including timestamps, all input variables, all output variables, and errors for input variables) at the end of the runs. The experiment had 8 test runs. So, it resulted in 8 such time series.

Note that the datasets contain both independent variables and dependent variables. The vehicle position measures are independent variables controlled by the operators. We also call them input variables. The aerodynamics measures are dependent variables. We also call them output variables. They change when the vehicle position changes. A hidden independent variable is the design parameters of the vehicle, which are different in each run of testing. The engineers are not only interested in how the aerodynamics measures change when the position changes, but also how the aerodynamics measures are different under the same position settings in different runs, and how the differences change when the position settings change. Although the runs in a testbench experiment usually follow the same position change plan, the time ranges of the runs vary - random delays happen in position adjustments within the runs, which are common and unpredictable in testbench experiments.

3.2.3 Tasks and Requirement Analysis

In the bi-weekly meetings and lab experiments, the testbench operators and engineers explained how a testbench works and what kind of analyses they conduct. According to them, the data is analyzed while a test is running. Multivariate time series are generated in a stream. The testbench operators need to make real-time decisions based on the results of the analyses. For example, if they discover that the performance of a vehicle in a running test is worse than a previous run, they may terminate the run immediately to reduce cost. The tasks of interest are summarized below:

R1 - Monitor a run to make sure it is conducted smoothly. If the operators observe anomalies, they may need to stop the test to address the problems. To conduct this task, the operators need to monitor how the input and output variables change over time. Additionally, in the streaming mode, every time when new data comes in, called a **burst**, the average delta, which is the difference of each dimension between the current burst and respective data of a reference run, for any dimension need to be highlighted for the judgment of the current moment.

R2 - Compare the input of a running test with the input of one or more reference runs. The operators need to know whether the position changes are conducted along a similar path as in the reference runs. If the path is significantly different from the reference runs (e.g., there are unexpected errors), the operators will not be able to compare the results of the current run with the reference runs. The operators may need to terminate the current run immediately to reduce cost.

R3 - Compare the output of a running test with the output of one or more reference runs. The operators need to know whether the tuning conducted for a run has positive or negative effects on the performance. Toward this goal, they may compare the output variables of multiple runs in either the whole input space or segments of the input path of interest. The results will help the operators understand how the tuning affects the performance of the vehicle in different positions. They will also help them decide how to further tune the vehicle in the tests to be conducted. For example, if the vehicle was turned along a direction in the most recent run and the performance was better than other runs, the operators may further tune the vehicle along with that direction. If the tuning has shown clear negative effects in a current run, the operators may terminate it immediately to reduce cost.

R4 - Examine temporally evolving relationships between the input variables and the output variables. A basic observation required during a test is how an adjustment of input variables impacts the output. The operators also need to observe the trend of this relationship.

To support these tasks, a visual interface designed for the target users must satisfy the following requirements:

1) Since the users are not visualization experts and their decisions are made in a time-critical environment, the visualizations need to be intuitive and straightforward.

2) The delays in the runs bring noises to the data analyses. Their impacts on the visualizations need to be minimized.

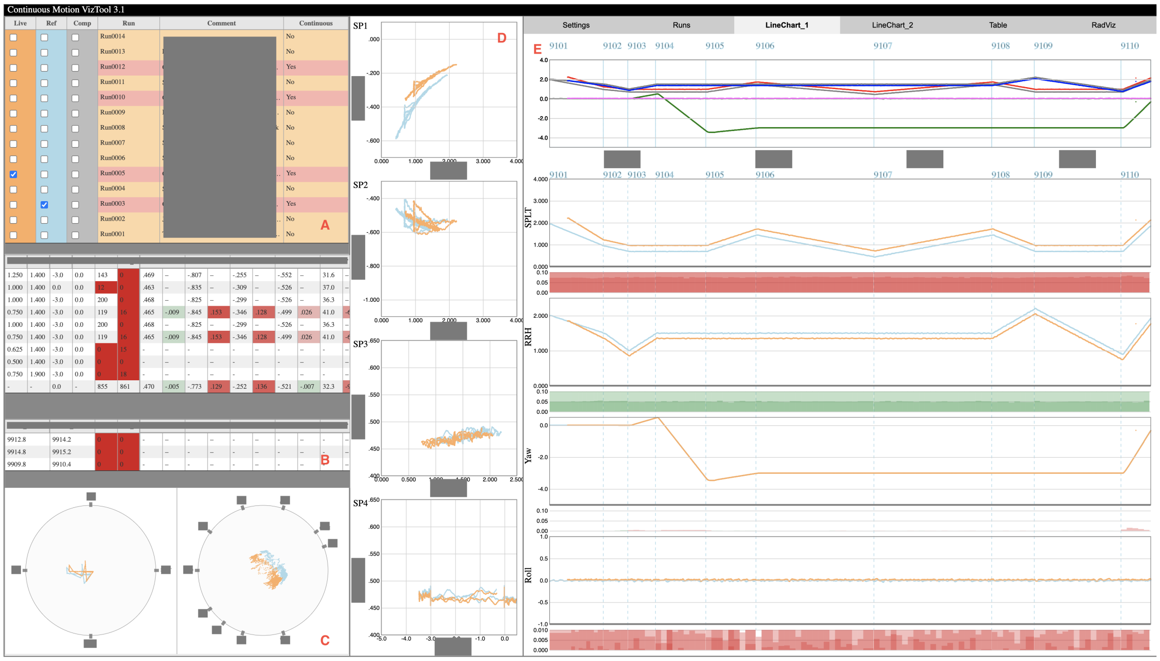
3) The interests of the users may vary from time to time. The visual interface should allow them to customize the visualizations to fit their drifting needs.

3.3 *t*-RadViz Prototype

I have developed *t*-RadViz, a visual interface for monitoring the streaming testbench data. It allows operators to monitor and analyze testbench experiments in real-time. The interface consists of four parts: a control panel to manage finished and current runs and set reference runs for comparison (Figure 3.2A), a monitor table to show real-time status based on presetting of specific input values (Figure 3.2B), a RadViz panel to visualize and compare multiple runs with multiple variables simultaneously considered (Figure 3.2C), a set of scatterplots to show how relationships between pairs of variables of a live run change over time and how they are different from reference runs (Figure 3.2D), and a set of line charts to monitor how single variables of a live run change over time and how their values are different from reference runs (Figure 3.2E). Since the time series have absolute timestamps and are not synchronized with regard to their test plan, an algorithm has been developed to identify and align matching input path segments of a current run and its reference runs.

3.3.1 Prototype Implementation and Data Processing

For better flexibility and interactivity, the client of *t*-RadViz is written by HTML and JavaScript, which can be run on any browser. The raw data is binary data (0 and 1) updated at 10Hz generated by sensors from a lab. To collaborate with *t*-RadViz, the raw data is transformed into a CSV file using Matlab. The CSV file



Continuous Motion VizTool 3.1

Live	Ref	Comp	Run	Comment	Continuous
□	□	□	Run0014	[redacted]	No
□	□	□	Run0013	[redacted]	No
□	□	□	Run0012	[redacted]	Yes
□	□	□	Run0011	[redacted]	No
□	□	□	Run0010	[redacted]	Yes
□	□	□	Run0009	[redacted]	No
□	□	□	Run0008	[redacted]	No
□	□	□	Run0007	[redacted]	No
□	□	□	Run0006	[redacted]	No
☑	□	□	Run0005	[redacted]	Yes
□	□	□	Run0004	[redacted]	No
□	☑	□	Run0003	[redacted]	Yes
□	□	□	Run0002	[redacted]	No
□	□	□	Run0001	[redacted]	No

[redacted]															
1.250	1.400	-3.0	0.0	143	0	.469	–	-.807	–	-.255	–	-.552	–	31.6	–
1.000	1.400	0.0	0.0	12	0	.463	–	-.835	–	-.309	–	-.526	–	37.0	–
1.000	1.400	-3.0	0.0	200	0	.468	–	-.825	–	-.299	–	-.526	–	36.3	–
0.750	1.400	-3.0	0.0	119	16	.465	-.009	-.845	.153	-.346	.128	-.499	.026	41.0	-6
1.000	1.400	-3.0	0.0	200	0	.468	–	-.825	–	-.299	–	-.526	–	36.3	–
0.750	1.400	-3.0	0.0	119	16	.465	-.009	-.845	.153	-.346	.128	-.499	.026	41.0	-6
0.625	1.400	-3.0	0.0	0	15	-	–	-	–	-	–	-	–	-	–
0.500	1.400	-3.0	0.0	0	0	-	–	-	–	-	–	-	–	-	–
0.750	1.900	-3.0	0.0	0	18	-	–	-	–	-	–	-	–	-	–
-	-	0.0	-	855	861	.470	-.005	-.773	.129	-.252	.136	-.521	-.007	32.3	-9

[redacted]													
9912.8	9914.2	0	0	-	–	-	–	-	–	-	–	-	–
9914.8	9915.2	0	0	-	–	-	–	-	–	-	–	-	–
9909.8	9910.4	0	0	-	–	-	–	-	–	-	–	-	–

Figure 3.2: The prototype of *t*-RadViz. A). Control panel; B). Monitor table; C). RadViz panel; D). Scatterplots; E). Line charts. (All dimension names and comments are covered for confidentiality.)

keeps updating at a consistent frequency and is stored at an accessible file storage server.

3.3.2 socket.io

To work with the real-time updating data, I used socket.io [47] to rebuild the communication between server and client ends. Compared to the traditional HTTP request, which only allows the client to send requests to the server, socket.io enables real-time, bi-directional, and event-driven communication between the client and the server.

When the system starts, a socket.io connection is built. Under this connection, the client end, namely the browser, can listen to any event emitted from the server end. In our case, the server monitors the CSV data file and emit the new data to the client whenever the CSV file changes. The browser then updates the visualization with the new data coming in real-time.

3.3.3 RadViz panel

The RadViz panel provides an overview of multiple runs with a set of selected variables examined simultaneously. Unlike the most common use of RadViz for revealing clusters in multivariate data, *t*-RadViz mainly uses it to show paths of changes for multivariate time series. Figure 3.3 shows the RadViz projections of three runs of testing. The runs are represented by distinct colors. Four input variables are placed along the circle as anchor points. Each multivariate data point in a run is projected to a 2D point in the RadViz space. The closer a 2D projection to an anchor, the bigger the normalized value of the corresponding variable compared to other variables at that data point. For each run, the data points are chronological, so are their 2D projections. There is an option to connect the 2D projections of the same run using a polyline following their chronological order to reveal the path of changes over time. From Figure 3.3, we can see that the operators kept changing the input variables in many different directions. We can also see that the three runs displayed match in the input space pretty well since their polylines largely overlap. We can also identify outliers from the figure, such as the blue points which are far away from their immediate neighbors. Also, from the figure, we can quickly find the offsets between the orange run and the green run (the blue run matches the green run quite well). In addition, the figure also reveals the input space covered and uncovered by the input paths.

Besides examining and comparing paths of input variables, the users can also use the RadViz panel to check how outputs changes when input changes. Figure 3.4 shows when analyzing the live data, two RadViz circles are displayed side by side to show the projections of input and output respectively. The left RadViz displays four input variables and the right RadViz displays ten output variables. Through the position of the trajectories in the output RadViz (right), we can see the current run, which is displayed in orange, was offset to the bottom left, which indicates the tuning

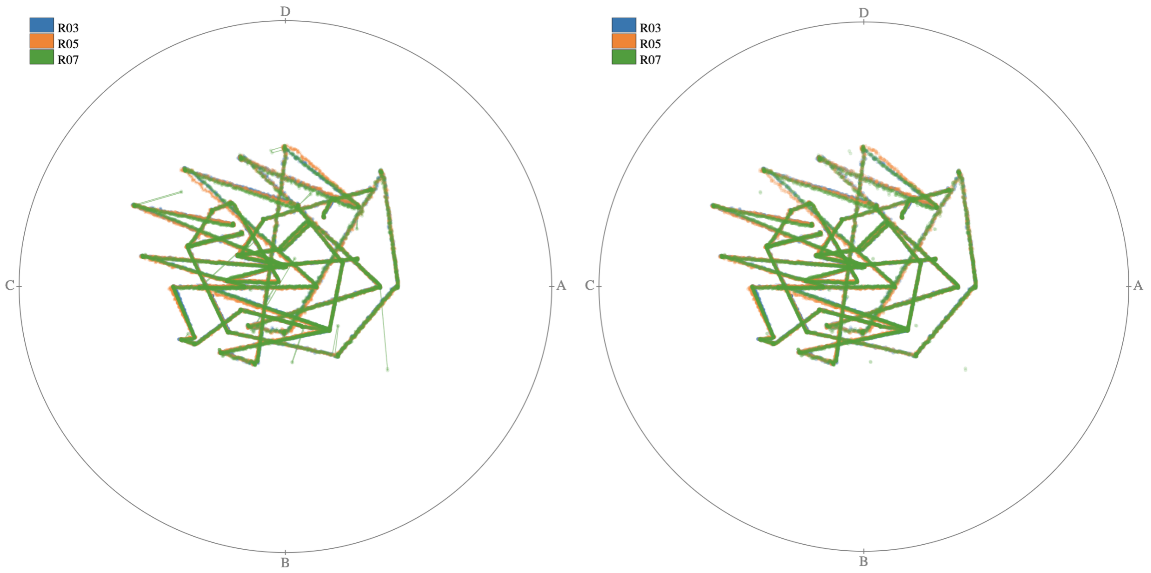


Figure 3.3: RadViz with four input variables. The right figure only displays points. The left figure displays points and polylines connecting the points, which makes it easier to detect outliers

for the current run has a greater impact on the variables set to the bottom left in the RadViz view.

When a test is running, the RadViz panel shows the complete path of any reference runs selected by users as the background and updates the current run as the foreground with the most recent burst highlighted with a bigger point size (see Figure 3.5).

3.3.4 Line Charts

Line charts are among the most intuitive visualizations to represent a univariate time series. In a line chart, the X-axis is a time axis and the Y-axis represents a variable. *t*-RadViz provides this basic visualization for monitoring individual input or output variables of the streaming data and analyzing the existing data.

A line chart panel of *t*-RadViz contains five line charts, according to the preference of the target users. The top line chart displays all four input variables, each represented by a line, for a current run and a reference run, respectively. The current run's lines are displayed in distinct colors and the reference run's lines are all in gray. This

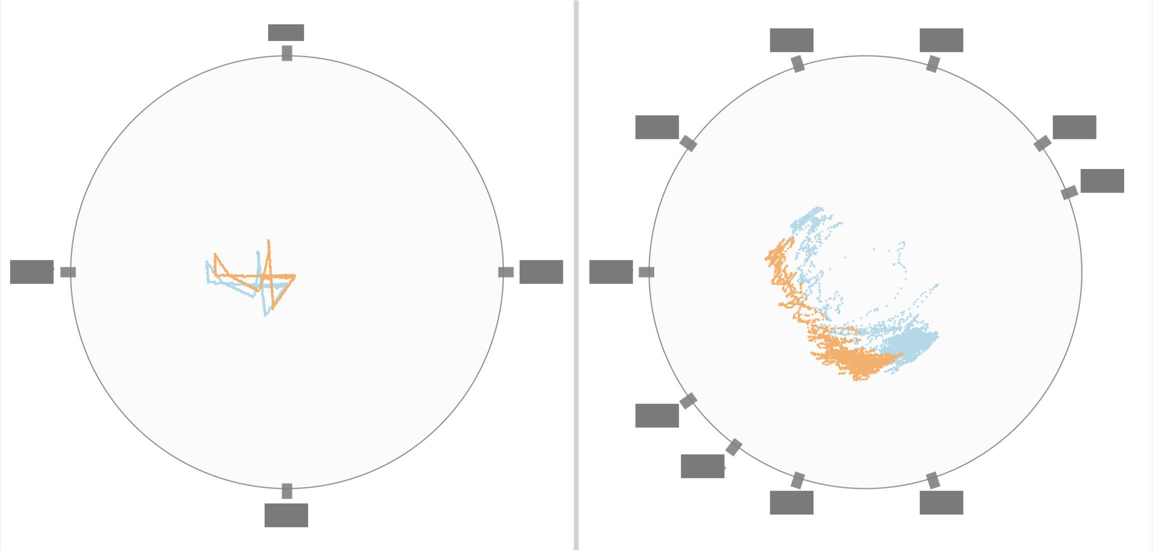


Figure 3.4: Left: RadViz with four input variables. Right: RadViz with ten output variables.

line chart is useful for engineers to compare the input change between the current run and the reference run and to analyze the relative changes of the input variables. Below the top line chart are four line charts, each displays an output variable selected by users. The output of the current run is represented by orange lines and the output of the reference run is represented by blue lines in these line charts.

3.3.5 Bar Charts

It is critical to alert users about significant performance differences between a current run and a reference run in this application. With the line charts, users need to manually compare the performance of the current run and the reference run by comparing the vertical positions of their lines and decide whether the differences are significant. It is not straightforward for decision-making. Moreover, "significant differences" are often unnoticeable in the line charts since the lines are scaled by the whole value range of an output variable, which could be much larger than the scale of "significant differences".

To address this problem, I coupled a bar chart with each of the four line charts to display the value differences between the current run and the reference run on

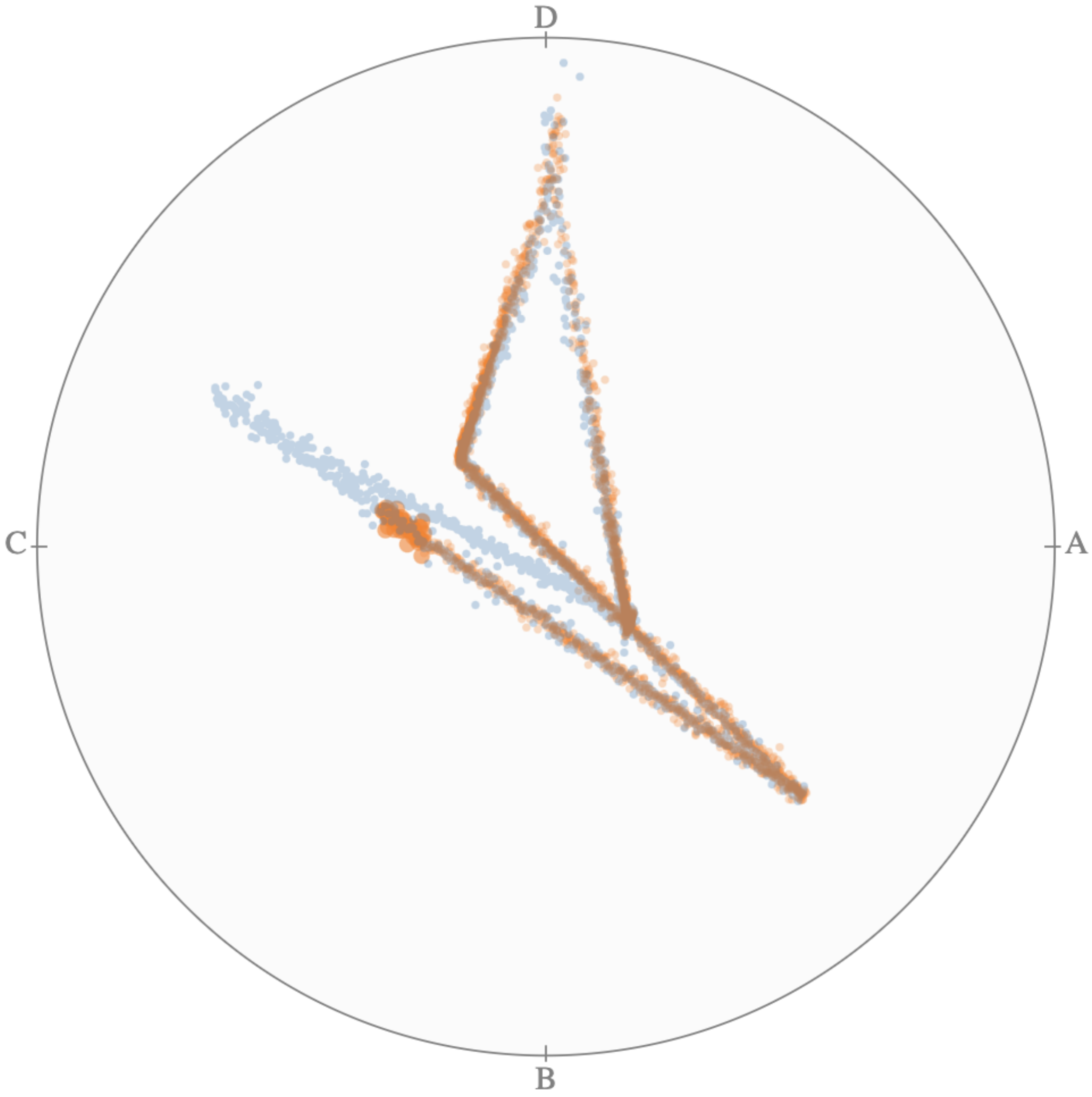


Figure 3.5: RadViz is showing the current test run in live.

an output variable explicitly. A bar chart is displayed below its corresponding line chart. In particular, the time axis is split into time slices. The average value difference between the current run and the reference run on the output variable in each time slice is visually represented by a bar. It is called a **delta bar** since it represents a different value. It adopts *the horizon graph* [48] to save the vertical space and highlight differences. All the bars are displayed on the top of a horizontal line based on the absolute value of the difference. For each output variable, domain experts define whether a positive difference is good or a negative difference is good on that variable. The delta bars are filled in green for good performance or red for bad performance. Meanwhile, the experts define a threshold beyond which a difference value is considered significant. If a bar is taller than this value, its top beyond the threshold line is cut and moved to the bottom of the bar and displayed in a darker green or a darker red. This approach explicitly displays difference values and highlights significant differences in a way that is semantically meaningful and intuitive to users.

Figure 3.6 shows an example of the coupled line charts and bar charts. The top line chart shows four input variables, with variable names hidden for confidentiality. The four line charts below it show four pre-selected output variables, with variable names hidden for confidentiality. In this example, on all four output variables, a run with higher values has a better performance. It can be seen from the figure that the current run has better performance on the first output variable and worse performance on the second and third output variables than the reference run. The delta bars with dark colors highlight the time points when the differences are significant.

In this interface, users can select variables displayed in the line charts via a drop-down menu trigger by clicking the variable names. The value and delta range of each variable are pre-defined by engineers. Users can also interactively change them via a pop out dialog triggered by clicking the Y-axis ticks.

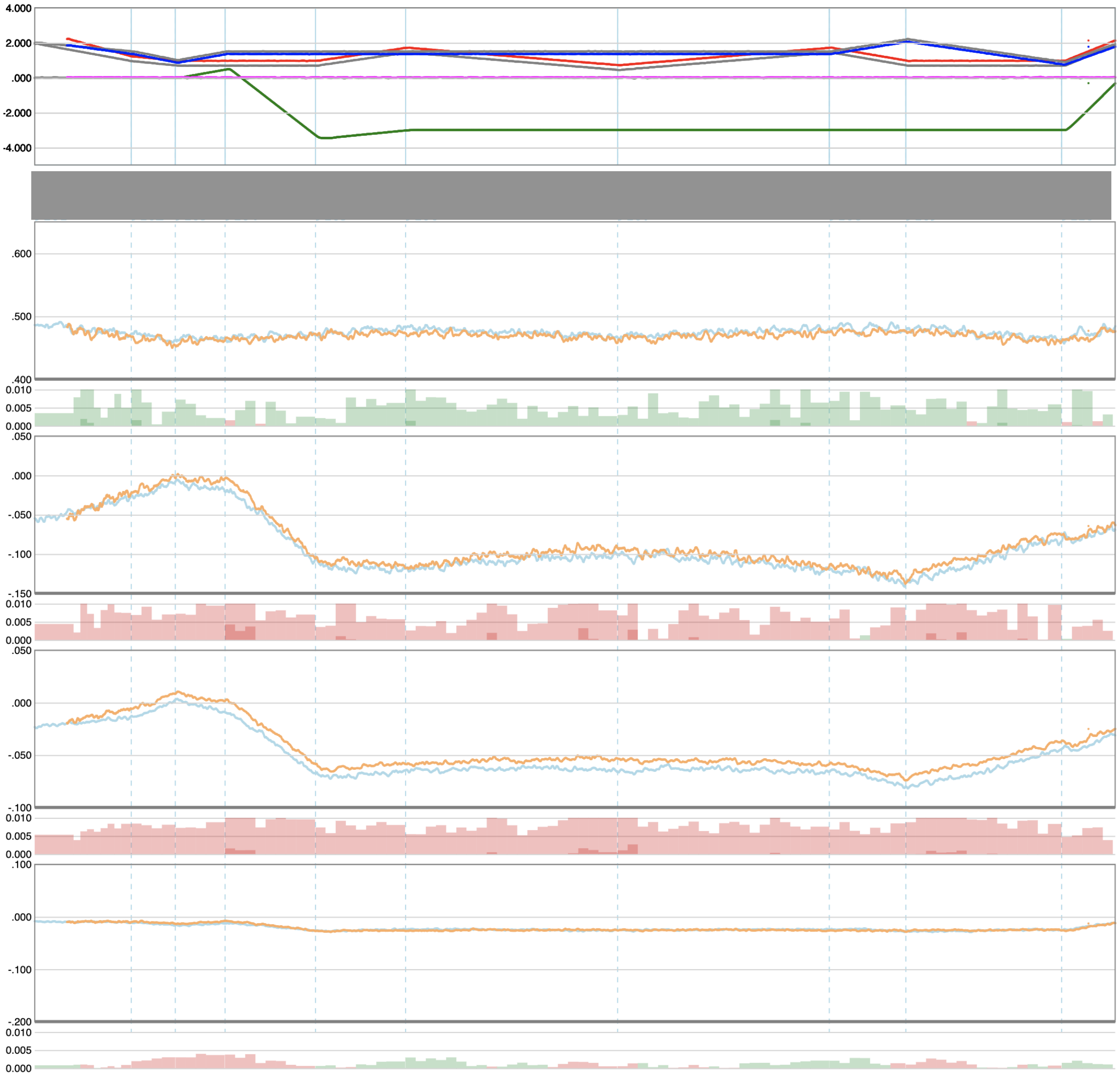


Figure 3.6: The top line chart contains all four input variables. Each of the bottom four line charts and its delta bars are related to a pre-selected output variable. In the line charts, the current/reference run is displayed in orange/blue. In the bar charts, the delta bars in red/green indicate the current run has a worse/better performance than the reference run. For confidentiality, the names of input are covered and the names of variables of the bottom four line charts are removed.

3.3.6 Scatterplots

Scatterplots is a basic way to explore the relationship between two independent variables. *t*-RadViz provides several scatterplots to represent the relationships between pairs of variables.

Users can select variables for both X-axis and Y-axis for all the scatterplots and interactively change their value ranges. However, the scatterplots lack a time dimension. They become cluttered soon when more and more data come in. To reduce clutter, an interaction is provided to allow users to check data in a selected time range. In particular, users can select a time range in the line charts and only data within that time range will be displayed on the scatterplots.

3.3.7 Time Series Alignment

When data points of multiple runs are projected to the RadViz plane, users can easily compare the runs based on the positions of the 2D projections. However, if we display two or more runs in a line chart to compare them, we need to align the runs for effective comparison. The reason is that not only the runs have a different start time, but also the variables adjustment speed varies from run to run. For example, Figure 3.7 shows two line charts (one for an input variable and one for an output variable) where a current run (the orange line) is compared with a reference run (the blue line). A clock time scale is used in the line charts and the two runs are aligned at their start points. It can be seen that the current run has a different input adjustment speed from the reference run, and thus the current run has a different input value from the reference run at the same X position. It makes comparisons of the line charts difficult and the delta bars lose their effectiveness.

To compare the runs effectively, we need to plot the current burst of the current run to the corresponding X positions at the time axis of the reference run on the fly. Since the input change path involves multiple variables, all input variables need to be

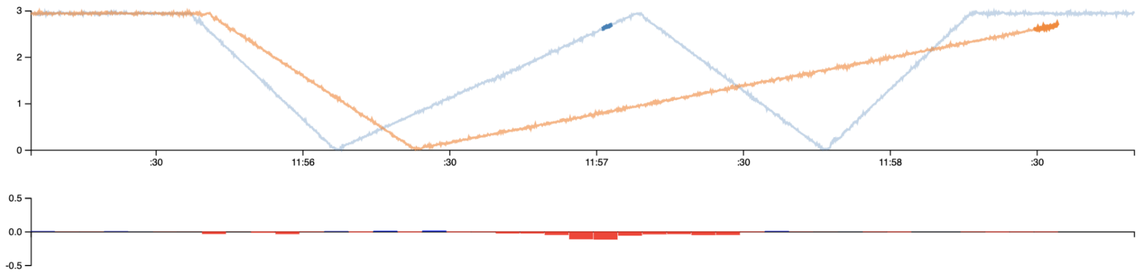


Figure 3.7: The line chart for the current test running in live.

considered in the alignment. Therefore, I developed the following algorithm to align two multivariate time series based on their similarity in the input space as well as the temporal orders of the data points (all runs follow the same input adjustment plan, so the adjustments are made in the same order):

With this algorithm, we used the RadViz projections to find a time series segment from a reference run that matches the current burst of the current run. The current burst and its matching segment are displayed in the same time range in the line charts. In the scatterplots, they are both displayed in dots with a bigger sizes for easy comparison (see Figure 3.8).

When users do not specify a time range for the display in the scatterplot, the system will automatically display the most recent bursts and their matching segments in the reference run identified using the algorithm. In this way, the users can examine the performance of the most recent data and compare it with the reference run in the scatterplots. Figure shows the comparison of scatterplots for pairs of variables.

3.4 Expert Feedback

t-RadViz has been used in in-situ continuous motion tests of the manufacturer after a few round of field tests with positive feedback. To learn whether *t*-RadViz is useful in real work and collect suggestions for improving the system, a google form questionnaire was recently distributed to two direct users of *t*-RadViz, including an end-users and a project manager from the manufacturer. They provided feedback based on their experience with *t*-RadViz. Below are the questions and answers of the

Algorithm 1 Align Segment

```
Declare a $Segment$ as a two element array with its start and end points
Define the minimum length of the segment $MinSeg$
Define related distance $Radius$
Select point $P$ from the reference run
if Length of $Segment$ is 2 then
    Empty $Segment$
end if
if Length of $Segment$ is 1 then
    if Distance between $P$ and $Segment[0]$ less then $MinSeg$ then
        Return
    end if
end if
$Q = GetPoint(P)$
if $Q$ is $Null$ then
    Return
end if
Push $(P, Q)$ to $Segment$
procedure $GetPoint(P)$
    $MinDist$=the minimum distance to $P$ from all points of live run
    Define $Q$ is the point of live run with distance to $P$ is $MinDist$
    if $MinDist \leq Radius$ then
        Return $Q$
    else
        Return $Null$
    end if
end procedure
```

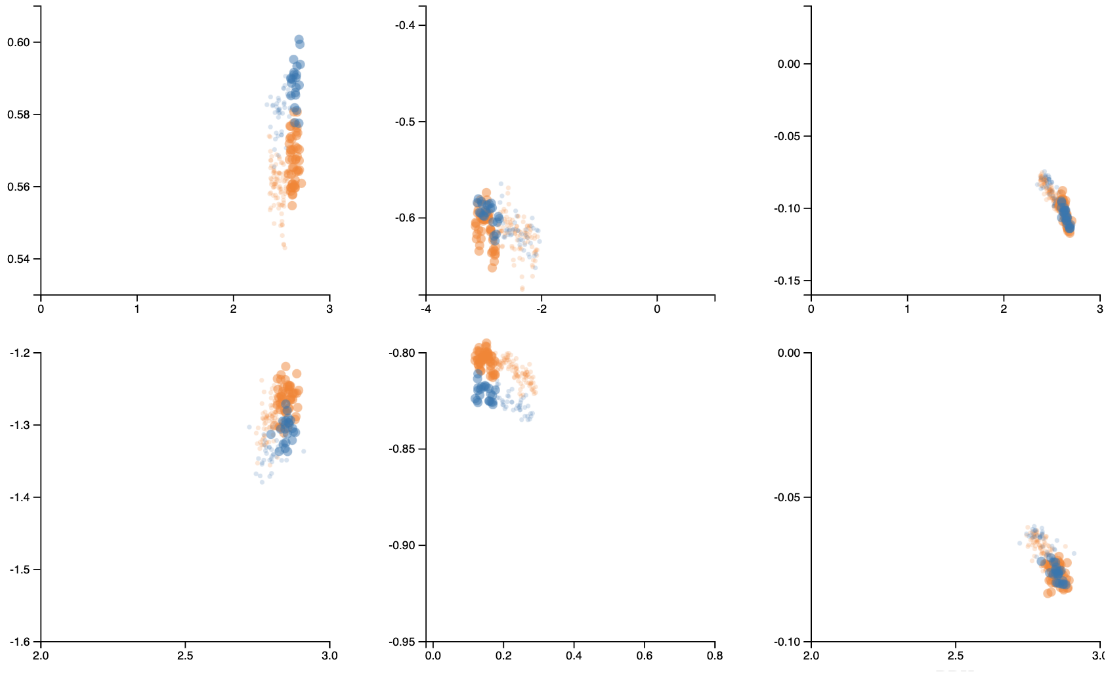


Figure 3.8: The scatterplot for the current test running in live.

questionnaire (the original words of the users are quoted):

Do you think the visualizations and functions are useful in your work? Both users answered yes and left comments on the usefulness of t-RadViz. "The application provides a very useful tool to analyze and compare large data sets, in near real-time." "We are able to quickly evaluate data as it comes to us." "This allows us to more quickly find potential issues and to more quickly make decisions regarding the direction of the testing."

What are the potential benefits/advantages (if any) of monitoring data using this software compared with the other visualization or other software you used? The users compared t-RadViz with the analysis tool they used before. Here are their comments: "The speed of the data processing and analysis provides a more efficient use of wind tunnel time." "The benefits of having it very user configurable make it useful for many of the test configurations we typically encounter." "This software does not require any additional hardware as it has been written specifically for this purpose." "Other software would have required considerably more effort from a hardware perspective. Web-based software potentially allows more lightweight analysis."

Would you like to use the software in your work? Both users provide positive comments to this questions: "The software is getting used more frequently as the development process moves toward increased continuous motion data collection." "Other users are making use of the near real-time analysis to help make decisions faster and more efficiently." "Multiple analysis methods give more complete information for better decision making." "We currently use this software at almost all tests."

What are the limitations? Please provide your top 2 suggestions to improve the software. The users provided a few suggestions to improve the current design. One concern is about the performance. "Large data sets seem to increase the frequency of app failures." "This is only problematic due to needing time to reset the conditions during testing." "The software should include a state file which can reconstruct the views" The other concern is about the functionality. "It would also be quite useful to have more customization capability for the frame layout" The feedback provides useful insights for future improvement of t-RadViz.

3.5 Conclusion

In this chapter, I presented t-RadViz, an interactive visual interface developed for a manufacturing company to help testbench operators and engineers monitor and analyze real-time streaming testbench data. It uses socket.io to build the communication between the server end and the client end to monitor the real-time data stream. At the client end, the user interface consists of a runs management, RadViz view, line charts, scatterplots, and a table view enhanced with color coding. t-RadViz is among the first effort to use RadViz on visualizing multivariate time series data. It also makes use of the RadViz projection to align multivariate time series segments of multiple runs.

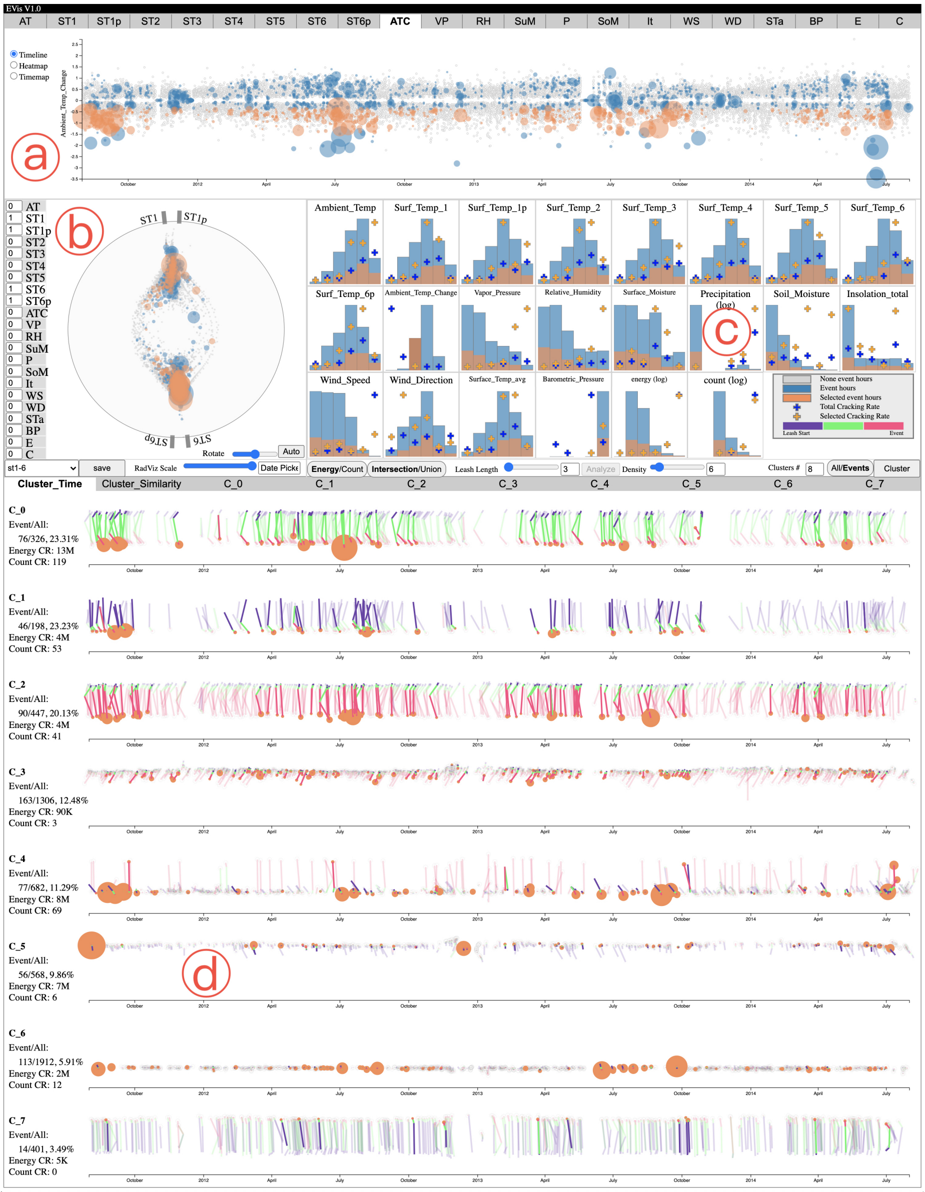
t-RadViz has been used in a real working environment to analyze real-time data of vehicle tests on the spot. According to the feedback of its users, it is useful and greatly improves the work efficiency of testbench operators and engineers.

CHAPTER 4: EVis: Visually Analyzing Environmentally Driven Events

4.1 Introduction

Earth's environmental systems—its atmosphere, biosphere, hydrosphere, and lithosphere—are all linked together via complex pathways that make a prediction of changes in any one system extremely challenging. In particular, discrete '**events**' such as rock fracture, flooding, and landslides, for example, are driven or limited by environmental factors like air temperature, rainfall, or river discharge. To best prepare for climate change's impacts, Earth scientists are monitoring **Environmental conditions** (**E-conditions**) at increasingly high temporal resolutions to understand **triggers**, namely the driving environmental factors, of natural event phenomena (hereafter events) [49, 50]. However, the large number of variables contributing to events, combined with the convolution of their covariance, has hindered a large majority of scientists from completing more than the most basic analyses [51]. Most researchers employ line plots, 'wiggle matching', Fourier, or bivariate analyses, and rely on hand filtering to test correlations within data subsets related to one or more dimensions. As a result, there is little room for the discovery of relationships not hypothesized by the viewers. Thus, there is a strong need for time series analysis tools that readily facilitate - without coding - more functionalities than plotting multicolored lines or heatmaps versus time.

Moreover, current approaches employed by Earth scientists lack the flexibility to readily data-mine for **temporally lagging relationships** like those related to antecedent conditions. The rates and magnitude of many environmental phenomena are predicated not only on simultaneous conditions but also on antecedent conditions for periods of minutes to days or weeks prior. Currently, the state of the science regarding



C_0
Event/All:
76/326, 23.31%
Energy CR: 13M
Count CR: 119

C_1
Event/All:
46/198, 23.23%
Energy CR: 4M
Count CR: 53

C_2
Event/All:
90/447, 20.13%
Energy CR: 4M
Count CR: 41

C_3
Event/All:
163/1306, 12.48%
Energy CR: 90K
Count CR: 3

C_4
Event/All:
77/682, 11.29%
Energy CR: 8M
Count CR: 69

C_5
Event/All:
56/568, 9.86%
Energy CR: 7M
Count CR: 6

C_6
Event/All:
113/1912, 5.91%
Energy CR: 2M
Count CR: 12

C_7
Event/All:
14/401, 3.49%
Energy CR: 5K
Count CR: 0

Figure 4.1: The New Mexico Rock Dataset [9] in EVis. (a) The value-time scatterplot of Ambient Temperature Change (ATC). A group of discrete-time-points with negative ATCs are selected as POIs. (b) RadViz with two top surface temperature variables placed on the top and two bottom surface temperature variables placed at the bottom. (c) Histograms showing event hours and event rates of the whole dataset (in blue) and the POIs (in orange). (d) The cluster timeline view where three-hour leashes of the POIs are grouped into eight clusters and displayed over timelines. The clusters are sorted by event rate in descending order. Leashes of non-event hours are dimmed.

lag-correlations requires scientists to know the period of the antecedence or to iteratively test multiple periods - a time-consuming and dissuading process. In addition, there is a need to readily explore the relationships among multiple environmental variables in the time(s) leading up to events of interest.

In this chapter, I present EVis (Fig. 4.1), a fully working visual analytics prototype, developed to help scientists who have minimal coding or data mining expertise visually analyze time series of complex environmental data. EVis was developed via a two-year collaboration between Earth scientists (scientists for short) and visualization researchers (vis-researchers for short). The New Mexico Rock dataset [52, 9], a previously well-studied dataset devoted to identifying key environmental variables driving natural rock cracking, was used to develop and evaluate the prototype. The scientists used sensors attached to natural rocks to capture rock cracking events and E-conditions in real-time. This dataset records 280,979 rock cracking events and 22 E-conditions simultaneously collected from a rock at an interval of one minute over three years. The overall goal of the research is to identify driving factors of the rock cracking events, with temporally lagging relationships related to antecedent conditions considered. Importantly, this dataset represents a typical exemplar of a broad array of scientific problems that may be addressed using continuous environmental monitoring in the context of any particular events of interest.

EVis was created to allow scientists to evaluate known and discover unknown event triggers - not only in terms of simultaneous E-conditions when events happened but also environmental changes and other events occurring before the events of interest occurred. Toward these goals, EVis provides a set of basic and advanced graphical charts as well as a novel visual analytics approach, all coordinated in the same visual interface for a smooth visual analytics workflow. The basic charts include scatterplots, histograms, and heatmaps. The advanced chart is a RadViz view [53, 8] with new interaction techniques. They allow users to interactively explore relationships between events and one, two, three, or more simultaneous E-conditions. EVis also provides a new visual analytics approach named **RadViz-Leash**. It integrates

RadViz projection [53, 8], trajectory clustering, and visualizations to allow users to interactively explore temporally lagging relationships between events and multiple **Preceding Environmental conditions** (**PE-conditions**), namely the environmental changes before the events occurred. The usefulness of EVis is illustrated by two case studies conducted with a senior Earth scientist and feedback collected from 11 domain experts.

The main contributions of this chapter include (1) EVis, a visual analytics prototype specifically designed for Earth scientists to explore environmentally driven events and their relationships to multiple E-conditions, (2) RadViz-Leash, a new visual analytics approach for interactively analyzing temporally lagging relationships between events and their multidimensional preceding conditions, and (3) case studies and expert feedback to evaluate the usefulness of EVis and RadViz-Leash.

4.2 Background - Rock Mechanical Weathering

One co-author of this work is a senior Earth scientist. She and her team have been studying the type of rock mechanical weathering datasets presented herein for about 15 years. Mechanical weathering refers to the physical breakdown - cracking - of rock that naturally occurs when rocks are exposed to Earth's atmosphere and to water. It is a key component of all Earth surface processes and one for which data and understanding have been strongly lacking [54]. The scientists employed Acoustic Emission (AE) technology to study mechanical weathering by 'listening' to rock cracking events in real-time [52, 9]. They collected the dataset used in this chapter from a granite boulder placed in Sevilleta National Wildlife Refuge in New Mexico [55]. An instrumentation system was affixed to the boulder, which consisted of multiple thermocouples, strain gauge rosettes, surface moisture sensors, and AE sensors. All sensors were installed on the boulder and calibrated in control. Measures of surface strain and temperature, surface moisture, and other E-conditions from an adjacent weather station were collected once a minute for about three years. During the same period of time, cracking events, as recorded by AE, were monitored continuously and recorded whenever they occurred. Each AE signal exceeding a defined threshold was recorded as an individual 'event' with associated 'energy' related to the strength of the captured signal. Therefore, in this case, cracking is the environmental event of interest, and that cracking also has an additional attribute of energy. This scenario is typical of other Earth science applications. For example, the 'event' might be a flood and the additional attribute volume of water.

Twenty-two E-conditions were recorded for 1,578,240 minutes. The E-conditions included not only sensor-collected variables but also derived variables such as Vapor Pressure, which was calculated from Ambient Temperature and Relative Humidity. Meanwhile, 280,979 AE events were detected and recorded during this period. The dataset was fully vetted for quality in past work before this study.

In past work, the scientists employed stacked and annotated line charts (hand-selected and assembled in drawing software), histograms of hand-filtered data, and bivariate plots to analyze this dataset and similar datasets collected from other rocks [52]. Selecting days and hours of key data from the datasets required thousands of people-hours of data mining through hundreds of individual purpose-built graphs to find trends in a handful of the 22 dimensions available. They improved this approach by developing heatmaps of cracking rates under different E-conditions, but could only co-plot up to two dimensions simultaneously [9]. Based on the analyses presented in past publications, the scientists hypothesized not only that multiple dimensions are interconnected in contributing to observed cracking, but also that the temporal relationship between these dimensions and the onset of cracking varies. For example, they make the qualitative observation that cracking occurs in concert with rapid cooling immediately before cracking—brought on by both wind and rain—following week-long periods of similar hot days [52, 9]. To date, however, there has been no ability to readily test or explore these hypotheses. The scientists have identified a sequence of weather phenomena that appear to lead to events, but had not quantified that sequence in any way, nor identified repeated sequences of phenomena that lead to other events in the database.

4.3 Development Process and Requirement Analysis

4.3.1 Development Process

The vis-researchers and scientists in my team have been working together on this project for two years. The first year was an exploration stage with bi-weekly meetings. The collaboration started with helping the vis-researchers understand the domain problem, the data, and the current practices in Earth sciences. Then, several Tableau [56] dashboards and ad hoc D3 [57] prototypes with different visualization designs were created using real datasets from the scientists. Some designs were suggested by the scientists based on their previous practices and leveraged by the vis-researchers. Others were contributed by the vis-researchers. The dashboards and prototypes were demonstrated to the scientists in face-to-face meetings for feedback.

The exploration stage was important for the collaboration - the scientists did not have a thorough understanding of what the vis-researchers could do for them at the beginning. New tasks and requirements often popped up after they were inspired by the visualizations brought to them. Also, the exploration allowed the vis-researchers to explore the design space and find the most desired visualizations by the scientists. For example, the RadVis-Leash was found to provide a 'new' view of the data, whereas an attempt to plot the daily time series of selected data was found to be redundant and unnecessary.

In the second year, EVis was developed using Node.js [58] and D3 [57]. It integrated the designs most positively perceived by the scientists. The team met weekly to report the development progress and collect feedback for recently developed functions. Throughout March 2021, the team met intensively (multiple one-hour Zoom meetings per week) to test the system using the pair analytics method [59]. In these meetings, a senior scientist explored her data using EVis. She drove the data exploration process and a vis-researcher conducted interactions for her. Usability issues noticed in the process were recorded and addressed before the next meeting. New functions were also added to EVis to address needs not captured in the original design but deemed desirable in the initial stages of evaluation. Chapter 5 recorded three case studies conducted in those meetings and feedback from a group of domain experts.

4.3.2 Requirement Analysis

Tasks and requirements for EVis became more and more clear when the collaboration went on. The high-level tasks of EVis are to evaluate known and discover unknown triggers for events. To achieve these goals, the scientists need to: (1) rapidly explore relationships among time (both calendar and diurnal time), one or more E-conditions, and events (e.g., the **foundational analyses**), such as learning conditions coupled with high cracking rates and examining cracking rates at different value ranges; (2) effectively explore temporally lagging relationships between E-conditions and events, such as identifying typical PE-conditions associated with high cracking rates and analyzing events whose PE-conditions were different from prior knowledge of the scientists; (3) interactively conduct intuition-based exploration of hypotheses (e.g., the **exploratory analyses**), such as highlighting data items with characteristics of interest and then examining their distributions in other dimensions and their relationships to events.

High-level requirements include the ability to: (1) employ vetted visualization practices of Earth scientists and leverage them; (2) effectively and efficiently examine—without coding—a large number of relationships, no matter if they are hypothesized ahead of visualizing the data or not; (3) examine intuitive visualizations through easy-to-use interactions since Earth scientists are usually not visualization experts. More minor specific requirements included the ability to: (1) visualize E-conditions with events, together with E-conditions without events, to eliminate biases, since a value range may contain a large/small number of events simply because the E-conditions are heavily/rarely sampled in this value range; (2) analyze events and environmental factors in the context of calendar time and diurnal time for seasonal and diurnal patterns; (3) provide quantitative information, such as event statistics, to guide visual exploration.

4.4 EVis

4.4.1 Data Preprocessing and Concepts

To generate visualizations, the continuously monitored E-conditions were aggregated using a domain-expert-selected time granularity (one hour in this chapter). This resulted in a time series dataset where the dimensions are environmental variables, and each data item recorded the aggregated values of these variables in an hour (thus a data item is called a **discrete-time-point**). Then, metrics for the natural phenomenon under investigation (e.g. sum of energy or count of rock cracking events in this chapter) were aggregated using the same granularity and added to the dataset as a **Dimension of Interest (DOI)**. Discrete-time-points with no event detected are called **non-event hours** and those with non-zero event counts are called **event hours**. The New Mexico Rock dataset contains 26,304 discrete-time-points (the null-value missing data was linearly interpolated), including 2,240 event hours. Users could set either event energy or event count as the DOI in an exploration. The discrete-time-points users selected in exploration are called **Points Of Interest** (**POIs**). The total number of event counts (or the energy sum) divided by the total number of hours of a group of discrete-time-points is called the **event rate** of the group.

4.4.2 Visualization Overview

EVis provides a set of coordinated visualizations (see Fig. 4.1 for a screenshot of EVis). To conduct foundational analyses that are common across virtually all

Earth science applications, EVis provides scatterplots (Fig. 4.1(a)) to depict how individual environmental variables change over time and how they are related to the DOI, histograms (Fig. 4.1(c)) to show total numbers of event hours and event rates of discrete-time-points in different value ranges on each dimension, and heatmaps (Fig. 4.2(a)) to depict event rates of discrete-time-points in value ranges defined by three environmental variables. Although not yet widely employed in Earth sciences, EVis includes RadViz visualization [53, 8] (Fig. 4.1(b)) because of its strong potential in this application [60], particularly when examining the relationships between the DOI and multiple variables and their temporal trends. The new approach **RadViz-Leash** is implemented in EVis for visually analyzing temporally lagging relationships among the DOI and multiple variables.

To coordinate and simplify visualizations for users, the same bubble metaphor and color coding are implemented in most visualizations. In particular, discrete-time-points are represented by bubbles in scatterplots, RadViz, and RadViz-Leash, whose sizes represent the DOI values of the discrete-time-points. The bigger the values, the bigger the bubbles. Event hours are represented by blue (unselected), orange (selected), or green (selected with a different approach) bubbles, and non-event hours are represented by small, hollow gray bubbles.

EVis supports intuition-based exploratory analyses and a smooth, flexible visual exploration pipeline. Users can select discrete-time-points from scatterplots, histograms, and RadViz. The selected discrete-time-points (POIs) are highlighted in all scatterplots and RadViz so users can examine their distributions in these views. Users can also examine aggregated information of the POIs in the histograms. Moreover, users can explore PE-conditions of the POIs within a user-selected time span at multiple levels of detail and examine discrete-time-points with similar PE-conditions of a focus POI using RadViz-Leash .

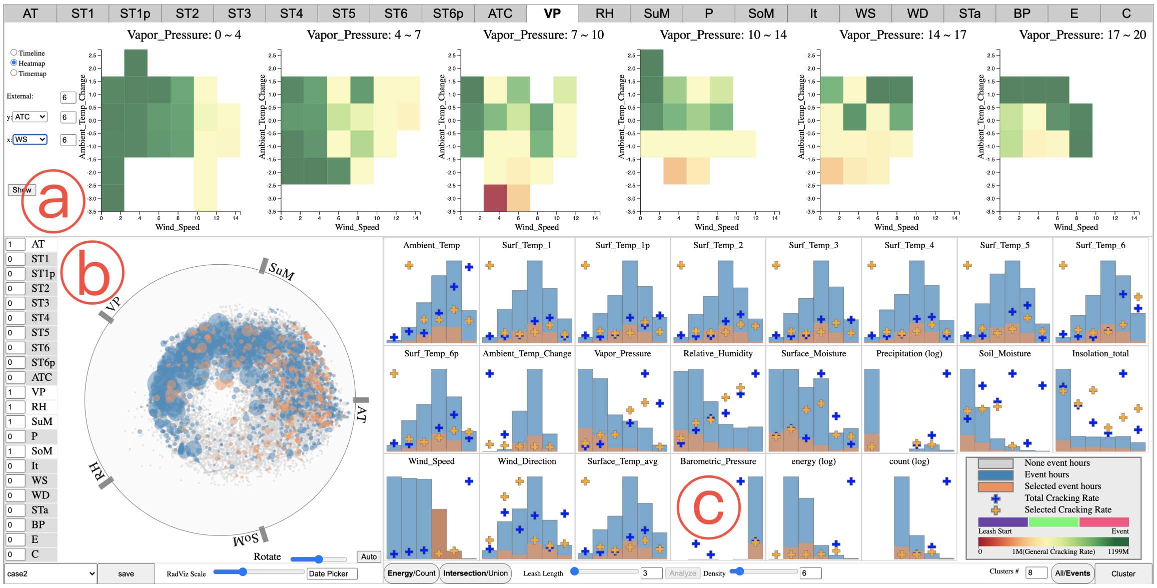


Figure 4.2: Studying POIs with high wind speeds. (a) Heatmaps with Vapor Pressure (outer dimension), ATC (Y-axis), and Wind Speed (X-axis). (b) RadViz with a few well-known triggers as anchor dimensions. Highlighted POIs are well-distributed in this view. (c) Histograms. Bars of high wind speeds are selected.

4.4.3 Basic Graphical Charts

Value-time scatterplot: Discrete-time-points are mapped to a 2D 'value vs. time' scatterplot as bubbles (Fig. 4.1(a)). The X and Y axes of the scatterplot are time and an environment variable, respectively. Users can observe the relationships among the DOI, the variable, and time via the positions, colors, and sizes of the bubbles. Users can click the tags above the scatterplot to switch among different environment variables. I chose this simple approach since it is intuitive but reveals rich insights. Line charts were not used since the many zero values in the dataset would lead to very cluttered line charts.

Date-diurnal time scatterplot: Discrete-time-points are also mapped to a 2D 'date vs diurnal time' scatterplot, inspired by LastHistory [61], to reveal diurnal patterns from midnight to midnight, which is of interest to scientists (Fig. 4.3(a)). The X and Y axes of this view are date and diurnal time, respectively.

Heatmaps: The heatmaps (Fig. 4.2(a)) provide insights into multivariate relationships among the DOI and three user-selected environmental variables. The variables are mapped to X, Y, and an outer dimension of the heatmaps, respectively. The outer dimension is divided into multiple value ranges, each for a heatmap. The X and Y dimensions are also divided into multiple value ranges. In each heatmap, the color of a cell represents the event rate of all discrete-time-points falling in the 3-dimensional region defined by the corresponding X, Y, and outer value range. The depicted color scale for event rate is normalized to the event rate of the whole dataset: green/red colors represent relatively low/high event rates. Users can interactively change the divisions of the value ranges to examine event rates at different granularities. The side-by-side heatmap layout allows users to compare event rates in different value ranges of the outer dimension easily.

Since E-conditions are not evenly distributed (e.g., there are very few hours with ambient temperatures lower than -10 Celsius degree), it is important for scientists to learn the densities of discrete-time-points in the cells. To support this task, EVis overlays a scatterplot over the heatmap, whose X and Y axes are the same as the heatmap, and only discrete-time-points falling in the value range of the outer dimension for that heatmap are displayed as bubbles. Users can turn the scatterplots on or off by clicking a button. To reduce the complexity of the interface, the scatterplots and the heatmaps share the same display space. Users can switch among them by clicking a button.

Histograms: EVis provides a set of histograms, one for each variable, to facilitate selection and reveal relationships between the DOI and the variables with relation to the whole dataset and POIs. A histogram consists of a set of bars and crosses (Fig. 4.1(c) and 4.2(c)). Its X-axis is divided into equal-width value ranges of the variable it represents. Heights/Y positions of blue bars/crosses represent numbers of event hours/event rates of the whole dataset in the corresponding value ranges, normalized within each histogram to make the best use of the space. Numbers of event

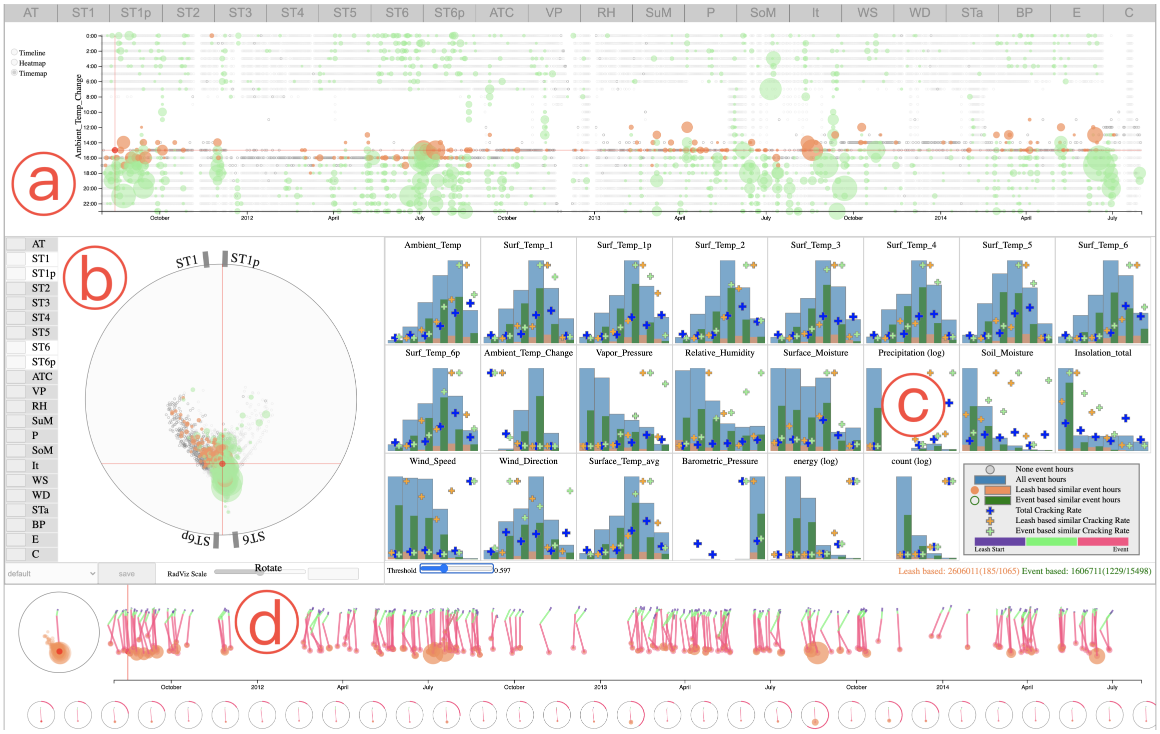


Figure 4.3: Studying a leash of interest using similarity search. Red: the focus POI. Orange: POIs with similar leashes to the focus POI. Green: POIs with dissimilar leashes but similar POI positions to the focus POI. (a) The date-diurnal time scatterplot shows orange POIs are distributed around 15:00 pm. (b) RadViz shows different distributions of orange POIs and green POIs. (c) The histograms show orange POIs and green POIs have different event rates in many value ranges. (d) Leashes of orange POIs are displayed in a timeline view and a detail view similar to Fig. 4.7 (not fully displayed due to the page limit).

hours/event rates of POIs in the value ranges are represented by oranges bars/crosses so that users can compare them with those of the whole dataset. A bar/cross is hidden if the number of event hours/rate is zero to distinguish it from those with small values.

Selection: Users can drag a rectangle to select discrete-time-points in any of the scatterplots. They can also click a bar of a histogram to select discrete-time-points in that value range. Users can merge a new selection with a previous selection via an intersection or union set operation. For example, a user selects discrete-time-points with low ambient temperatures from the Ambient Temperature histogram and then intersects it with a selection from a rectangle in the date-diurnal time scatterplot to highlight discrete-time-points with low ambient temperatures that also occurred around noon. The POIs are highlighted in other visualizations, so the user can learn how the POIs are related to Wind Speed, Precipitation, and other variables. They may explain why the event counts/energy are different within similar ambient temperatures and times of the day.

4.4.4 RadViz

EVis provides a RadViz view (Fig. 4.1(b) and 4.2(b)) to explore relationships among the DOI and multiple user-selected environmental variables. These variables are interactively selected and placed along the circumference of a circle as anchor dimensions. Discrete-time-points are projected to the interior of this circle as bubbles, whose sizes and positions reveal relationships among the DOI and the anchor dimensions. Assume $[\overrightarrow{S}_1, ..., \overrightarrow{S}_n]$ are vectors for anchor positions of n dimensions. After normalizing all dimensions to $[0, 1]$, a discrete-time-point $[d_1, ..., d_n]$'s projection vector $\overrightarrow{u}$ is calculated using the following equation:

$$\overrightarrow{u} = \frac{\sum_{j=1}^{n} \overrightarrow{S}_j d_j}{\sum_{j=1}^{n} d_j} \tag{4.1}$$

I chose RadViz over other dimension reduction methods because (1) RadViz explicitly reveals the relationships between the discrete-time-points and the anchor dimensions and (2) the projections are solely decided by the anchor positions and the domains of the anchor dimensions and, thus, I can use RadViz to generate trajectories of multivariate time series (see Section 4.4.5).

A drawback of RadViz is that the projections are often concentrated in a small region of the 2D space. To overcome this problem, I propose two new interactive distortion methods for RadViz. Note that there are automated anchor dimension relocation algorithms [43, 44] for clutter reduction in RadViz. I did not choose them because they would change the anchor layout created by scientists, which may carry

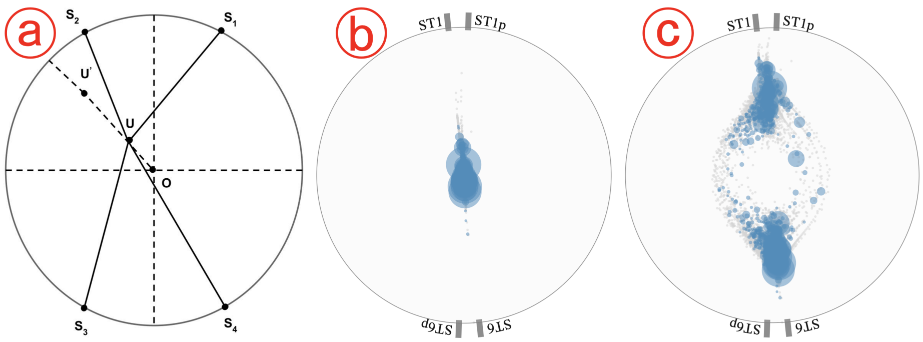


Figure 4.4: Rescale the projections. (a) An illustration of how rescaling works. (b) A cluttered RadViz view. (c) (b) rescaled with $s = 5$.

their domain knowledge and hypotheses. In addition, my distortion methods are fully controllable by users, which makes it easier for users to understand the distortion.

Distortion method 1: Rescale the projections If the anchor dimensions are highly correlated, the projections may be concentrated near the center of the circle, which makes RadViz less effective in revealing subtle patterns. For example, Fig. 4.4(b) shows a RadViz view where anchor dimensions are top and bottom surface temperatures. These temperatures are highly correlated and all the points are concentrated at the center. Such clutter cannot be reduced by reallocating anchor positions using algorithms such as RadViz Deluxe [43]. To address this problem, I introduce a global rescale factor s to move $\overrightarrow{u}$ to $\overrightarrow{u'}$ using the following equation:

$$\overrightarrow{u'} = \frac{\overrightarrow{u}}{||\overrightarrow{u}||} \cdot ||\overrightarrow{u}||^{\frac{1}{s}}, (s \geq 1) \tag{4.2}$$

Since $0 \leq ||\overrightarrow{u}|| \leq 1$, increasing s will monotonically push $\overrightarrow{u'}$ away from the center without changing its original direction. If $||\overrightarrow{u_1}|| < ||\overrightarrow{u_2}||$, then $||\overrightarrow{u_1'}|| < ||\overrightarrow{u_2'}||$ for all $s \geq 1$. The smaller $||\overrightarrow{u}||$, the bigger the distortion. Therefore, this distortion amplifies tiny distances among data points around the center of RadViz while keeping their angles and orders in the radial coordinate unchanged. Users can interactively change s through a scale to enlarge or reduce the distortion effect. Fig. 4.4(a) illustrates how u moves to u' with rescaling. Fig. 4.4(c) shows Fig. 4.4(b) rescaled with $s = 5$.

Distortion method 2: Adjust weights of dimensions It is often observed that data points are concentrated in a region distant or proximal to an anchor dimension in RadViz (see Fig. 4.5(b) for an example). I propose a new approach to pull data points toward or push them away from an anchor dimension to reduce clutter. Inspired by iPCA [62] where users can interactively set weights for dimensions in a PCA, I allow users to interactively assign a non-negative weight w_j to an anchor dimension j. The deformed projection $\overrightarrow{u'}$ is calculated with the following equation:

$$\overrightarrow{u'} = \frac{\sum_{j=1}^{n} \overrightarrow{S}_j w_j d_j}{\sum_{j=1}^{n} w_j d_j} \tag{4.3}$$

Equation 4.1 is a special case of equation 4.3 where all weights are 1. Assume I adjust w_1 and keep the other weights 1. It is trivial to see that if $d_1 = 0$, $\overrightarrow{u'}$ will be the same as $\overrightarrow{u}$, the position calculated with equation 4.1. If $d_1 > 0$ (all dimensions are normalized to [0,1]), by defining $\alpha = \frac{\sum_{j=2}^{n} d_j}{d_1}$, I get the displacement from $\overrightarrow{u}$ to $\overrightarrow{u'}$ with the following equation:

$$\overrightarrow{u'} - \overrightarrow{u} = \frac{w_1 - 1}{w_1 + \alpha}(\overrightarrow{S_1} - \overrightarrow{u}) \tag{4.4}$$

It shows that when $w_1 > 1$ or $w_1 < 1$, all the projections will move toward or away from the anchor position of dimension 1 along straight lines connecting the anchor position and the original projections. It makes the distortion results visually easy to follow. The smaller α, the bigger the movement. Thus, increasing the weight will amplify the influence of a dimension in RadViz. When $w_1 = 0$, dimension 1 has no influence on RadViz according to equation 4.3.

Fig. 4.5(a) illustrates how u moves to u' after the weight of the dimension on the top is increased. Fig. 4.5(c) shows Fig. 4.5(b) after increasing the weight of the top dimension from 1 to 5.

Besides the distortion techniques, EVis allows users to **interactively add/remove**

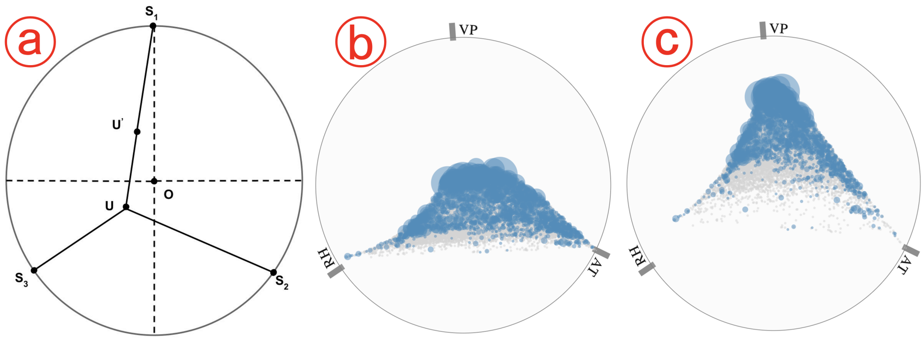


Figure 4.5: Adjust the weight of a dimension. (a) An illustration of how the distortion works when the weight of the dimension on the top is increased. (b) A skewed RadViz view. (c) (b) after the weight of the dimension on the top is increased to $w = 5$.

dimensions to/from RadViz and **manually relocate the dimension anchors** by dragging the anchors around the circle. The discrete-time-point projections will be changed accordingly. After a manual relocation, users can use the **auto-adjustment** function to evenly distribute the anchors around the circle without changing their order to get a balanced view. If users do not like the angles of the anchors, they can **rotate** the whole RadViz view by dragging a scale for preferred orientations.

By interactively selecting dimensions for RadViz and changing their anchor positions, scientists can experiment with different RadViz setups for hypothesis testing. For example, a scientist hypothesized that the larger the differences between the top and bottom surface temperatures of a rock, the more frequently the rock will crack. Thus, she placed the top/bottom surface temperatures to the top/bottom of the RadViz circle, as shown in Fig. 4.1(b). The distribution of large bubbles verified her hypothesis. Scientists can also experiment with different layouts and keep the layouts with interesting patterns, which may reveal novel relationships they have not noticed before.

Users can interactively **select** discrete-time-points from RadViz using a rectangle or a selection box with an arbitrary shape. Since dimension reduction always causes information loss, examining the POIs in other views helps users get a more precise picture of them.

4.4.5 RadViz-Leash

RadViz-Leash aims to allow scientists to visually analyze temporally lagging relationships among the DOI and multiple variables. It projects the time series ahead of a POI, which consists of the anchor dimensions of RadViz within a user-selected time span, as a trajectory on RadViz. Because the time series is temporally 'in front of' the POI, the trajectory is called a '**leash**' of the POI (Fig. 4.6(a) and (b)). The geometric shape and position of the leash visually depict the changing PE-conditions of the POI within the time span. To analyze PE-conditions of a large number of POIs, RadViz-Leash automatically clusters the POIs based on their leash similarity. The clusters are visually presented to users in multiple views, where users can visually examine groupings of the most commonly recurring PE-conditions associated with the POIs, as well as their relationships to the DOI.

Projection and Leash Visualization: Denote a POI as $p_t = (d_{1,t}, d_{2,t}, ...d_{n,t})$, where n is the number of anchor dimensions and t is the time stamp of the POI. The PE-conditions (anchor dimensions only) of p_t in a time span of length k is a multivariate time series $\{p_{t-k}, ...p_{t-1}, p_t\}$. To visualize this time series, RadViz-Leash projects $p_{t-k}, ...p_{t-1}, p_t$ to the RadViz space. The projections are denoted as $p'_{t-k}, ...p'_{t-1}, p'_t$. RadViz-Leash connects $p'_{t-k}, ...p'_{t-1}, p'_t$ in their temporal order using line segments and draws a bubble at p'_t to represent the DOI at the POI. Colors of the line segments indicate their temporal distances to the POI. The resulting visualization is a leash representing the POI and its PE-conditions (see Fig. 4.6(a) and (b)). Fig. 4.6 shows a set of leashes with $k = 24$ hours. The lengths and directions of the line segments in a leash provide rich information about how the PE-conditions of a POI evolve over time. For example, the PE-conditions of the POIs in the third row changed much more significantly than those of the POIs in the first row of Fig. 4.6.

Distance Calculation: RadViz-Leash clusters POIs based on leash distance rather than PE-condition distance in the raw data space. In this way, clustering results are

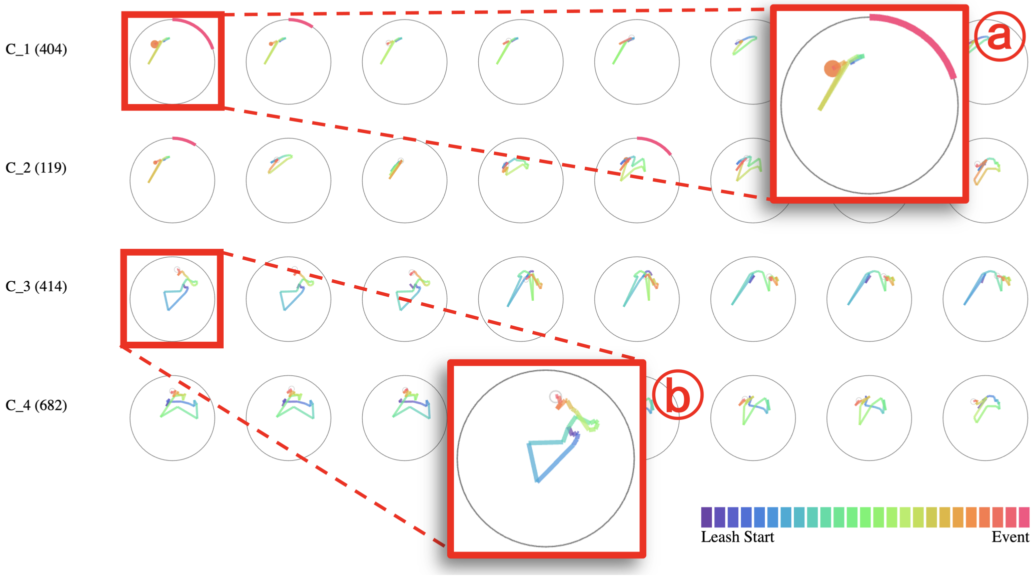


Figure 4.6: A portion of the cluster sequential view (sorted by leash similarity) of POIs at freezing temperatures (leash length = 24 hours). (a) and (b) are zoomed-in views of leashes from different clusters. The POI in (a) is an event hour and the POI in (b) is a non-event hour.

consistent with what users see from the visualization, and domain knowledge integrated into the dimension reduction process is preserved. To calculate leash distance, I borrowed ideas from trajectory distance metrics. Magdy et al. [63] presented a survey on measures for trajectory distance between two moving objects. Measures considering both spatial information and speeds of the moving objects were unnecessarily complex for my application since trajectories in this application have uniformed time steps. Because the absolute positions of the leashes in a RadViz view are important, I excluded spatial assembling distance [64] and angular metric for shape distance [65] from consideration because they are not strict in the distance of absolute positions. Hausdorff and Frechet distance [66] did not have this drawback. It defined the distance of two trajectories A and B using the following equation:

$$H(A,B) = \max_{a \in A} \min_{b \in B} ||a - b|| \tag{4.5}$$

This distance captured how dissimilar two undirected trajectories are based on their geometry. However, it was not strict in the directions of the trajectories. For example, the distance of two leashes with the same geometry but with opposite directions, indicating conditions changing in opposite directions in my application, would be '0' with Equation 4.5. This was misleading in my application. Thus, I decided to modify Hausdorff and Frechet distance to take the temporal order of the leashes into consideration. Denoting leash A as $\{a_k, a_{k-1}, ...a_0\}$ and leash B as $\{b_k, b_{k-1}, ...b_0\}$, I calculate the distance between leash A and leash B $\delta(A, B)$ as follows:

$$\delta(A, B) = \max(\delta(A \to B), \delta(B \to A)) \tag{4.6}$$

$$\delta(A \to B) = \max_{a_i \in A} \min_{b_j \in B[i-\phi, i+\phi]} ||a_i - b_j|| \tag{4.7}$$

$B[i - \phi, i + \phi]$ is a subsequence of B with the range of index from $i - \phi$ to $i + \phi$. $\phi = 0$, if $0 < i < 4$; $\phi = 1$, if $4 < i < 7$; $\phi = 2$, if $i > 7$. $\delta(A, B)$ is decided by the maximum of the point-to-point distances between the two leashes. In general, a point in a leash should be compared with the point with the same temporal order in the other leash. However, it is too strict for long leashes. Thus, equation 4.7 relaxes this requirement by ϕ to compare a point with points appearing a little sooner or later in the other leash and use the minimum distance when the point is temporally far away from the POI. The parameters used here are set by expert users based on domain knowledge.

Clustering: RadViz-Leash uses k-means clustering [38] to group a set of POIs into a user-defined number of clusters based on their leash distances. The centroid of a cluster in each iteration is defined as the POI with the minimum sum of distances to all other POIs in the same cluster. I chose k-means over other clustering algorithms since (1) it minimizes within-cluster variances, which is a desired feature for the POI clusters. Clustering algorithms generating non-spherically shaped clusters are not suitable in my application since they may group leashes representing quite different PE-conditions together; (2) it allows scientists to flexibly set the number of clusters to examine; and (3) it is fast enough for interactive visual exploration.

Overview: Overviews of clustering results are provided to allow users to (1) browse typical PE-conditions associated with POIs and learn their relationships with the DOI, anchor dimensions, and time; (2) identify PE-conditions of interest for further examination. I chose to display PE-conditions in the overviews as leashes since they are compact and information-rich. Two alternative designs were considered: (1) to display all or a cluster of POIs and their leashes in the same RadViz view; (2) to display each POI and its leash in their own RadViz view using small graphics. I chose Design 2 since Design 1 led to a heavily cluttered RadViz view. In addition, the layout of the small graphics can reveal useful information to users.

Multiple layout strategies were tested and two were chosen in the final prototype. One is the **cluster timeline view**. It emphasizes relationships between the clusters and time. It places the small graphics on a set of parallel, horizontal timelines, each for a cluster (see Fig. 4.1(d)). The circles of RadViz are hidden, but the centers of the circles of the same cluster are vertically aligned to allow users to compare the vertical positions of the leashes in RadViz. The X positions of the endpoints of the leashes indicate the timestamps of the POIs. This view allows users to examine and compare leash colors, leash shapes, and time frequencies of different clusters in a compact display. Since it is important to compare event rates of different clusters, interactions are provided to allow users to **sort** the clusters by event rate, **dim** leashes of non-event hours to highlight leashes of event hours, and read event rates which are displayed as text on the left end of the timelines (see Fig. 4.1(d)).

In my initial design, I displayed the circles to provide a better picture of the relative positions of leashes in RadViz. Since the circles caused severe clutter and distracted users from observing the colors and shapes of the leashes, they were removed from the final design. I experimented using the Y positions of the circle centers to reveal the similarity between a POI and the center of its cluster or a user-selected POI. This layout was confusing after the circles were removed. I also tried mapping the timestamps of the POIs to the X positions of the circle centers, but since the circle centers are not displayed, users could not judge when an event happened with this design.

Although the cluster timeline view is space-efficient and presents a nice overview of the clusters, it is difficult to select and compare individual leashes in it. To address this problem, a **zoom** function is provided. It allows users to drag and drop a rectangle on this view. All leashes in the rectangle will be displayed, together with their circles, in a pop-up window without overlaps. Users can create multiple pop-up windows at the same time to compare leashes in different time ranges or different clusters. However, it is trivial to do so. Thus, the second layout, the **cluster sequential view** is provided in the final prototype.

The **cluster sequential view** keeps the row-by-row layout of clusters in the timeline view so that users can switch between the two views without losing the positions of the clusters. In each row, the leashes and their circles are displayed sequentially without overlaps. Users need to horizontally scroll the screen if there are more POIs in a cluster than the screen width can hold. To reduce the need for scrolling and highlight important features of a cluster, sorting, and filtering interactions are provided. Users can **sort** the leashes by **sum of DOIs of all discrete-time-points in the leashes** in descending order since scientists are interested in PE-conditions of significant events, or by **leash similarity**, so that similar leashes are placed adjacent to each other to help users observe variations within a cluster (the sorting algorithm presented in [67] is used to minimize the total distance between adjacent leashes). A cluster may contain many POIs with redundant leashes, namely those leashes only shifting one or two hours in time. Inspired by Schreck et al.[30], a **filtering** interac-

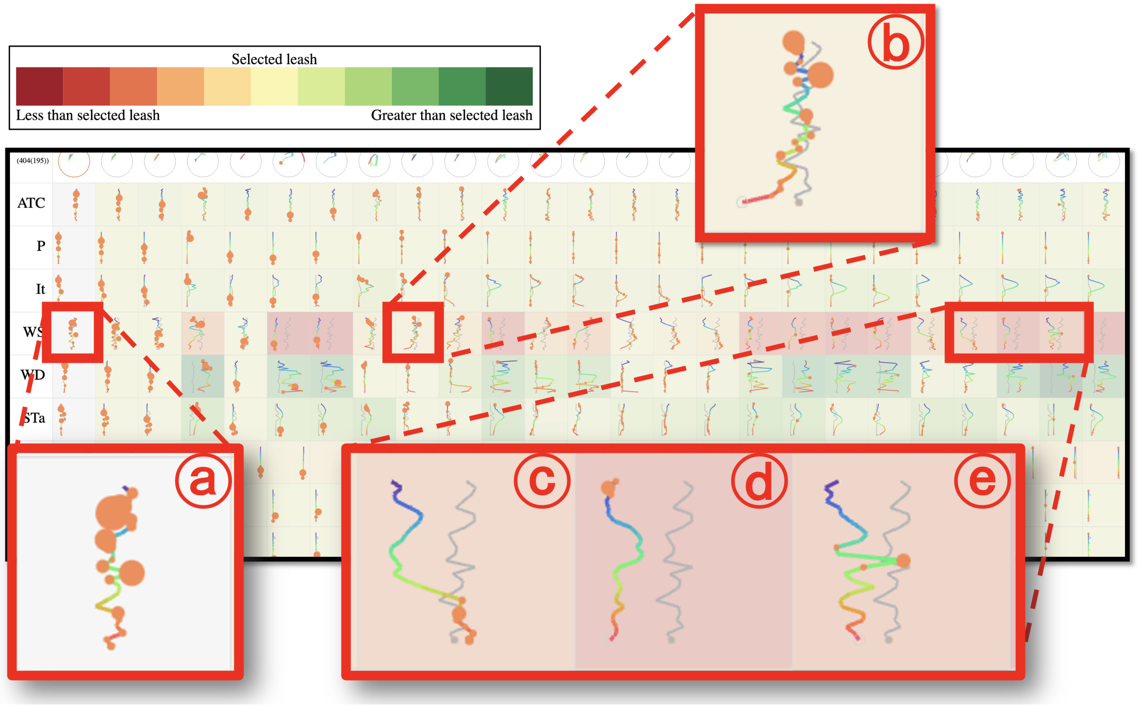


Figure 4.7: A portion of the detail view of a cluster. The first column is selected for comparison. (a)-(e) are zoomed-in views of several line charts of the Wind Speed variable. (a) is the chart of the selected POI. (b)-(e) compare (a) (gray lines) with charts of other POIs. They reveal high wind speeds might be related to events.

tion is provided to filter out POIs with redundant leashes using a user-defined time subinterval. Only the latest POI among consecutive POIs within the time subinterval is displayed in a cluster.

There are other possible ways to layout the leashes to form an overview (see [68] for a survey of glyph placement). For example, all leashes can be mapped to a 2D space where the distances between two leashes reveal the similarity between them. However, they may introduce extra complexities and clutter to the visualization.

Detail View: To understand relationships between events and PE-conditions, scientists need to examine and compare POIs and their PE-conditions in full detail. To support this task, a detail view is provided in the prototype. It is a matrix of small graphics to show leashes and detailed PE-conditions of a group of POIs (see Figure 4.7). In the matrix, each column displays a POI and its PE-conditions. The columns can be sorted and filtered by the methods introduced in the cluster sequential view. The first row displays the leashes of the POIs. Each of the other rows displays line charts [19] of the POIs on an environmental variable in small graphics. The anchor dimensions of RadViz are displayed on the top, followed by non-anchor dimensions. In the line charts, the Y axes are time axes, with the time of the starting points of the leashes at the top and the POIs at the bottom. The Y values of the line are double encoded using colors of the lines, in the same way as the leash segments. The X-axes represent the variables with the smallest values on the left. Note the X-axes in the same row have the same scale for all columns while the Y axes are aligned by the timestamps of the POIs.

Since each line chart only shows one time series on one dimension, I can map the DOI values of all event hours in the leashes to bubbles without cluttering the line charts (see Figure 4.7(a)). Our domain experts commented that this is extremely useful since the bubbles enable them to directly relate the evolving PE-conditions to all events that occurred during the periods of the leashes, which is of interest.

The detail view is useful for finding new event triggers by comparing POIs with similar leashes (selected by clustering or similarity search). Those POIs have similar temporal patterns on the anchor dimensions, which may be selected by scientists since they are known triggers. Why would a POI/leash have different DOI values from other POIs/leashes? The reason might be that they behave differently on non-anchor dimensions, which may be triggers not noticed by the scientists. The detail view provides a **comparison** interaction to facilitate this type of analyses. In particular, users can set a POI as a focus by clicking its leash, and compare it with other POIs on all the dimensions (see Fig. 4.7). In the comparison mode, the line charts in focus are copied to other line charts in the same rows and displayed in gray for a direct comparison (see Fig. 4.7(b)-(e)). The background color of the other line charts indicates the difference. Red means the focus has higher average values and green means the focus has lower average values on that dimension. The larger the difference, the darker the color. The background colors allow users to capture insights from a large matrix at a glance. In Fig. 4.7, the dark red background reveals a focus with large events having higher wind speeds than most POIs with small or no events, which inspires the scientists that wind speed might be a trigger for rock cracking.

An alternative design to the line chart matrix was to overlay all the line charts of the same dimensions in the same view. This design was not used since the view was cluttered and it was difficult for users to examine and compare multiple leashes in detail with the overlapping line charts. Another alternative design was to map the time series to timelines, one for each POI on each dimension, whose colors indicated the values of the PE-conditions. This design was not used because the color was used in leashes to encode the time dimension. Using color in a different way confused users.

The detail view can be triggered by clicking a cluster name in the overviews to examine POIs in the cluster. It can also be triggered by clicking a POI to set it as a focus, and then setting a leash distance threshold d to select POIs with similar leashes to the focus (leash distance $< d$) from the whole dataset. In this **similarity search mode**, a new window is opened (see Fig. 4.3), which contains basic charts (Fig. 4.3(a) and (c)), RadViz (Fig. 4.3(b)), a timeline view (Fig. 4.3(d)) and a detail view for the selected POIs. The basic charts and RadViz allow users to compare selected POIs against POIs with similar conditions on the anchor dimensions when the events happened (POI projection distance $< d$), but dissimilar PE-conditions on the anchor dimensions (leash distance $>= d$). They are displayed in green (event hours) or light gray (non-event hours) to provide context to selected POIs (orange for event hours and dark gray for non-event hours). Other discrete-time-points are hidden in this window to reduce clutter.

4.5 Conclusion

Earth scientists are increasingly employing time series data with multiple dimensions and high temporal resolution to study the impacts of climate and environmental changes on Earth's atmosphere, biosphere, hydrosphere, and lithosphere. However, the large number of variables and varying time scales of antecedent conditions contributing to natural phenomena hinder scientists from completing more than the most basic analyses. In this chapter, I present EVis, a new visual interface to help scientists analyze and explore recurring environmental events (e.g. rock fracture, landslides, heatwaves, floods) and their relationships with high dimensional time series of continuous numeric environmental variables, such as ambient temperature and precipitation. EVis provides coordinated scatterplots, heatmaps, histograms, and RadViz for foundational analyses. They allow users to interactively examine relations between events and one, two, three, or more environmental variables. EVis also provides a novel visual analytics approach to allowing users to discover temporally lagging relationships related to antecedent conditions between events and multiple variables, which is a critical task in Earth sciences. In particular, this latter approach projects multivariate time series onto trajectories in a 2D space using RadViz, and it clusters the trajectories for temporal pattern discovery.

CHAPTER 5: Evaluation of EVis

EVis has been evaluated via case studies and expert feedback. They are reported in this chapter.

5.1 Case Studies

I illustrate the usefulness of EVis using two case studies. They were conducted on the New Mexico Rock dataset [9] by a senior Earth scientist and two vis-researchers from our team via several Zoom meetings (all recorded). In the meetings, a vis-researcher ran EVis on his PC and shared his screen. The scientist orally instructed the vis-researcher what she wanted to do and why, while drawing on the screen with the Zoom annotation function to point to the places where she wanted to select or zoom in. She also verbalized what she found and which hypotheses she had. The other vis-researcher took notes during the meetings and summarized the case studies after the meetings based on the notes and the recordings. EVis settings used in the explorations were saved using a save function provided by EVis, so the results could be duplicated when writing this paper.

5.1.1 Case 1: Temperature and Rock Cracking

The scientist had discovered through past work that temperature is related to cracking [9]. However, there was a lack of detailed analysis—or full understanding—of this complicated relationship. Thus, she decided to use EVis to get more insights into how temperature is related to cracking events. Based on theoretical modeling [69], she hypothesized that rocks would crack when their top and bottom temperatures are diverging in their rate of temperature change. Previously, she had only employed surface air temperature change as a proxy for this effect but had not observed it directly in the dataset. So, using EVis, she manually set up four anchor dimensions to the RadViz view. Surface_Temperature_1 and Surface_Temperature_1p, which record top surface temperatures of the rock, were placed on the top of RadViz. Surface_Temperature_6 and Surface_Temperature_6p, which record the bottom surface temperatures of the rock, were placed on the bottom of RadViz (Fig. 4.1(b)). The discrete-time-points were concentrated around the center. Thus, she distorted the projections to separate the points cluttered at the center. From Fig. 4.1(b), she observed that most hours with large DOIs are either high or low in the circle, which meant that the top surface temperatures were much higher or much lower than the bottom temperatures. This supported her discovery from the previous study, as well as the theoretical models.

According to her previous research, high cracking rates correlate with negative Ambient Temperature Change (ATC). A negative ATC value means a large per-minute drop in temperature within the discrete-time-point. To explore more details, she opened the value-time scatterplot for ATC (Fig. 4.1(a)). She saw large bubbles (hours with high DOI values) with negative ATC values. She dragged a rectangle in the scatterplot to select discrete-time-points in that region (Fig. 4.1(a)). She scanned the histograms (Fig. 4.1(c)) for patterns of the selected hours. According to the orange crosses in the histogram, the POIs have much higher event rates than the whole dataset in most value ranges of variables such as Vapor Pressure and Temperature. It confirmed her prior knowledge that multiple variables align during high DOI periods.

Next, she explored the temporally-lagging relationships between the top and bottom surface temperatures of the POIs. She set a leash length of three hours with a cluster number of eight, based on her prior knowledge of expected lag-importance and the data selected. Then, she clustered the POIs. From the cluster timeline view (Fig. 4.1(d)), she examined the leashes of each cluster and checked their event rates from both the text and the ratio between bright leashes (event hours) and dimmed leashes (non-event hours). She found several clusters with long leashes, indicating the temperatures changed significantly within the three hours before the selected hours. Interestingly, she found that after the temperatures changed significantly from a situation where top temperatures were lower than bottom temperatures to a situation where top temperatures were higher than bottom temperatures (C_7), the cracking rates in both energy and counts were much lower than when they changed in the opposite direction (C_0, C_1, C_2). She also noticed that these long leash clusters are dominated by different colors, which indicate different temporal patterns. For example, most POIs in C_2 were preceded by relatively slow changes in the first two hours (short purple and light green lines) and then rapid changes in the third hour (long red lines). Identifying these clusters of different, specific, lagging conditions leading to events was a novel and impactful finding from the dataset.

To further explore one of the clusters, the scientist set a leash from C_2 as a sample and opened the similarity search view to further analyze it (Fig. 4.3). In this view, she set a similarity threshold. All event/non-event hours whose leash distances to the sample are less than the threshold are displayed as orange/dark gray circles. Other event/non-event hours whose distance to the event hour of the sample leash in the RadViz view are displayed in green/light gray circles. From the date-diurnal time scatterplot (Fig. 4.3(a)), it can be seen that the orange circles, namely the event hours with leashes of the pattern of interest, are distributed close to the daily time of 15:00 pm. The green circles do not have this pattern. This result supported the idea that leashes can better describe the characteristics of cracking events than discrete-time-points. In addition, from the histograms (Fig. 4.3(c)), she found that the orange set has different event rates from the green set in multiple bars. The discrete-time-points, however, have similar conditions themselves in the anchor dimensions - again a novel finding.

5.1.2 Case 2: Exploratory Analyses and Unexpected Patterns about Wind Speed

The scientist conducted another exploratory analysis on the New Mexico Rock dataset [9]. Her initial interest was in events with low to sub-freezing ambient temperatures, a condition well-documented in her field to produce cracking. Thus, she selected those discrete-time-points from the Ambient Temperature histogram and set up RadViz with Vapor Pressure, Relative Humidity, Ambient Temperature, Surface Moisture, and Soil Moisture which were hypothesized to influence cracking rates. Then, she set the leash length to 24 hours and conducted clustering to examine how event rates are related to leash shapes in this RadViz space. The clustering results are shown in Fig. 4.6. Cluster C_1 with small leashes but big events triggered her attention. She opened its detail view (Fig. 4.7). Sorting the leashes by total event energy in 24 hours, she noticed multiple leashes with many large bubbles on them in the first several columns. As expected, their conditions with respect to the anchor dimensions in 24 hours are fairly stable, since their leashes are small in RadViz. This is similar to other members of this cluster with small or no events. Are there any other triggers for those big events? She clicked a leash with many large bubbles (Fig 4.7(a)) to compare its line charts with other columns (Fig 4.7(b)-(e)). A row with light backgrounds for event-heavy time series and dark backgrounds for few/no event time series was immediately visible. It was the Wind Speed dimension. She zoomed in to this row and found that high wind speeds are almost always coupled with high cracking rates under the selected low ambient temperatures.

The scientist knew that negative temperature changes are related to cracking. She suspected those high cracking rates were related to temperature drops caused by strong winds. She opened the heatmaps to test her hypothesis. She set Wind Speed as the X-axis of the heatmaps, Ambient Temperature Change as the Y-axis, and Vapor Pressure, another strong event trigger, as the outer dimension (Fig. 4.2(a)). The heatmaps indicate that with low vapor pressures, wind speed has a clear correlation with event rate. Then, she opened the scatterplots overlaying on the heatmaps. To her surprise, she found POIs with big events under conditions of high wind speed and positive ambient temperature changes from the heatmap of the lowest vapor pressures. This was different from her understanding. Now she learned that temperatures may increase when the winds are strong. Thus, the cracking that occurred in strong winds may also be related to some thermal mechanisms.

Were the selected high-wind event hours caused by antecedent temporal changes of the triggers she knew? She added ambient temperature change into the RadViz view and created another selection of discrete-time-points with high wind speeds. She observed that event hours are well distributed in the RadViz view (Fig. 4.2(b)). In other words, there are event hours with high wind speeds no matter their values of the anchor dimensions are high or low. She ran the clustering algorithm and found all clusters had an extremely high event rate, no matter whether their leashes were long or short. The conclusion that can be drawn from this visualization is that cracking is not strongly related to antecedent temporal changes of the anchor dimensions. Finally, she opened the value-time scatterplot of wind speed. She found multiple non-event hours with high wind speed. This indicated that another hypothesis she had, namely that cracking could be caused by sands blown against the rock by strong winds, might also be wrong since otherwise, all high wind speed hours should have had events. The findings were unexpected and exciting to her. She decided to further investigate these relationships between high wind speeds and rock cracking.

Visualization Take-away for the Case Studies: The process of conducting the case studies was very exciting to the scientist. In a few hours, she was able to confirm multiple insights for which she and her collaborators and graduate students spent multiple years to discover, as well as to discover new insights she had not expected before the case studies. According to her, the visualizations and interactions are intuitive, and the combination of them provides novel and powerful functions never feasible to her in the past. Though the interface has multiple views, she readily and continuously employed needed information from multiple portions of the interface—providing evidence that the interface density is warranted. She commented that both distortion methods are intuitive and useful. She liked that the second distortion method allowed her to strengthen the impact of critical E-conditions, which was exactly what she wanted. It was observed that the coordinated visualizations enabled her to analyze her hypotheses following multiple exploration paths smoothly—she started her explorations from multiple views, depending on the hypotheses she had.

5.2 Expert Feedback

My team demonstrated EVis to Earth scientists from a range of disciplines to learn whether EVis was applicable and useful for their research, as well as to collect feedback for improving EVis (their concern about existing visualizations has been addressed in this paper; new functions they requested will be the future work). Thirteen scientists attended a one-hour Zoom meeting with the co-authors. They were known to us to be actively collecting or working on complex time-series datasets. They represented subdisciplines of climatology, geology, rock mechanics, physical geography, and ecology. They included a range of professions and career stages: late-stage Ph.D. students (2), post-doctoral researchers and early-career academics (3), mid-career academics (3), international governmental employees (1), full professors (2), and experienced industry professionals (2). In the Zoom meeting, the Earth scientist coauthor first introduced her rock mechanical weathering research for five minutes and then gave a live demo of EVis with the New Mexico Rock dataset [9] for 40 minutes. A 15-minute discussion about EVis followed the demonstration. A Google Form questionnaire was distributed to the scientists before the demo started. Eleven out of the thirteen scientists returned their answers to us within four days after the meeting—there had been no requirement to submit the form to attend the demonstration. Below are the questions and answers of the questionnaire (the original words of the experts are quoted):

Is the software applicable in your field? Nine scientists answered yes to this question. They commented that EVis is “extremely relevant and applicable” to their fields of research. “Its unique capability to visualize high-density multi-variable time-series data is exciting and is directly applicable” to their projects dealing with processes such as landscape evolution and subsurface fracturing. It is “useful for testing and exploring hypothesized mechanistic linkages between an observed process in the field and variability in the environmental conditions that drive it”, or even for “revealing linkages that were not hypothesized ahead of the experiment”. One scientist wrote: “This software could totally be useful in my field (structural geology). Research projects in my field typically collected 20+ variables which are almost never visualized or compared due to not having software capable of doing this. Being able to show all variables across multi-year projects might show relationships that were not considered before.” Two scientists answered maybe since they either do not work with complex time series or do not conduct open-end exploratory analyses.

Do you think the visualizations and functions are useful in data exploration? Ten scientists answered yes to this question. They commented that “based on the demo EVis is a really novel and useful set of visualizations” and “is great for opening up the possibilities of seeing connections that one might not have been previously looking for”. They confirmed the usefulness of RadViz-Leash: “Rock properties are inherently linked but it’s very difficult to see how many trends co-vary. The plots of data leashes show how the complex strain fields vary leading up to deformation events, which is very difficult to visualize, even with models and lab analysis. This is often done as 2D models which lose a lot of the important details. The tails allow you to see what is happening and better interpret the data.” “RadViz is a useful quantitative visualization tool to explore relations within multi-variable data and in the time domain. The ’leash’ function is extremely useful to help zoom in on specific data streams and their relationship with independent measures of process." They also commented that the basic visualization charts and the interactions are beneficial. One scientist answered not sure since he does not use complex time series.

What are the potential benefits/advantages (if any) of exploring data using this software compared with existing approaches researchers are using in your field? The scientists commented their work was done piecemeal and they were not aware of any visualization tools tailored to their needs. They wrote "EVis seems better customized to flexible numbers of variables and to providing a straightforward workflow compared to existing more generic software" and "while EVis does not prove causation, it can shed light on potential connections. Traditional tools require you to actively look for specific connections (or lack thereof) that you had already suspected." They commented "EVis has a strong potential to attract many more of us" and "EVis has the potential to be a game changer".

What are the limitations? Please provide your top 2 suggestions to improve the software The scientists provided a set of suggestions to improve the usability and functionality of EVis. Two commented that the detail view was busy with too many plots. To address this problem, I (1) added the dissimilarity background, (2) added the sorting interactions so that users can focus on more interesting POIs, and (3) added the filtering interaction to hide redundant leashes. The scientists also suggested extending the applicability of EVis from the following aspects: (1) add 3D spatial visualization; (2) add statistics and signal analysis capabilities, such as basic 1D spectral analysis (e.g., FFT and wavelets); (3) provide thorough documentation, tutorial videos, and a user manual with examples; (4) allow users to upload datasets with different time scales, provide build-in tools to extrapolate and interpolate data between points, and allow users to create new variables via calculations; and (5) provide output functions for further analysis and make EVis compatible with other software such as R. I will add these functions into EVis in the future.

5.3 Discussions

Compared with popular dimension reduction methods, RadViz is less familiar to Earth scientists and, therefore, has a steeper learning curve. However, it supports required tasks better than existing commonly employed techniques. The projection of a data point generated by MDS [29, 39], PCA [40, 25], t-SNE [41], and UMAP [27] is influenced by other data points in the input. Therefore, leashes generated by them will change when the set of POIs changes. RadViz does not have this problem—the projection of a data point is independent of other data points. Moreover, RadViz allows users to examine data points/leashes in relation to their attributes, which is a desired feature for Earth scientists. PCA [40] and related displays, such as biplots [42], are not optimized for this task [43]. MDS [29, 39] and t-SNE [41] cannot support this task explicitly since they lose the data-dimension relationships in the projection process [43].

In EVis, users can interactively adjust RadViz, which will be propagated to leashes, using distortion and anchor relocation interactions. I believe these interactions are necessary and beneficial. First, without distortions, with an anchor setting as the one used in the first case study, most points are clustered at the center of RadViz, which makes insight discovery and leash clustering extremely difficult. Second, since distortions magnify subtle differences in data, RadViz-Leash can group data with those differences captured, which helps users capture insights hidden from other analysis approaches. Third, the process of interactively adjusting RadViz is a visual exploration process for insights—when users see interesting patterns in RadViz (e.g., big bubbles are clustered in RadViz), they can fix the layout and make selections from RadViz (e.g., selecting the big bubbles clustered) for further analyses from other views. This is a unique approach unavailable in existing practices. A limitation of our current approach is that users may need to experiment with multiple layouts to find interesting ones. In the future, I will study how to leverage this with automated approaches so that users can conduct explorations more efficiently.

5.4 Conclusion

In this paper, I propose EVis, a new visual analytics prototype for Earth scientists. It provides a rich set of visualizations and interactions for exploratory analyses of natural phenomena and their driving E-conditions. The case studies with a domain expert illustrated the usefulness of EVis in rock mechanical weathering studies. Feedback from eleven scientists from varying sub-disciplines of Earth science revealed the broad applicability and potential usefulness of EVis in Earth science. Meanwhile, my practice of RadViz on Earth science further illustrates the functionality and uniqueness of this long-existing dimension reduction technique. The evaluations also demonstrated the usefulness of RadViz-Leash, the new RadViz projection and clustering-based visual analytics approach for multivariate time series analyses I proposed.

I believe EVis has the potential to bridge the gap between Earth scientists and their visualization needs. To be fully employed by scientists, input and output functions, insight management functions, signal analysis and statistics functions commonly used by Earth scientists, and more functions the domain experts suggested in Section 5.2 still need to be added into EVis.

CHAPTER 6: EVis 2.0 and a Case Study on Rockfalls at Yosemite National Park

6.1 Introduction

Rockfall, which is a natural and dynamic geologic process involving the detachment and rapid downward movement of rock, has been a serious threat to people's safety [70, 71, 72]. The earliest study found for rockfalls was given by Ritchie in 1963 [73]. He proposed simple guidelines for the design of boulder trap ditches or trenches at the toes of rock slopes. The time to start collecting rockfall data was even earlier. Earth scientists have been studying rockfalls to reduce the hazard to human life.

Yosemite National Park, which contains a beautiful great valley, experiences many rockfalls each year. Historical records indicate that more than 1,000 rockfalls have occurred in the park during the past 150 years [73]. Earth scientists have been collecting rockfall information since 1857, including location, time, and types of rockfalls. Additionally, Earth scientists have linked climate data of the park to the rockfalls to discover the relationships between the rockfalls and the climate. The dataset is called the Yosemite Rockfalls dataset.

EVis is a promising tool for analyzing the temporal lagging relationship and other relationships between rockfalls and the climate. However, EVis suffered from the following problems when visualizing the Yosemite Rockfalls dataset. First, there are a large number of missing data in the dataset since it covers a very long period (1905-2017). Missing data processing was not addressed in EVis since missing data has been removed from the New Mexico Rock dataset, the dataset used to develop EVis. Second, the rockfall events have multiple categorical attributes, which can't be processed in EVis. Third, the Yosemite Rockfalls dataset has a large number of attributes and data records, which makes the rendering of the visualizations too slow to support interactive visual explorations. In addition, the New Mexico Rock dataset evenly sampled environmental conditions while the Yosemite Rockfalls dataset only recorded environmental conditions when there were events in many years. Therefore, the event rate calculation needs to be revised to handle this issue. I upgraded EVis to EVis 2.0 to address these issues so that the Yosemite Rockfalls dataset can be analyzed.

In this chapter, I will describe the Yosemite Rockfalls dataset in detail and introduce the new features of EVis 2.0 to address the aforementioned issues. They include a new rendering method, missing data processing capacity, new categorical event attributes visualization, and a new data import interface. The new features make EVis 2.0 a more general visualization tool than EVis since the aforementioned issues are common in datasets analyzed by Earth scientists.

6.2 Data Description

The Yosemite Rockfalls dataset consists of rockfall data and climate data, both of which are recorded daily. Each rockfall is considered an event, similar to the AE events of the New Mexico rock cracking dataset. Scientists collected rockfall records from 1857 to 2020. A total of 1,489 records were collected. Each record has 28 columns. They contain the date and location of the occurrence, size of the rock, damage caused, triggering conditions, and narrative descriptions. To study this dataset, after discussing with Earth scientists, I picked the median volume of the rock as an event measure (similar to the cracking energy of the New Mexico Rock dataset). Triggering Condition and General Location are considered as categorical attributes of the events. They are visually explored in EVis 2.0 using the new features.

The daily climate data records of Yosemite park came from two datasets. One is the climate records of Yosemite park headquarter from January 12, 1905, to December 31, 2020, with climate variables of minimum, maximum, and average of temperature, precipitation, and snow. The other is the climate records of Yosemite village from

September 28, 2007, to December 31, 2020, with the minimum, maximum, and average value of daily temperature, surface temperature, and relative humidity, along with vapor pressure, and soil moisture and soil temperature for 5, 10, 20, 50, and 100 days.

The rockfall records and climate records were joined via time to get 41,003 daily **discrete-time-point**. It includes 858 **event days** and 40,145 **non-event days**. The rockfall events out of the time range of the climate records are not collected. There are much missing data in the dataset.

6.3 System Improvement

6.3.1 SCanvas, a JS Library of Scalable Canvas

The two most used rendering containers of Web Applications are **svg** and **canvas**. The HTML element **svg** is a vector-based graphic container with high flexibility for interactions since the container renders the visualization as vector graphics. However, when there are a large number of graphical objects, the rendering of **svg** turns slow. On the other hand, the HTML 5 element **canvas** is a raster-based graphic container and has better performance than **svg** with a large number of graphical objects. However, a **canvas** with all rendered objects is like a painted picture. It needs to be re-rendered with any tiny changes, which makes a **canvas** not easy to be rescaled, updated, interacted with, and resized.

To take advantage of both techniques and overcome their disadvantages, I developed a JS library **SCanvas**, which stands for Scalable Canvas, to improve rendering performance without losing interactivity. The container of **SCanvas** consists of a **canvas** layer and an overlapping **svg** layer. The **canvas** layer is used to render graphical objects and the **svg** layer provides event listeners to respond to users' operations. The container provides an attribute **viewBox** that is similar to the **viewBox** of **svg**. Users calculate graphics according to their relative positions and sizes to the **viewBox** without considering the real size of the container. The build-in rescaling function handles the rescaling of the container and synchronizes the two layers to map users' operations to the rendered graphical objects.

I conducted an experiment comparing the rendering speed with D3 and the rendering speed with **SCanvas**. The experiment used D3 and **SCanvas** to render a million circles from the same synthesized dataset respectively. As a result, D3 rendering took 12.15 seconds, and **SCanvas** rendering took just 1.54 seconds for this task. EVis 2.0 have employed **SCanvas** for rendering to improve the performance of the whole application.

6.3.2 Missing data processing

When processing the New Mexico Rock data, I used linear interpolation to fill missing values in the time series data. However, the Yosemite Rockfalls dataset contains climate records from two datasets with different time ranges and even different variables. We got many missing values while combining them together. For example, the daily average temperature at Yosemite part headquarter were recorded from January 12, 1905, to December 31, 2020, and the vapor pressure at Yosemite village and were recorded from September 28, 2007, to December 31, 2020 (Figure 6.1). In this case, it is impossible to apply any interpolation algorithm to fill a big missing data for vapor pressure. So, EVis 2.0 needs methods to handle the missing data, especially in the value-time scatterplots and the RadViz view.

A value-time scatterplot maps discrete-time points to a 2D 'value vs. time' scatterplot as bubbles. Even if the value of a specific dimension is missed, the bubble still contains time and event information. Hollow gray bubbles represent non-event days and the solid blue bubble represents event days with size representing the sum of the volumes of all rocks that fell on the day or the count of how many rocks fell. So a bubble needs to be kept even if the value on the Y dimension is missing. A horizontal line is added above the scatterplot and all bubbles with missing values are placed on it according to their timestamps. They are displayed as hollow circles to be

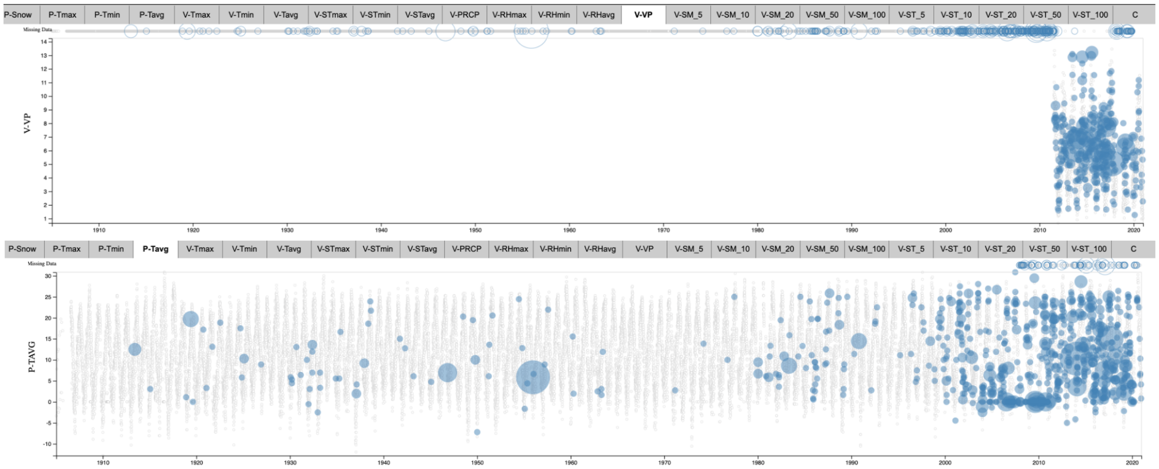


Figure 6.1: A value-time scatterplot for vapor pressure at Yosemite village (upper) and average daily temperature at Yosemite park headquarter (lower). There are a large number of days with missing values in the upper chart for quite a long period of time. They are displayed on the line above the scatterplot range as hollow bubbles so that analysts can still examine their time and the event information.

distinguished from solid event bubbles in the scatterplot. Analysts can still examine their time and event information (Figure 6.1).

The position of a data point in RadViz is decided by its values on the anchor dimensions. If the value on any of the anchor dimensions is missing, the position will be incorrect. So, in RadViz, all data with missing values on any of the anchor dimensions are removed from the RadViz view.

In RadViz-Leash, leashes with missing values on the anchor dimensions are removed. However, leashes with no missing values on the anchor dimensions may have missing values on non-anchor dimensions. They may cause clutter in the line charts of those dimensions in the RadViz-Leash detail view. To address this problem, points with a missing value on a dimension are removed from the line chart of that dimension, together with line segments connecting them to their adjacent points in the line chart. Figure 6.2 shows an example of a line chart matrix with missing values in the detail view. The call-out shows some line charts with missing values. It reserves more information than the solution of removing the whole leashes with missing data from

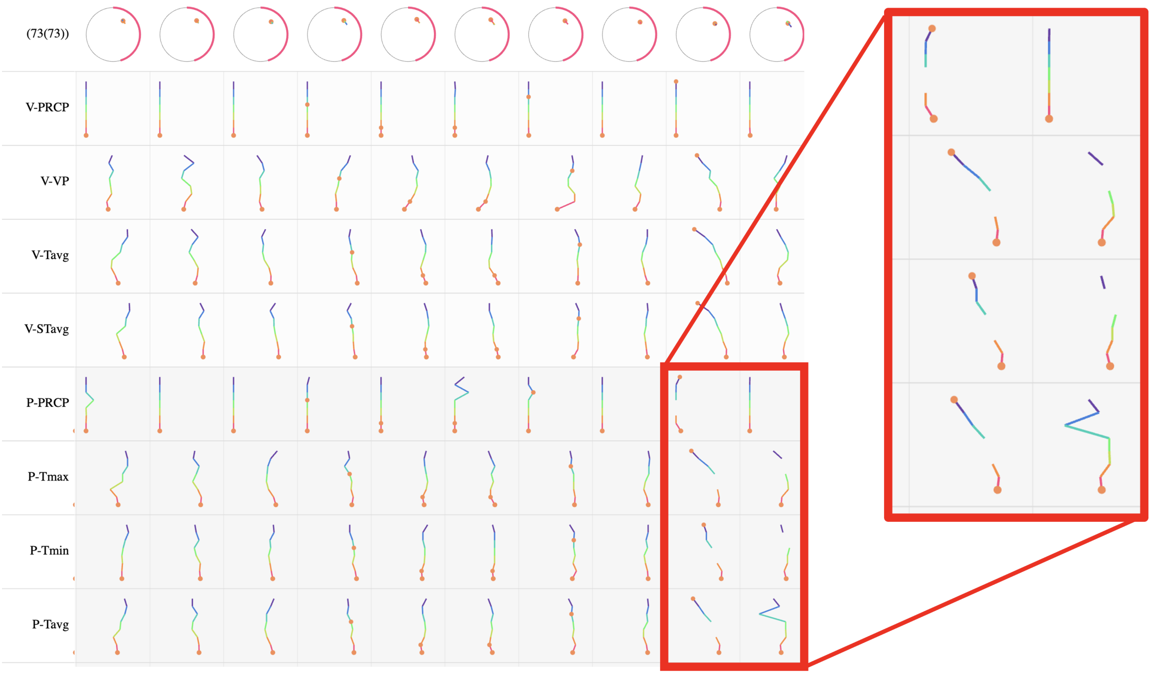


Figure 6.2: An example of the timeline matrix. Rows with a light background are anchor dimensions. The red frame highlights timelines of non-anchor dimensions with missing data.

the visualization without introducing clutters.

6.3.3 Categorical attributes of events

Each rockfall event has two categorical attributes, the triggering condition and the location of the event. It is important to allow users to filter data based on these attributes to analyze events under different triggering conditions or at different locations. Two new histograms, one for triggering conditions and another one for locations, are added to EVis 2.0. The triggering condition/location histogram has a bar for each triggering condition/location and the height of a bar indicates how many event days encountered the corresponding condition/happened at the corresponding location (Figure 6.3). Since only events have triggers, no non-event days will be selected when a user clicks a trigger bar. To provide context information to event days selected using a trigger bar, a gray bar representing non-event days is added to the triggering condition histogram. Users can select it along with the triggers to

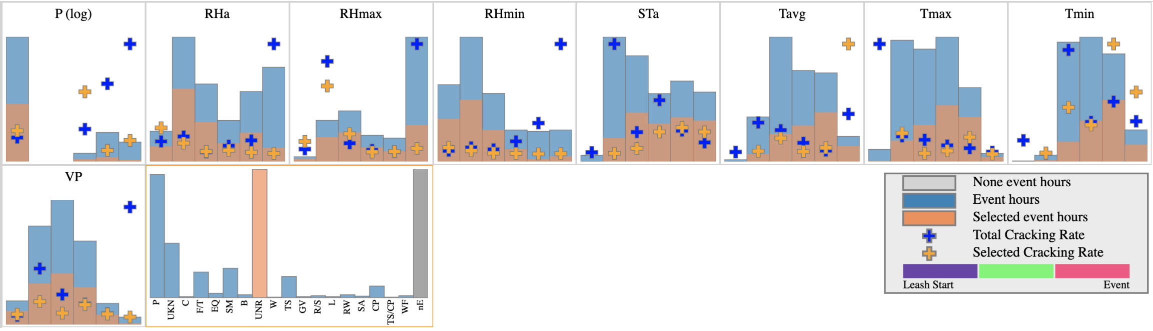


Figure 6.3: The triggering condition histogram (orange frame) for selecting events according to triggering conditions. It has a bar for each possible triggering condition whose height represents the count of event days with that trigger. The Y-position of the orange cross on a bar represents the event rate of that trigger, calculated with the enhanced event rate calculation method. All non-event days are represented by a gray bar.

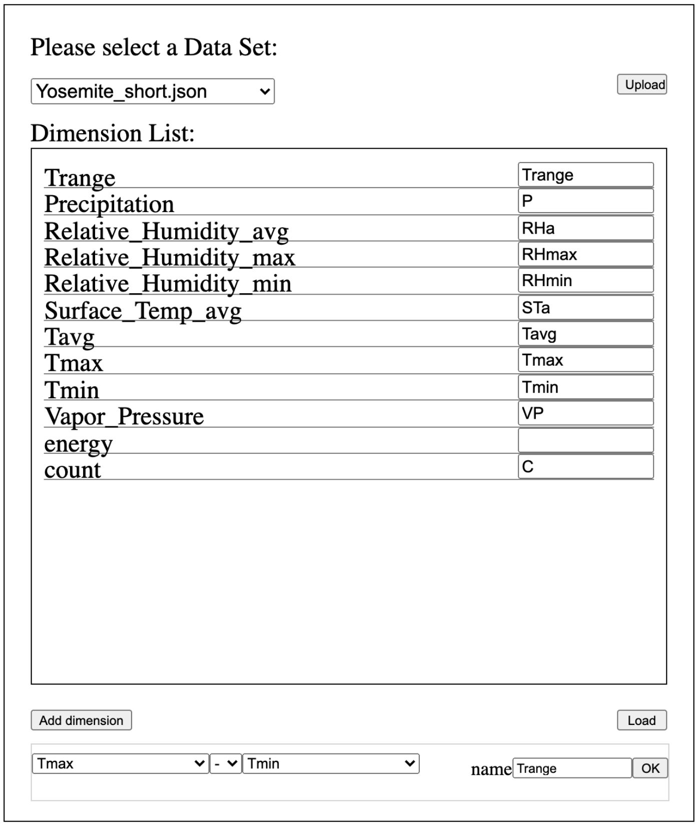
provide context to the event days selected. The non-event days are displayed as gray circles in other views such as scatterplot and RadViz.

6.3.4 New data import interface

To import the Yosemite Rockfalls datasets and other datasets to EVis 2.0, a new data import interface (see Figure 6.4) has been added to EVis 2.0. It allows users to upload new datasets to the database of EVis 2.0 and retrieve an existing dataset from the database to be visually explored in EVis 2.0. After a dataset is selected, users can select dimensions in the dataset to be visualized and assign an alias to them (the alias will be displayed as dimension labels in the visualizations) via this interface. In addition, users can create new dimensions derived from original dimensions using arithmetic operations and visualize them in the visualizations. In Figure 6.4, a new dimension $T_{range} = T_{max} - T_{min}$ has been created.

6.4 Case Study

When collecting the rockfall records, Earth scientists marked the triggering conditions that might cause a rockfall to occur. The triggers include but are not limited to earthquake, precipitation (rain or snow, or combination of both), snow avalanche,



Please select a Data Set:

Yosemite_short.json

Dimension List:

Trange	Trange
Precipitation	P
Relative_Humidity_avg	RHa
Relative_Humidity_max	RHmax
Relative_Humidity_min	RHmin
Surface_Temp_avg	STa
Tavg	Tavg
Tmax	Tmax
Tmin	Tmin
Vapor_Pressure	VP
energy	
count	C

Tmax - Tmin name Trange

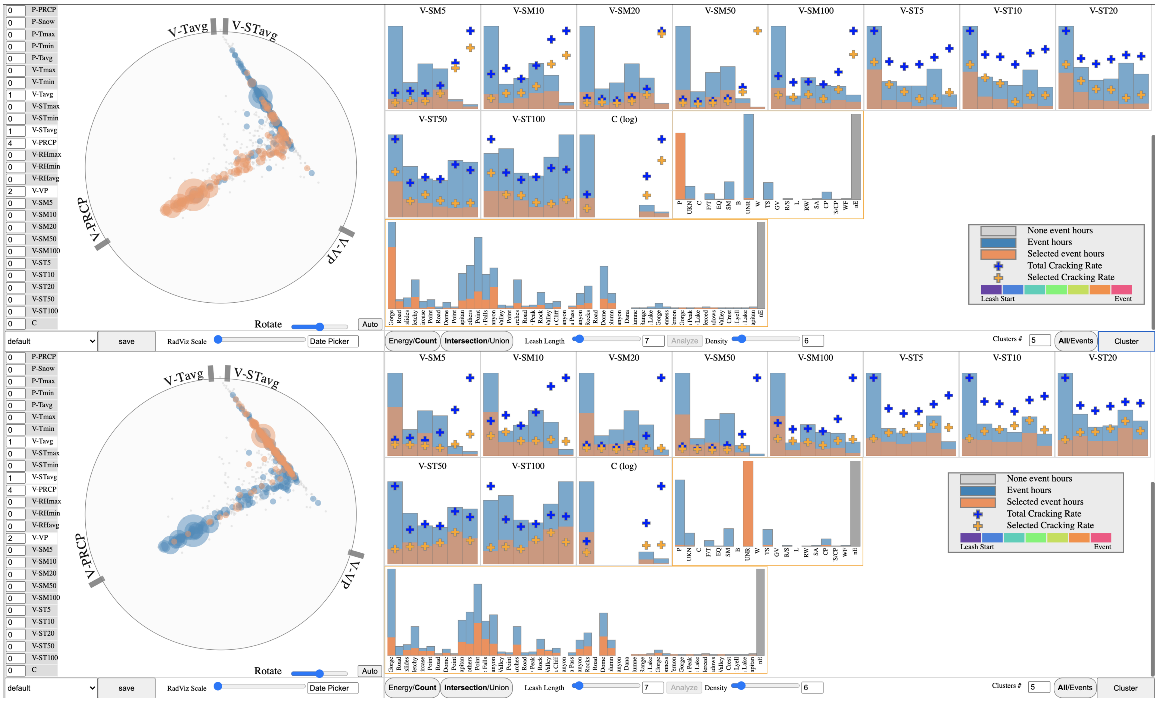
Figure 6.4: The data import interface. Users can upload a new data set to the EVis 2.0 database and select a dataset from the database for visual exploration. All dimensions of the selected dataset are displayed on the left of the list. Users can add a dimension to the visualization by typing an alias for the dimension on the right column. Users can create new dimensions derived from original dimensions using the entries placed at the bottom of the interface.

lightning, wind, wildfire, etc. Among the triggers, there is a category marked as 'unrecognized trigger', which indicates the observers were not sure of the reason that caused the rockfall. In this case study, I worked with a senior Earth scientist to explore how environmental conditions changes are related to rockfall events.

Since there were too much missing data in most dimensions to conduct effective analysis in climate records before 2007, we used a subset of the Yosemite Rockfall dataset in this study. It has a time range from September 28, 2007, to December 31, 2020. To simplify the settings of this case study, the Earth scientist only considered temperature, precipitation, and vapor pressure. They were considered as major contributors to the rockfall events according to previous research. The scientist set daily average temperature and surface temperature, precipitation, and vapor pressure from Yosemite village as anchors of RadViz (see left of Figure 6.5). She put the two temperature variables very close to each other on RadViz. The other two anchors, the precipitation and vapor pressure, were distributed evenly on the circumference of RadViz. To make the data points better distributed in RadViz, the vapor pressure was weighted to 2 and the precipitation was weighted to 4.

In the screenshot on top of Figure 6.5, events with a precipitation triggering condition were selected using the triggering condition histogram. Through the RadViz view, it can be easily seen that the precipitation triggered rockfall events are distributed in the whole value range of precipitation. In the screenshot at bottom of Figure 6.5, events with an unrecognized triggering condition were selected. It can be seen that the selected points (events with unrecognized triggers) are mainly distributed in the area with no precipitation. To further explore the subset of events with an unrecognized triggering condition, the scientist set the leash length to 7 days and set the number of clusters to 5, and ran K-Means clustering. The clustering result is shown in Figure 6.6.

The resulting five clusters revealed several potential relationships between rockfall

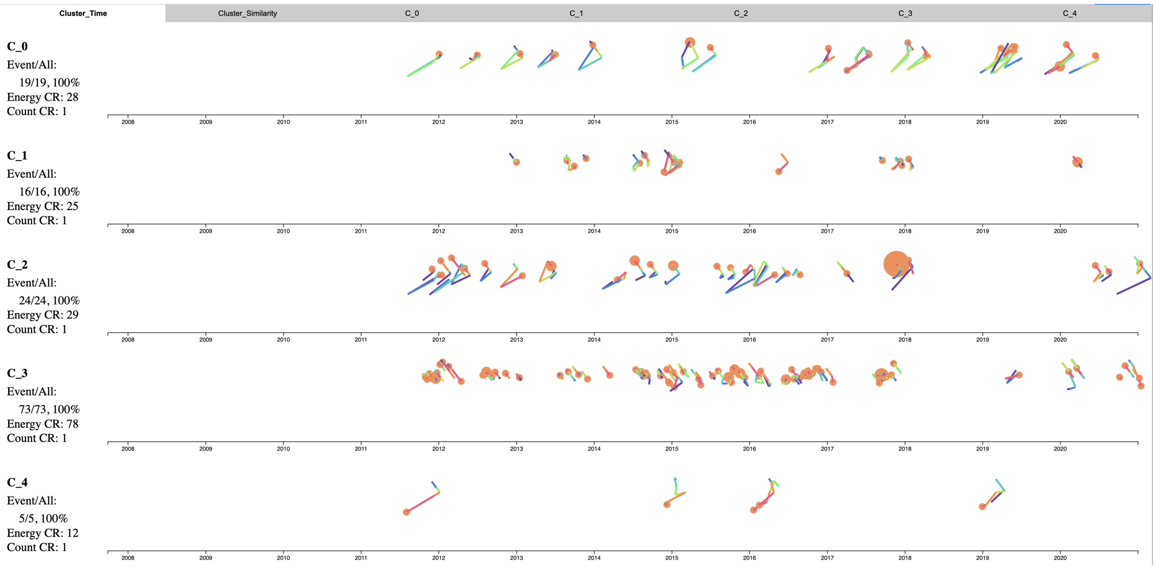


P-PRCP	0
P-Snow	0
P-Tmax	0
P-Tmin	0
P-Tavg	0
V-Tmax	0
V-Tmin	0
V-Tavg	1
V-STmax	0
V-STmin	0
V-STavg	1
V-PRCP	4
V-RHmax	0
V-RHmin	0
V-RHavg	0
V-VP	2
V-SM5	0
V-SM10	0
V-SM20	0
V-SM50	0
V-SM100	0
V-ST5	0
V-ST10	0
V-ST20	0
V-ST50	0
V-ST100	0
C	0

default save RadViz Scale Date Picker

Energy/Count Intersection/Union Leash Length 7 Analyze Density 6 Clusters # 5 All/Events Cluster

Figure 6.5: The settings for the case study. Top: Events with a precipitation triggering condition were selected using the histogram. Bottom: Events with an unrecognized triggering condition were selected. Left of both figures: The RadViz view. Two temperature anchors were put close to each other. The weight of Vapor Pressure was set to 2 and the weight of Precipitation was set to 4 to make the whole shape balanced.



Cluster_Time Cluster_Similarity C_0 C_1 C_2 C_3 C_4

C_0
Event/All:
19/19, 100%
Energy CR: 28
Count CR: 1

C_1
Event/All:
16/16, 100%
Energy CR: 25
Count CR: 1

C_2
Event/All:
24/24, 100%
Energy CR: 29
Count CR: 1

C_3
Event/All:
73/73, 100%
Energy CR: 78
Count CR: 1

C_4
Event/All:
5/5, 100%
Energy CR: 12
Count CR: 1

Figure 6.6: The clustering result of leashes of events with an unrecognized triggering condition.

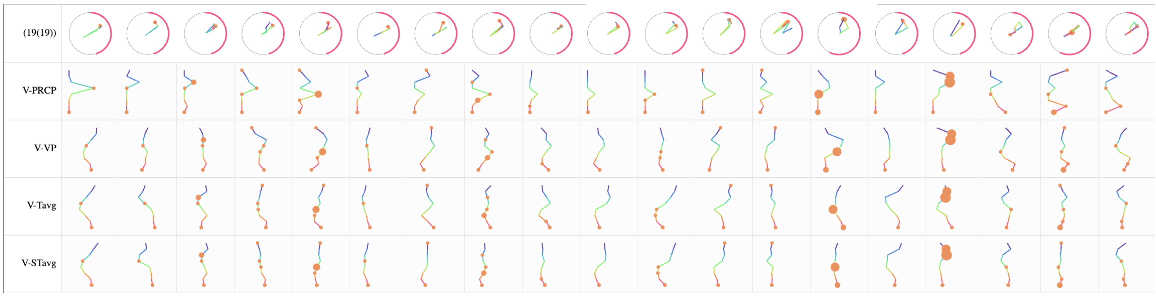


Figure 6.7: The detailed view of cluster *C_ 2* with a timeline charts matrix.

events and their antecedent environmental conditions. In the clusters *C_ 0* and *C_ 2*, even though the events occur on days without precipitation, the leashes indicate that before the events happened, the precipitation was going up and then down. Then the temperature increased till the events occurred. The difference between clusters *C_ 0* and *C_ 2* was the colors of leashes. According to the colors, the precipitation increase happened around four days before the events in cluster *C_ 0* and seven days before the events in cluster *C_ 2*.

To check the detailed information of this pattern, the scientist opened the detailed view of cluster *C_ 0*, shown in (Figure 6.7). The first row of the matrix shows leashes of all events in this cluster. Below the first row is a matrix of line charts. It can be seen that there was always a peak of precipitation around four days before the event and the precipitation always reduced to zero right when the events happened. There is also a slight rise in the two temperature variables when the events happened.

This pattern is among multiple hidden patterns of how environmental condition changes are related to the rockfall events we discovered using EVis 2.0. It is difficult to discover such patterns with traditional analysis methods.

6.5 Conclusion

This chapter presented EVis 2.0, an expansion of EVis to improve its applicability. A data import interface has been added to allow EVis 2.0 to upload customized data (in a given format). The data import interface also provides a way for users to create derived dimensions from original dimensions. To improve the rendering performance, I developed a standalone JavaScript library, **SCanvas**, to set up a scalable HTML **canvas** with **svg** combined. The library improves the rendering performance while keeping as much scalability for interactions as possible. The EVis 2.0 also handles missing data. For scatterplots, data points with missing values on the Y dimension are displayed out of the scatterplot. But in RadViz, any data point with a missing value of any anchor dimensions will be hidden since the missing data affects the position of the data point on the RadViz space, which may mislead users to data distributions on the anchor dimensions. EVis 2.0 visualizes categorical event attributes via histograms and allows users to select events based on those attributes.

To evaluate EVis 2.0, the Yosemite Rockfalls dataset was loaded to EVis 2.0 and a case study was conducted with a domain expert. Rockfall events with different triggering conditions were analyzed and leashes with different patterns were discovered. They revealed the varying antecedent environmental conditions of the rockfall events. This evaluation shows the ability of EVis 2.0 to explore the temporally lagging relationships between rockfall events and changes in environmental conditions.

CHAPTER 7: Summary and Future Direction

7.1 Summary

Multidimensional time-series data visualization always attracts and challenges data analysts. In this dissertation, I demonstrated two design studies to reflect the efforts in multidimensional time-series data visualization.

In Chapter 1, I introduced the typical workflow of data science and the role of data visualization in the process. I then demonstrated the importance of multidimensional time-series data visualization and explained the challenges in this field.

In Chapter 2, I discussed existing approaches to multivariate time series visualization. Approaches such as TimeSearcher2 [2] coordinated multiple views of univariate time series; approaches such as Fujiwara et al. [24] applied dimensional reduction method to visualize the multivariate time series in a 2D plane; approaches such as Takami and Takama [3] used animation to show how multivariate status changes over time; approaches such as a trajectory-based financial time series visualization proposed by Schreck et al. [30] created trajectory to represent the time process and implement unsupervised machine learning to analyze the trajectories. All but not limited to these approaches inspired my research. I also introduced RadViz in this chapter and discussed existing efforts to extend its usability. These efforts were mainly focused on developing algorithms to reduce the clutter of data points on the RadViz plane. The main purpose of RadViz in existing work was to classify data points and examine data clusters. In the visual analytics approaches I presented in this dissertation, RadViz is used to visualize multivariate time series, which is an innovative new use case for RadViz.

In Chapter 3, I proposed *t*-RadViz, a new, fulling working prototype that is currently being employed by an automobile manufacturer to analyze the real-time, streaming continuous motion workbench data. t-RadViz uses **socket.io** to build the communication between a server end and a front end to achieve real-time data detection. t-RadViz provides users a comprehensive visual interface that can be interactively customized to address their analysis needs. t-RadViz is among the first efforts to use RadViz for visualizing multidimensional time series. It projects real-time multidimensional time series to a 2D RadViz plane to form time-oriented trajectories to examine temporal trends and compare multiple time series. I also developed an algorithm to synchronize line charts of multiple testbench runs based on their projections on RadViz. In addition, t-RadViz provides coupled delta bar charts and line charts to help users compare the performance of multiple testbench runs on the fly. t-RadViz received positive feedback from its target users and is currently used in almost all their continuous motion tests.

In Chapter 4, I proposed EVis, another new, fulling working prototype to analyze time series of multiple environmental conditions and rock cracking events. EVis provides basic visualizations such as scatterplots, heatmap, and histograms to allow users to interactively select an interesting subset and explore the relationship between events and different dimensions. EVis also provides a RadViz view to project the multiple environmental conditions along with events to the 2D RadViz plane. I also developed a new visual analytics approach called RadViz-Leash for discovering temporally lagging relationships between events and antecedent multiple environmental conditions. RadViz-Leash projects multidimensional time series of antecedent environmental conditions of events to trajectories, called leashes, on the RadViz plane, calculates similarities among the leashes based on distances of the trajectories with temporal attributes considered, and clusters the leashes based on the similarities for interactive visual exploration of temporally lagging relationships between events and antecedent multiple environmental conditions. All the visualizations are coordinated for flexible analyses.

In Chapter 5, to evaluate EVis, I worked with a senior Earth scientist to conduct two case studies using EVis to study rock mechanical weathering. The first case confirmed multiple insights from the existing knowledge of the Earth scientist. The second case discovered a few insights the Earth scientist had not expected. Besides the case studies, I also collected feedback from a set of Earth scientists of a range of sub-disciplines. The feedback was positive and many suggestions were collected for the future development of EVis.

In Chapter 6, I proposed EVis 2.0 with significant updates to improve the applicability of EVis according to the feedback from the Earth scientists. A data management interface has been added to allow users to upload data and derive dimensions. A standalone library, named **SCanvas**, has been developed and employed in EVis 2.0 to improve its rendering performance on large datasets. EVis 2.0 also handles missing data, supports the visualization of categorical variables, and uses an enhanced event rate calculation method in the histogram. A case study has been conducted with EVis 2.0 to analyze the relationship between rockfall events and the climate of Yosemite national park. It illustrated the applicability and effectiveness of EVis 2.0.

7.2 Future Direction

Visualization of time series data always attracts researchers' attention and multidimensional time-series visualization has been a challenge for a long time. With more and more needs for time-oriented data analyses, the methods proposed in this dissertation will be valuable for future research.

The automobile manufacturer is planning to add more dimensions to the data stream. It means t-RadViz is facing the challenge of more data with more complex information. Using **SCanvas** library to increase the rendering speed of t-RadViz to support real-time updates of the visualizations is under discussions with engineers of the manufacturer, the true end users. Also, the method I proposed in this dissertation to synchronize incoming data with matching historical data segments seems to have the potential to be applied in more application areas.

EVis and EVis 2.0 have been used to analyze rock mechanical weathering and rockfall in Yosemite national park. It is exciting that scientists from many sub-disciplines of Earth sciences have shown their interests in applying EVis to their research areas. Many research topics in Earth sciences share a similar scenario that involves multiple environmental variables changing over time and time-oriented events. It is important to discover the hidden relationship between the event occurrences and the environmental changes. EVis is good for analyzing this type of relationship and thus it will be valuable for these applications. Besides Earth sciences, other disciplines may also have similar challenges. One of the future directions of EVis development is to make it a more general tool for more application of more areas.

During my research, I have developed a standalone library **SCanvas** to improve the rendering speed without damaging the interactivity of HTML graphics. I also plan to create standalone libraries for the rich set of RadViz interactions and RadViz-Leash I developed in my research. These standalone libraries will allow more researchers to reuse my research results in their applications.

REFERENCES

[1] J. Kohlhammer, D. Keim, M. Pohl, G. Santucci, and G. Andrienko, "Solving problems with visual analytics," *Procedia Computer Science*, vol. 7, pp. 117–120, 2011.

[2] P. Buono, A. Aris, C. Plaisant, A. Khella, and B. Shneiderman, "Interactive pattern search in time series," in *Visualization and Data Analysis 2005*, vol. 5669, pp. 175–186, International Society for Optics and Photonics, 2005.

[3] R. Takami and Y. Takama, "Visual analytics interface for time series data based on trajectory manipulation," in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 342–347, 2018.

[4] S. Cheng, K. Mueller, and W. Xu, "A framework to visualize temporal behavioral relationships in streaming multivariate data," in *2016 New York Scientific Data Summit (NYSDS)*, pp. 1–10, 2016.

[5] S. Havre, B. Hetzler, and L. Nowell, "Themeriver: Visualizing theme changes over time," in *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, pp. 115–123, IEEE, 2000.

[6] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic, "Time curves: Folding time to visualize patterns of temporal evolution in data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 559–568, 2016.

[7] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, "Reducing snapshots to points: A visual analytics approach to dynamic network exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 1–10, 2016.

[8] L. Novakova and O. Štepanková, "Radviz and identification of clusters in multidimensional data," in *2009 13th International Conference Information Visualisation*, pp. 104–109, IEEE, 2009.

[9] M. Eppes, B. Magi, J. Scheff, K. Warren, S. Ching, and T. Feng, "Warmer, wetter climates accelerate mechanical weathering in field data, independent of stress-loading," *Geophysical Research Letters*, p. e2020GL089062, 2020.

[10] J. F. Roddick and M. Spiliopoulou, "A survey of temporal knowledge discovery paradigms and methods," *IEEE Transactions on Knowledge and data engineering*, vol. 14, no. 4, pp. 750–767, 2002.

[11] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, pp. 37–37, 1996.

[12] E. Keogh, S. Chu, D. Hart, and M. Pazzani, “Segmenting time series: A survey and novel approach,” in *Data mining in time series databases*, pp. 1–21, World Scientific, 2004.

[13] M. L. Hetland, “A survey of recent methods for efficient retrieval of similar time sequences,” in *Data mining in time series databases*, pp. 23–42, World Scientific, 2004.

[14] G. Zeira, O. Maimon, M. Last, and L. Rokach, “Change detection in classification models induced from time series data,” in *Data mining in time series databases*, pp. 101–125, World Scientific, 2004.

[15] H. G. Funkhouser, “A note on a tenth century graph,” *Osiris*, vol. 1, pp. 260–262, 1936.

[16] E. Bertini, “From data visualization to interactive data analysis,” Dec. 2017.

[17] “A picture is worth a thousand words,” May 2020.

[18] K. A. Cook and J. J. Thomas, “Illuminating the path: The research and development agenda for visual analytics,” tech. rep., Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2005.

[19] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, *Visualization of time-oriented data.* Springer Science & Business Media, 2011.

[20] B. Horst and K. Abraham, *Data mining in time series databases*, vol. 57. World scientific, 2004.

[21] C. Tominski, J. Abello, and H. Schumann, “Axes-based visualizations with radial layouts,” in *Proceedings of the 2004 ACM symposium on Applied computing*, pp. 1242–1247, 2004.

[22] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman, “Lifelines: using visualization to enhance navigation and analysis of patient records,” in *The craft of information visualization*, pp. 308–312, Elsevier, 2003.

[23] E. Bertini, L. Dell’Aquila, and G. Santucci, “Springview: Cooperation of radviz and parallel coordinates for view optimization and clutter reduction,” in *Coordinated and Multiple Views in Exploratory Visualization (CMV’05)*, pp. 22–29, IEEE, 2005.

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

[25] T. Fujiwara, J. K. Chou, S. Shilpika, P. Xu, L. Ren, and K. L. Ma, "An incremental dimensionality reduction method for visualizing streaming multidimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 418–428, 2020.

[26] T. Fujiwara, J.-K. Chou, S. Shilpika, P. Xu, L. Ren, and K.-L. Ma, "An incremental dimensionality reduction method for visualizing streaming multidimensional data," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 418–428, 2019.

[27] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[28] D. Yang, Z. Dong, L. H. I. Lim, and L. Liu, "Analyzing big time series data in solar engineering using features and pca," *Solar Energy*, vol. 153, pp. 317–328, 2017.

[29] J. B. Kruskal, *Multidimensional scaling.* No. 11, Sage, 1978.

[30] T. Schreck, T. Tekušová, J. Kohlhammer, and D. Fellner, "Trajectory-based visual analysis of large financial time series data," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 30–37, 2007.

[31] T. Kohonen, *Self-organizing maps*, vol. 30. Springer Science & Business Media, 2012.

[32] S. Ghassempour, F. Girosi, and A. Maeder, "Clustering multivariate time series using hidden markov models," *International journal of environmental research and public health*, vol. 11, no. 3, pp. 2741–2763, 2014.

[33] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: A review," *SIGKDD Explor. Newsl.*, vol. 6, p. 90â105, June 2004.

[34] Y. Chen, L. Zhou, S. Pei, Z. Yu, Y. Chen, X. Liu, J. Du, and N. Xiong, "Knn-block dbscan: Fast clustering for large-scale data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–15, 2019.

[35] W. Meesrikamolkul, V. Niennattrakul, and C. A. Ratanamahatana, "Shape-based clustering for time series data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 530–541, Springer, 2012.

[36] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[37] T. Dasu, D. F. Swayne, and D. Poole, "Grouping multivariate time series: A case study," in *Proceedings of the IEEE Workshop on Temporal Data Mining: Algorithms, Theory and Applications, in conjunction with the Conference on Data Mining, Houston*, pp. 25–32, Citeseer, 2005.

[38] J. A. Hartigan, *Clustering Algorithms.* USA: John Wiley Sons, Inc., 99th ed., 1975.

[39] J. H. Lee, K. T. McDonnell, A. Zelenyuk, D. Imre, and K. Mueller, "A structure-based distance metric for high-dimensional space exploration with multidimensional scaling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 351–364, 2014.

[40] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[41] P. E. Rauber, A. X. Falcão, A. C. Telea, *et al.*, "Visualizing time-dependent data using dynamic t-sne.," p. 73â77, 2016.

[42] K. R. Gabriel, "The biplot graphic display of matrices with application to principal component analysis," *Biometrika*, vol. 58, no. 3, pp. 453–467, 1971.

[43] S. Cheng, W. Xu, and K. Mueller, "Radviz deluxe: An attribute-aware display for multivariate data," *Processes*, vol. 5, no. 4, p. 75, 2017.

[44] M. Angelini, G. Blasilli, S. Lenti, A. Palleschi, and G. Santucci, "Towards enhancing radviz analysis and interpretation," in *2019 IEEE Visualization Conference (VIS)*, pp. 226–230, 2019.

[45] F. Zhou, W. Huang, Y. Zhao, Y. Shi, X. Liang, and X. Fan, "Entvis: A visual analytic tool for entropy-based network traffic anomaly detection," *IEEE Computer Graphics and Applications*, vol. 35, no. 6, pp. 42–50, 2015.

[46] H. Adams, *Chassis engineering.* Penguin, 1993.

[47] G. Rauch, "Socket.io: the cross-browser websocket for realtime apps.," 2012.

[48] J. Heer, N. Kong, and M. Agrawala, "Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1303–1312, 2009.

[49] H. Roy, M. Pocock, C. Preston, D. Roy, J. Savage, J. Tweddle, and L. Robinson, "Understanding citizen science & environmental monitoring. final report on behalf of uk-eof. nerc centre for ecology & hydrology and natural history museum," *Natural History Museum, London, UK. See http://www. ukeof. org. uk/co_ citizen. aspx (accessed 28/11/2012)*, 2012.

[50] Z. Orémuš, K. A. Hassan, J. Chmelík, M. Kňažková, J. Byška, R. G. Raidou, and B. Kozlíková, "PINGU principles of interactive navigation for geospatial understanding," in *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 216–225, 2020.

[51] T.-c. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[52] M. C. Eppes, B. Magi, B. Hallet, E. Delmelle, P. Mackenzie-Helnwein, K. Warren, and S. Swami, "Deciphering the role of solar-induced thermal stresses in rock weathering," *Bulletin*, vol. 128, no. 9-10, pp. 1315–1338, 2016.

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

[54] M.-C. Eppes and R. Keanini, "Mechanical weathering and rock erosion by climate-dependent subcritical cracking," *Reviews of Geophysics*, vol. 55, no. 2, pp. 470–508, 2017.

[55] M. Eppes, K. Warren, E. Hinson, and L. Dash, "Long term monitoring of rock surface temperature and rock cracking in temperate and desert climates," *AGUFM*, vol. 2012, pp. EP41F–0848, 2012.

[56] "Tableau." https://www.tableau.com/.

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

[58] S. Tilkov and S. Vinoski, "Node.js: Using javascript to build high-performance network programs," *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, 2010.

[59] R. Arias-Hernandez, L. T. Kaastra, T. M. Green, and B. Fisher, "Pair analytics: Capturing reasoning processes in collaborative visual analytics," in *2011 44th Hawaii international conference on system sciences*, pp. 1–10, 2011.

[60] P. Sajadi, A. Singh, Y.-F. Sang, S. Mukherjee, and K. Chapi, "Assessing the key drivers of stream network configuration dynamics for tectonically active drainage basins using multitemporal satellite imagery and statistical analyses," *Geocarto International*, pp. 1–32, 2021.

[61] D. Baur, F. Seiffert, M. Sedlmair, and S. Boring, "The streams of our lives: Visualizing listening histories in context," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1119–1128, 2010.

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

[63] N. Magdy, M. A. Sakr, T. Mostafa, and K. El-Bahnasy, "Review on trajectory similarity measures," in *2015 IEEE seventh international conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 613–619, IEEE, 2015.

[64] Y. Chen, M. A. Nascimento, B. C. Ooi, and A. K. Tung, “Spade: On shape-based pattern detection in streaming time series,” in *2007 IEEE 23rd International Conference on Data Engineering*, pp. 786–795, IEEE, 2007.

[65] T. Nakamura, K. Taki, H. Nomiya, K. Seki, and K. Uehara, “A shape-based similarity measure for time series data with ensemble learning,” *Pattern Analysis and Applications*, vol. 16, no. 4, pp. 535–548, 2013.

[66] H. Alt, “The computational geometry of comparing shapes,” in *Efficient Algorithms* (S. Albers, H. Alt, and S. Näher, eds.), pp. 235–248, Springer, 2009.

[67] J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner, “Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets,” in *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714)*, pp. 105–112, IEEE, 2003.

[68] M. O. Ward, “A taxonomy of glyph placement strategies for multidimensional data visualization,” *Information Visualization*, vol. 1, no. 3-4, pp. 194–210, 2002.

[69] B. Ravaji, V. Alí-Lagoa, M. Delbo, and J. W. Wilkerson, “Unraveling the mechanics of thermal stress weathering: Rate-effects, size-effects, and scaling laws,” *Journal of Geophysical Research: Planets*, vol. 124, no. 12, pp. 3304–3328, 2019.

[70] C. Bunce, D. Cruden, and N. Morgenstern, “Assessment of the hazard from rock fall on a highway,” *Canadian Geotechnical Journal*, vol. 34, no. 3, pp. 344–356, 1997.

[71] C. Brawner and D. Wyllie, “Rock slope stability on railway projects,” *Area Bulletin*, vol. 77, no. Bulletin 656, 1976.

[72] P. Budetta and A. Santo, “Morphostructural evolution and related kinematics of rockfalls in campania (southern italy): A case study,” *Engineering Geology*, vol. 36, no. 3-4, pp. 197–210, 1994.

[73] A. M. Ritchie, “Evaluation of rockfall and its control,” *Highway research record*, no. 17, 1963.