



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학박사학위논문

Multi-scale Visualization Design for Interactively Analyzing Large Time-series Data

대용량 시계열 자료의 대화형 분석을 위한 다중스케일
가시화

2014년 2월

서울대학교 대학원

전기컴퓨터공학부

조 명 수

Multi-scale Visualization Design for Interactively
Analyzing Large Time-series Data

대용량 시계열 자료의 대화형 분석을 위한
다중스케일 가시화

지도교수 서 진 욱

이 논문을 박사학위논문으로 제출함

2013 년 11 월

서울대학교 대학원

전기컴퓨터공학부

조 명 수

조명수의 박사학위논문을 인준함

2013 년 12 월

위 원 장 : _____ 신 영 길 _____ (인)

부위원장 : _____ 서 진 욱 _____ (인)

위 원 : _____ 김 명 수 _____ (인)

위 원 : _____ 김 보 형 _____ (인)

위 원 : _____ 이 정 진 _____ (인)

Abstract

Multi-scale Visualization Design for Interactively Analyzing Large Time-series Data

Myoungsu Cho

School of Computer Science Engineering

College of Engineering

The Graduate School

Seoul National University

We propose a unified visualization model, called a ripple graph, that takes the benefits of both of the bar graph and line graph with enhanced graphical integrity for not only the regularly measured but also irregularly measured time-series data. The ripple graph also unveils uncertainty of values between two temporal measurements by varying color intensity depending upon the confidence of the values. In doing so, it can effectively reveal the measurement frequency or interval while still showing the overall temporal pattern of change. We further extend the ripple graph representation into a single unified multi-scale visualization model via an interactive 2D widget to

accommodate the advantages of other efficient time-series data visualization techniques while addressing the scalability issue. We have conducted a controlled user study to show the efficacy of the ripple graph by comparing it to existing representations (i.e. line graph, bar graph, and interactive horizon graph), after selecting representative tasks (i.e. Max, Same, Frequency, and Confidence task) for time-series data visualization. Results show that ripple graph is overall the best performing in terms of task time, correctness, and subjective satisfaction across all task types.

Following a participatory design process with neurologists, we design an interactive visual exploration tool for time-series data, called Stroscope, based on the ripple graph representation and the widget. Stroscope provides various interactions to navigate data in temporal aspect and supports algorithmic time-series analysis methods to accomplish certain analytical tasks. We have also performed long-term case studies with two neurologists dealing with blood pressure measurements for 1600 stroke patients to show the effectiveness of Stroscope. They have could visually explore individual blood pressure values and their changes over time while maintaining the context, which could lead to save time and effort on exploratory analyses in comparison with using conventional statistical tools. In analyzing blood pressure data, Stroscope enables them to (1) find patients with anomalous patterns, (2) compare between two groups in terms of measurement values, measurement frequency, and fluctuation, (3) confirm what they already knew, and (4) formulate a

new hypothesis.

keywords : Irregularly measured time-series data, Frequency-aware visualization, Uncertainty visualization, Controlled user study, Long-term case study

student number : 2008-30243

Contents

Abstract	i
Contents	v
List of Figures	viii
List of Tables	xiii
Chapter 1 Introduction	1
1.1 Background & Motivation	1
1.2 Main Contribution	6
1.3 Organization of the Dissertation	9
Chapter 2 Related Work	11
2.1 (Large) Time-series data visualization	11
2.2 Event sequences data visualization	16
2.3 Interaction	17
2.4 Evaluation	19
Chapter 3 Problem Analysis	21

3.1	Dataset	22
3.2	A Scenario – Status Quo	24
3.3	Design Process	25
3.4	Design Rationale.	26
Chapter 4 Ripple Graph: A Multi-scale Visualization Model for time-series data		29
4.1	Visual Representation	30
4.2	Multi-scale Modeling	32
4.2.1	Dimension zooming with range of interest (ROI) . . .	32
4.2.2	Color mapping to further distinguish bars	33
4.2.3	Moving the horizontal axis	34
4.3	Visualizing degree of certainty between measurements . . .	35
4.4	User interface for ripple graph manipulation	37
4.4.1	Control panel	37
4.4.2	Focus lens	39
Chapter 5 Usability Study		43
5.1	Participants and materials.	43
5.2	Tasks.	44
5.3	Procedure	46
5.4	Results	48
5.5	Discussion	50
Chapter 6 Controlled User Study		55
6.1	Participants and materials.	55

6.2	Visualization techniques	56
6.3	Tasks	57
6.4	Study design and procedure	58
6.5	Results	60
6.6	Discussion	68
Chapter 7	Stroscope	69
7.1	Layout	69
7.2	User Interaction	71
7.3	Analytical Features	74
7.4	Implementation	78
Chapter 8	Case Study	79
8.1	Procedure	79
8.2	Participant 1 (P1)	80
8.3	Participant 2 (P2)	85
8.4	Discussion	89
Chapter 9	Conclusion	93
	Bibliography	95
	Abstract in Korean	105

List of Figures

- Figure 1.1 Visualization tools for large time-series data. It is difficult to show the measurement frequency or degree of irregularity in measurement intervals. 3
- Figure 1.2 Concept of Ripple Graph. Ripple graph (c) takes the benefits of both line graph (a) and bar graph (b). 6
- Figure 1.3 Ripple graph seamlessly integrates different existing time-series representations (a)-(f) into a unified interface framework for supporting different tasks for exploratory data analysis. All graphs revealing the measurement frequency are representations made by using a control panel for ripple graph manipulation. 6
- Figure 1.4 Overview of Stroscope. (left) control panel for ripple graph manipulation, (center) timeline view for representing multiple time-series data, and (right) detail view showing the information of the selected series. 8
- Figure 2.1 Visualization tools for a (few) long time-series data. These are designed for handling a few long time-series, but not for showing an intuitive over-view of multiple time-series. 13
- Figure 2.2 Visualization tools for large time-series data. It is difficult to show the measurement frequency because connected lines or filled areas

made it difficult to tell if measurement points are real or not. 15

- Figure 4.1 Concept of Ripple Graph. Ripple graph (c) takes the benefits of both line graph (a) and bar graph (b) while revealing the uncertainty of values between two adjacent real measurements: It effectively shows the overall temporal trend and the measurement frequency/interval while revealing the uncertainty of values between real measurements. 30
- Figure 4.2 Dimension zooming and color mapping in ripple graph. The ripple graph represents a time-series with a sequence of 12 values ranging from min to max for the given height h . (a) General ripple graph. (b) Ripple graph after applying ROI parameters: Bars are more distinguishable from each other than bars in (a). (c) Ripple graph after assigning colors to measurement values: A color strip on the right is a user-defined color table. 31
- Figure 4.3 Moving the horizontal axis. The ripple graph represents the time-series used in Figure 4.2. The horizontal axis is located at the bottom. (a) Bars with the values less than ROI_{low} are filtered out. (b) All bars have the same height when ROI_{low} and ROI_{high} are set to the minimum of the measurement value. 34
- Figure 4.4 Ripple graph with confidence interval of (a) 3-hour and (b) 6-hour. The ripple graph fills the area between bars with changing the alpha value.. . . . 36
- Figure 4.5 Control panel for ripple graph manipulation. Controls for (a) connecting bars and flipping bars downward around the horizontal time axis, (b) adjusting confidence interval, (c) adjusting ROI values, (d) showing a preview and changing a position of the time axis, (e) selecting a pre-defined color palette, and (f) manipulating color-related parameters.. . . . 38
- Figure 4.6 Ripple graph with the focus lens. (a) A user has a difficulty in

seeing overall temporal pattern of change for the time-series within an orange rectangle. (b) After designating the corresponding time-series using the focus lens, the user can see the overall temporal pattern of the time-series while making the best use of the given space. 40

Figure 5.1 Usability testing software, showing the task in the Usability category. Participants had to make a graph (lower graph) the same as the given ripple graph (upper graph). 47

Figure 5.2 Participant’s Strategies for Question T8 “What are the dates in which the rainfall exceeds 1000mm?” (a) Maximization of the height of the bars for measurements greater than 1000 (b) Differentiation by the horizontal axis with measurement 1000 (c) Assigning a red color to measurements greater than 1000. (d) Minimization of the height of the bars for measurements less than 1000; This strategy caused a problem that the bars with values slightly greater than 1000 are indistinguishable from minimized bars. 51

Figure 5.3 Participant’s Strategies for Question T13 “Find periods in which the price of Apple stocks is over \$620.” (a) Minimization in the height of the bars for measurements less than 620 (b) Differentiation by the horizontal axis with measurement 620. 52

Figure 6.1 Visualization techniques for time-series. (a) filled line graph. (b) bar graph. (c) horizon graph by default. (d) horizon graph after pan and zoom interaction 57

Figure 6.2 (a) Task time, (b) Correctness, and (c) Difficulty for each visualization technique by task type. 60

Figure 6.3 Task time for each visualization technique by number of time-series (i.e. 20 and 40). * indicates a significant difference in task time between 20 and 40 time-series ($p<.05$). 64

Figure 6.4	Average of subjective confidence results for each visualization technique by time point type	65
Figure 7.1	Overview of Stroscope.	70
Figure 8.1	Stroscope showing only SBP increasing periods. The yellow mark indicates when an SHT onset event occurred.	82
Figure 8.2	Stroscope showing only SBP decreasing periods. After selecting Δ SBP (a difference between consecutive SBP values) as a measurement variable, P1 can observe Δ SBP values less than -20. The yellow mark indicates when an SHT onset event occurred.	82
Figure 8.3	Stroscope showing a “SBP-SBPMean” variable. A black arrow indicates a patient with an anomalous pattern. Dark blue represents a sudden decrease in SBP (a great negative value of “SBP-SBPMean”). The yellow mark is the indicator of SHT onset event.	84
Figure 8.4	Stroscope showing the detail of the patient with anomalous pattern in Figure 8.3. P1 identified that the “SBP-SBPMean” value decreased approximately from 25 to -50 about 7 hours before the SHT onset. The yellow mark is the indicator of SHT onset event.	84
Figure 8.5	MRI scans before (left) and after (right) the treatment to remove the offending thrombus.	85
Figure 8.6	Stroscope showing values under 85 in blue color. An orange arrow indicates a patient with an anomalous pattern because low values (i.e. values under 85) are observed frequently for only the patient.	86
Figure 8.7	Stroscope that all values are represented by the same height and color. An orange arrow indicates a patient with an anomalous pattern because measurement frequency is higher for the patient	

	than others.	87
Figure 8.8	Stroscope showing the fluctuation around the value of 110. An orange arrow indicates a patient with an anomalous pattern because of a sharp fluctuation.	87
Figure 8.9	Stroscope showing hierarchical grouping result. P2 aligned patients by onset time of stroke and grouped patients according to Cluster and END variables after performing a clustering function. Black rectangular spots in the timeline view represent the END_progression event. In the two clusters with low blood pressure (the first and last clusters), the event tends to occur within about 30 hours from onset.	88

List of Tables

Table 3.1	Four Clinical Variables (selected). There are 29 clinical variables in the 1600 acute ischemic stroke patients' data that we used for the design and development of Stroscope. END (Early Neurological Deterioration) stands for neurological worsening within hospitalization period from stroke onset.	23
Table 5.1	Tasks of Utility Category. Elementary tasks to focus on an individual value for a time-series and Synoptic tasks to focus on the overall trend or relation with other series.	45
Table 5.2	Number of participants who failed and average completion time for each task.	49
Table 6.1	Results of post-hoc test by each task type. The < sign represents the inequality relation with a statistical difference ($p < .05$). . . .	61
Table 6.2	Results of Post-hoc test by each task type for the task time. * means that there is a significant difference between the mean task times of two techniques.	62
Table 6.3	Results of Post-hoc test by each task type for the correctness. * means that there is a significant difference between the mean correctness of two techniques.. . . .	63
Table 6.4	Results of Bonferroni correction by each task type for the difficulty.	67

Chapter 1

Introduction

1.1 Background & Motivation

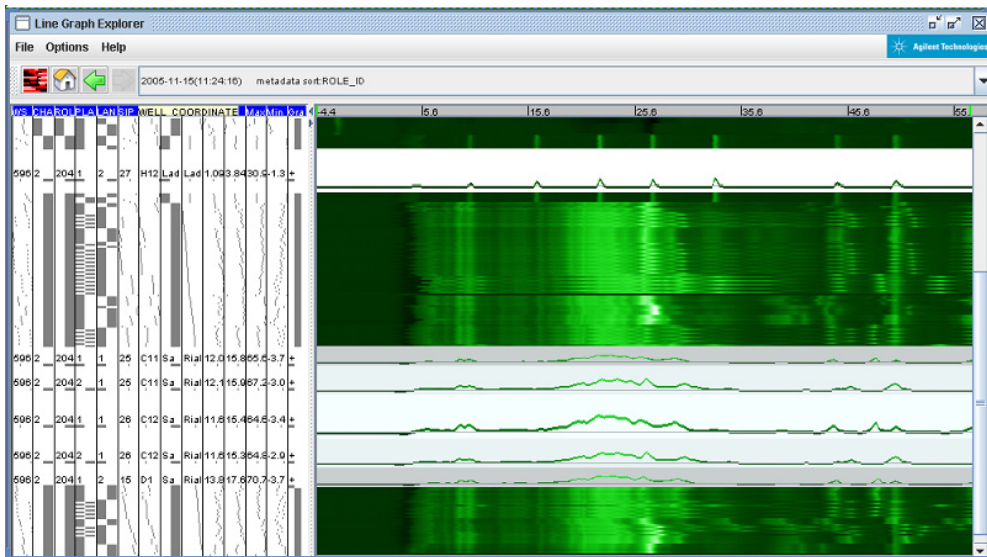
A time-series data is a sequence of data points, measured at successive points in time. It is common across various domains such as finance, science, business, biographies, history, planning, and medicine. Domain experts want to predict the future trends by exploring and understanding time-series data. Because data points of time-series are inherently linked to time, time-series data need to be considered differently than other kinds of data and require appropriate visual and analytical methods to investigate temporal aspects of underlying problems [5].

Well-known time-series data visualization, the line graph, is effective in revealing overall temporal trend of a time-series; however, it is not accurate in showing the measurement frequency or interval. Moreover, in some sense, it harms the graphical integrity [52] because the connected lines lead to false confidence in values between measurements, especially for irregularly measured time-series data.

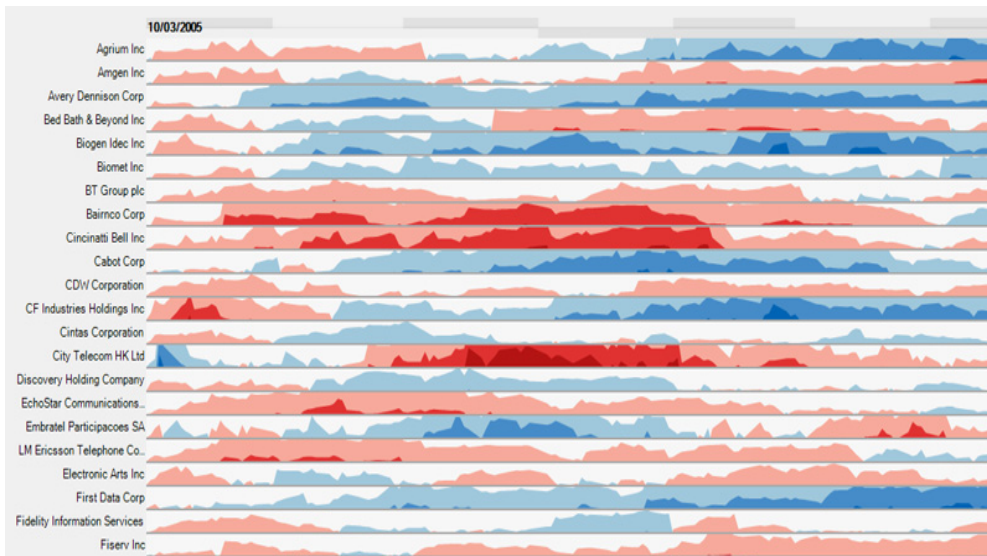
In contrast, the bar graph is effective in showing frequency/interval without interpolating values between measurements, but it is less efficient in showing an overall pattern than the line graph.

In information visualization domain, growing needs of finding important patterns and trends in time-series data have spurred the development of many interactive visual exploration tools: Line Graph Explorer [31], LiveRAC [35], SignalLens [30], and Data Vases [53] to name a few (Figure 1.1). As the size and complexity of time-series data increase, visualization researchers now face new challenges and requirements for design of the interactive visual exploration tools. In this dissertation, we identify some of such challenges and requirements and address them in a new interactive visualization tool.

Most existing time-series data visualization techniques or tools assume that (1) each time-series data is measured regularly over time and (2) all time-series data have the same measurement range. However, there are often the cases that the time-series data do not meet the conditions, e.g. online auction data, regional rainfall/snowfall data, and credit card usage data. For these data, existing visualization tools do not help us much in answering questions regarding frequency-related patterns or trends: For example, (a) which item has the greatest number of bids? (b) in which city does it seldom rain in summer? (c) what is the difference in monthly credit card usage pattern in terms of the amount and frequency?



(a) Line Graph Explorer [31] provides an interactive interfaces as well as overview for multiple time-series using Focus+Context technique.



(b) Horizon Graph [40] is a space-efficient technique by increasing the density of time series graphs by dividing and layering filled line charts

Figure 1.1. Visualization tools for large time-series data. It is difficult to show the measurement frequency or degree of irregularity in measurement intervals.

Not much effort has been put into developing visualization models or tools for such datasets in the infovis community. Aris et al. [9] called these data unevenly-spaced time-series data and suggested four representations (i.e. Sampled Events, Aggregated Sampled Event, Event Index, and Interleaved Event Index) for the interactive visual exploration of such data. The Sampled Events method and the Aggregated Sampled Event method generate an evenly spaced time-series data by sampling at a specific regular interval. The Event Index method distorts the time axis to highlight the number of events. The Interleaved Event Index method represents the sequence of events while preserving temporal order of events regardless of their real time interval. However, the first two methods can cause the data loss that can come from sampling and aggregation and the last two methods can distort the time axis by arbitrarily changing the time intervals between two consecutive events.

All of them basically regularize the measurement interval, which could lead to loss of information such as measurement frequency or interval which is likely crucial for many tasks. To address this problem, a new visual representation for irregularly measured time-series data is required to reveal the measurement frequency while maintaining graphical integrity.

These kinds of data are fairly abundant in the medical domain as well. The examples include, but are not limited to, body temperature, blood sugar level, and blood pressure level in patient records, where the number/interval of measurements and hospitalization time can vary depending on patients' condition. Researchers in

the medical domain often have to arduously collect these data to formulate and test hypotheses. Although they may rely on conventional statistical software or Excel for that matter, exploratory analyses for hypothesis formulation are not easy, not to mention that it is neither intuitive to use nor easy to understand the results. As the number of records and number of measurements each record are increased, researchers spend more time and effort on exploratory analyses. An interactive time-series data, especially large and irregularly measured time-series data, visualization tool is required to facilitate exploratory data analysis.

The visualization tool of large time-series involves effective interaction techniques to support the dialogue between users and data [54], which can lead to focusing on different parts of the data. For time-series data in distinction from other kinds of data, users need to interact with time such as aligning with a specific time, sorting by duration, or navigating in time.

In summary, the following two problems are identified in visually exploring and analyzing large time-series data. First, most existing time-series data visualization techniques or tools cannot reveal the measurement frequency or interval although irregularly measured time-series data are abundant in various domains. Second, in the medical domain, there is no visualization tool for exploratory data analysis of large time-series data while supporting interaction to navigate data.

1.2 Main Contribution

First, we propose a unified visualization model, called a ripple graph, that takes the benefits of both of the bar graph and line graph with enhanced graphical integrity for not only the regularly measured but also irregularly measured time-series data (Figure 1.2). The ripple graph also unveils uncertainty of values between two temporal

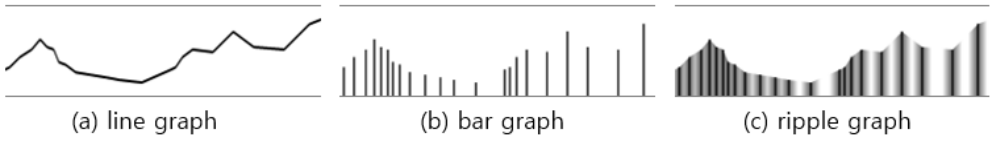


Figure 1.2. Concept of Ripple Graph. Ripple graph (c) takes the benefits of both line graph (a) and bar graph (b).

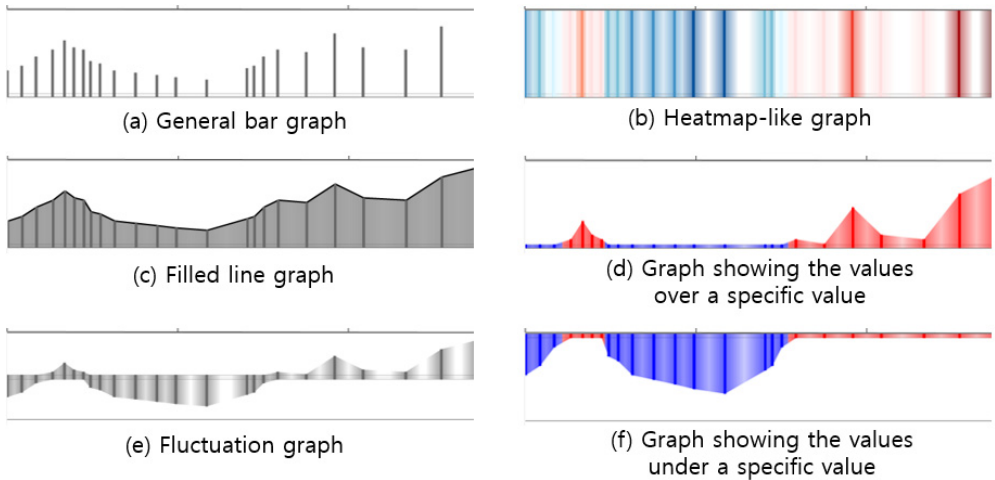


Figure 1.3. Ripple graph seamlessly integrates different existing time-series representations (a)-(f) into a unified interface framework for supporting different tasks for exploratory data analysis. All graphs revealing the measurement frequency are representations made by using a control panel for ripple graph manipulation.

measurements by varying color intensity depending upon the confidence of the values.

In doing so, it can effectively reveal the measurement frequency or interval while still showing the overall temporal pattern of change. We further extend the ripple graph representation into a single unified multi-scale visualization model via an interactive 2D widget to accommodate the advantages of other efficient time-series data visualization techniques while addressing the scalability issue (Figure 1.3). To verify the usefulness of ripple graph, we have conducted a usability study from usability and utility aspects. We have also conducted a controlled user study to show the efficacy of the ripple graph by comparing it to existing representations (i.e. line graph, bar graph, and interactive horizon graph), after selecting representative tasks (i.e. Max, Same, Frequency, and Confidence task) for time-series data visualization. Results show that ripple graph is overall the best performing in terms of task time, correctness, and subjective satisfaction across all task types.

Second, we design an interactive visual exploration tool for time-series data, called Stroscope (Figure 1.4), based on the ripple graph representation and the widget, following a participatory design process with neurologists. Stroscope provides various interactions to navigate data in temporal aspect. Stroscope supports algorithmic time-series analysis methods and on-demand on-the-spot visualization of statistical summary measures according as tightly integrating statistics and visualization improves exploratory data analysis, enabling users to generate significant discoveries [38].

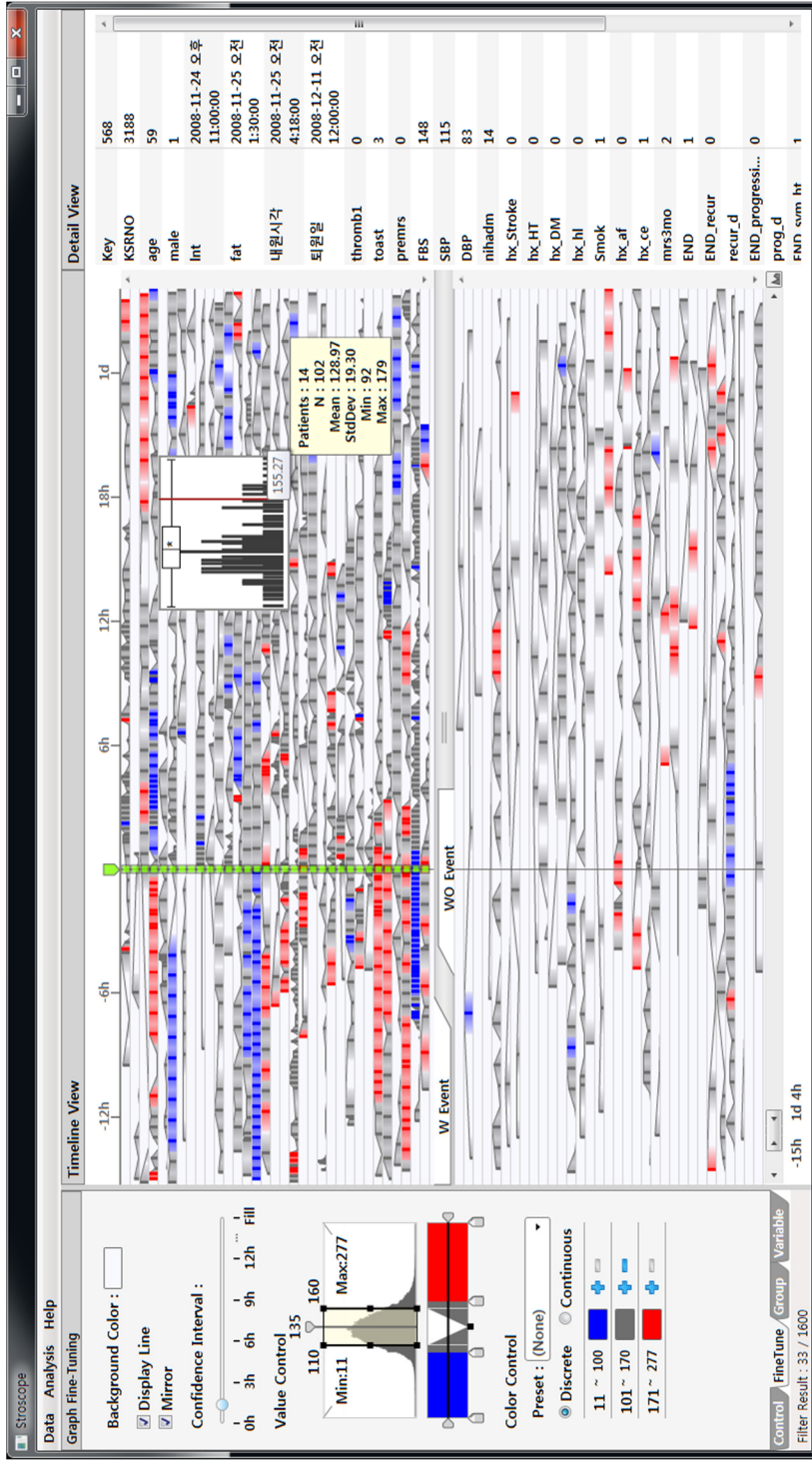


Figure 1.4. Overview of Stroscope. (left) control panel for ripple graph manipulation, (center) timeline view for representing multiple time-series data, and (right) detail view showing the information of the selected series.

We have performed long-term case studies following the Multi-Dimensional In-Depth Long-Term Case Studies protocol [51] to show the effectiveness of Stroscope with two neurologists dealing with time-series data of blood pressure measurements for 1600 acute ischemic stroke patients in their workplaces. They have could visually explore individual blood pressure values and their changes over time while maintaining the context, which could lead to save time and effort on exploratory analyses in comparison with using conventional statistical tools. In analyzing blood pressure data for stroke study, Stroscope enables them to (1) find patients with anomalous patterns, (2) compare between two groups in terms of measurement values, measurement frequency, and fluctuation, (3) confirm what they already knew, and (4) formulate a new hypothesis.

1.3 Organization of the Dissertation

This dissertation is organized as follows. In Chapter 2, we will describe the related works in time-series visualization, evaluation studies on graphical perception of visual representations for time-series data, and dynamic query interface. In Chapter 3, we will describe the design rationale along with real world data and a user scenario. Then, we explain the ripple graph and present the controlled user study results. After we introduce the visualization tool – Stroscope, we present the long term case study results. And then, we close this dissertation with plans for future work and conclusions in Chapter 9.

Chapter 2

Related Work

In this section, we review previous work on representation techniques and interactive visualization tools for time-series data. We deal with temporal event sequence visualization separately, since it has different design requirements than general time-series data visualization. We also review previous work on interaction and evaluation of time-series data visualizations.

2.1 (Large) Time-series data visualization

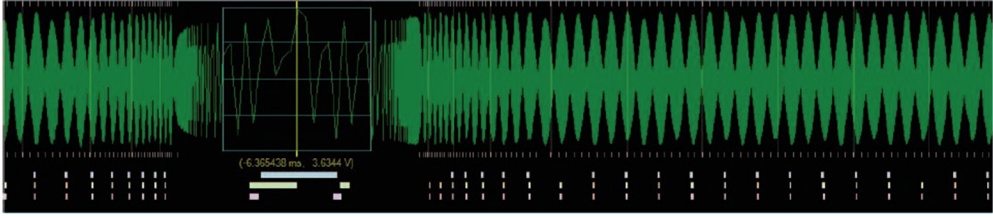
The goals of analyzing time-series are to grasp the evolution of data over time and detect trends and patterns for gaining insights and understanding data [4]. The most often used visual representations for time-series data are line graph [52], which encode time along the horizontal axis and encode time-varying values along the vertical axis. Users are understands easily and no learning is required because of their

simplicity and well-known form.

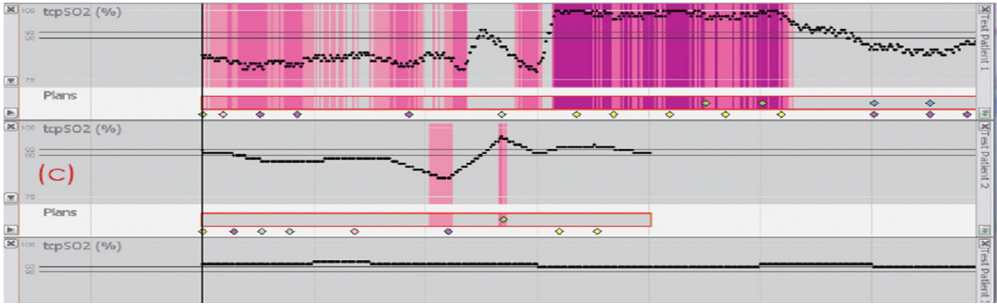
Line graph is effective in revealing overall temporal trend rather than individual values because it connects data points. As the size and complexity of time-series data increase, many representation techniques or visualization tools were developed.

VizTree [34] allows user to find anomalies by converting a time-series into a symbol string and visualizing the symbol as a tree. The flexible multi-foci navigation techniques were proposed in KronoMiner [68], TraXplorer [27] and SignalLens [30]. BinX [14] supports different aggregations on time dimension according to the abstraction level defined by a user. ChronoViz [18] is a visualization and analysis tool for time-based data from multiple sources. Visual exploration tools for patient data were proposed in [16], [23] and [43]. These tools are designed for handling a few long time-series, but not for showing an intuitive overview of multiple time-series (Figure 2.1).

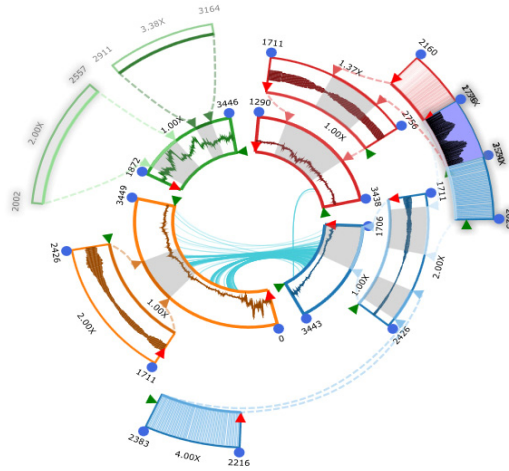
Many representation techniques or interactive tools for large amounts of time-series have been developed. Based on the two-tone pseudo coloring by Saito et al. [44], a more space-efficient visualization technique called a horizon graph [40] was developed using dividing, mirroring, and layering techniques. However, our controlled user study showed that this technique based on filled line chart is not suitable for frequency-related task.



(a) SignalLens [30] shows an amplitude modulated electronic signal.



(b) CareCruiser [23]. The time-oriented view of CareCruiser focuses on the temporal-qualities of applied treatment plans, clinical actions.

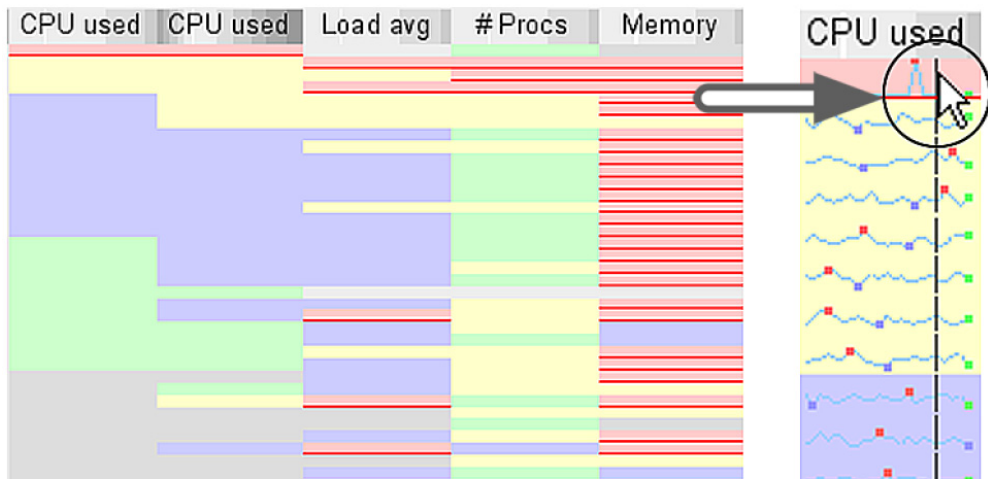


(c) KronoMiner [68] allows a user to explore four time-series

Figure 2.1. Visualization tools for a (few) long time-series data. These are designed for handling a few long time-series, but not for showing an intuitive over-view of multiple time-series.

Line Graph Explorer [31] and LiveRAC [35] provide interactive interfaces for exploratory analysis as well as overview for multiple time-series using a Focus+Context technique. ATLAS [17] visualizes massive temporal data using line graphs after pre-fetching data from remote database servers. Thakur et al. [53] suggested a two-dimensional representation using a symmetric glyph, called a kite diagram and presented Data Vases to compactly display multiple time-series. In [11], several representation techniques, each of which is efficient in revealing a different level-of-details were introduced using medical data as examples. All these techniques and visualization tools were designed assuming regularly measured time-series. Thus, it is difficult for them to show the measurement frequency or degree of irregularity in measurement intervals (Figure 1.1 and Figure 2.2).

Aris et al. [9] suggested four representations for unevenly spaced time-series data. The Sampled Events method and the Aggregated Sampled Event method generate an evenly spaced time-series data by sampling at a specific regular interval. The Event Index method distorts the time axis to highlight the number of events. The Interleaved Event Index method represents the sequence of events while preserving temporal order of events regardless of their real time interval. However, the first two methods can cause the data loss that can come from sampling and aggregation and the last two methods can distort the time axis by arbitrarily changing the time intervals between two consecutive events.



(a) LiveRAC [35] shows multiple time-series using a reorderable matrix of area-aware charts. (left) an overview of 50 time-series. (right) detail of time-series as sparkline charts



(b) A visual comparison of the financial market for all assets in 3 countries and 28 market sectors [69].

Figure 2.2. Visualization tools for large time-series data. It is difficult to show the measurement frequency because connected lines or filled areas made it difficult to tell if measurement points are real or not.

TimeRider [42] also deals with irregularly sampled data and reveals temporal aspects using animation in an animated scatter plot. In this tool, however, it is hard to see an overview because only one time frame can be seen at a time. Our tool supports an intuitive overview while maintaining the graphical integrity for irregularly sampled data.

2.2 Event sequences data visualization

Temporal event sequences in the data such as electronic health records, highway incident logs or web logs, can be thought of as a kind of time-series, where each event does not have a quantitative property but a categorical one with a timestamp. The irregularly measured time-series data handled in this dissertation covers this event sequence data. There have been visualization tools to help users discover frequent or anomalous patterns in these temporal categorical event sequence data.

VISUAL-TimePACTS [56] shows activities of individuals during a day using a space-time path. ActiviTree [57] visualizes a single event sequence using a circular tree-like representation and Continuum [7] visualizes a hierarchical relationship between temporal events. Lifelines2 provides visual temporal summaries to compare trends across multiple records [59] and an alignment operation that aligns sequences by a specific event [60]. Wongsuphasawat et al. [63] proposed a temporal categorical similarity measure, called a “Match & Mismatch”, to find similar records. LifeFlow [64] visualizes all possible patterns of event sequences through aggregation while maintaining temporal interval between events. CloudLines [32] represents each event

as a circle with the size and opacity varied by importance of the event. These tools represent multiple event sequences by stacking and interpolating them vertically after aligning each sequence by a time attribute. We took these approaches, except for interpolation, in our tool to show an overview and compare trends across multiple time-series.

2.3 Interaction

Brushing & linking is a well-known interaction, which means the connection of two or more views of the same data, such that a change to the representation in one view affects the representation in the other [12]. Hauser et al. introduced brushing based on angles between data items [24].

Dynamic queries are a well-known successful approach to information seeking that can allow users to deal with information overload by interactively formulating query and rapidly showing the results [3][48]. Many information visualization research and commercial systems adopted dynamic queries as an essential component.

Dynamic HomeFinder [62] was one of the first example systems that implemented dynamic queries. It helps users explore a real-estate database and find satisfying homes. Dynamap [41] also implemented traditional dynamic queries to help users find trends in Census data overlaid on a 2D map. PhotoFinder allows users to specify multiple disjunctive range selections using multiple two-sided slider bars to show a histogram chart for each attribute [29]. Filmfinder [50] supports users to search for certain films in large film database using various dynamic query devices.

Spotfire [1] is the first commercially successful information visualization tool that is based on traditional dynamic queries. These applications and tools implement traditional dynamic query widgets and support only conjunct or disjuncts [49]. Spatial proximity queries need dynamic queries that support disjunct or conjuncts.

Various interface widgets such as range sliders, check boxes, and buttons are used to formulate traditional dynamic queries. There also have been a few approaches to extend the expressive power of traditional dynamic queries. AlphaSlider [2][37] is a query device to allow users to select one item from an alphabetically arranged list without using a keyboard. Timebox widget of TimeSearcher [26] enables users to draw multiple rectangular boxes to specify conjunctive queries and find the time series data with a certain pattern. QuerySketch [58] allows users to draw a freehand line graph and find data that match the graph.

Young and Shneiderman also presented a novel water flowing model to support a complete set of Boolean expressions [67]. There also have been some approaches to allow users to perform more general Boolean queries using the dynamic query technique. Fishkin [20] presented a new dynamic query interface to support compound queries by overlapping multiple filters such as Magic Lens filters [15]. Users can formulate queries of various combinations such as disjunction of conjunctions as well as conjunction of disjunctions. This work is most similar to our approach, but it was not designed to support the formation of the spatial proximity query where the degree-of-interest varies (gradually increased/decreased) depending on the

distance to the query point.

2.4 Evaluation

There have been evaluation studies on graphical perception of visual representations for time-series data to recommend appropriate representations for different types of tasks. Javed et al. conducted a controlled user study to evaluate four different visualization techniques (simple line graph, braided graph, small multiples and horizon graph) for local/global tasks in terms of the graphical perception of multiple time-series [28]. This study showed that shared-space techniques are more efficient for local comparison tasks and separate-space techniques are more efficient for dispersed comparison tasks. Aigner et al. showed that an indexing technique - transforming scale of data to a comparable unit - was superior among different representation techniques in comparing two heterogeneous time-series data through a comparative study [6]. A study by Heer et al. [25] compared the performance time and accuracy between line graph and horizon graph. One of their results was that layered bands are more efficient for a small display space. They also proposed an optimal chart height for better graphical perception. These evaluation studies were concerned with line graph and its variants.

A task model was introduced by Andrienko and Andrienko [8] to specify the domain problems that users face in time-series data visualization and analyses. According to the model, there are two different classes of tasks: (1) Elementary tasks to focus on an individual value for a time-series data and (2) Synoptic tasks to focus

on the entire trend or relation with other data for multiple time-series data. We performed a usability study based on this model to evaluate the ripple graph.

Shneiderman et al. introduced “Multi-dimensional In-depth Long-term Case Studies (MILCs)” to evaluate information visualization tools in case studies with real users dealing with real datasets in their workplaces [51][48]. This is a qualitative evaluation methodology for evaluating the efficacy and utility of tool from diverse perspectives using a combination of methods such as interview, survey and observation. This evaluation method has been used in many studies [19], [21], [35], [46], [47], [61], [64], [65], and [66]. We also performed long-term case studies following the MILCs.

Arias-Hernandez introduced PairAnalytics [10], which is a method for capturing reasoning processes when users interact with visual representations of massive data. This method is an approach that a *subject matter expert* (SME) and a *visual analytics expert* (VAE) perform a given task together for real data and problems. In our case studies, we also employed a modified pair analytics method where an experimenter (i.e., SME) demonstrated our tool to participants (i.e., VAE) only when they asked for help as used in [33]. In this way, participants tried using our tool and quickly became familiar. Then we could improve our tool iteratively by removing roadblocks.

Chapter 3

Problem Analysis

One of the ultimate goals of medicine is to take care of the health and well-being of patients for their whole lifespan. To achieve this goal, it becomes necessary to keep track of individual health records throughout their entire lifespan, which makes it inevitable to deal with irregularly measured time-series data. Body temperature, blood sugar level, blood pressure level, and liver enzyme level are good examples. As an attempt to promote interactive information visualization techniques in the medical domain, a collaborative participatory project was launched with neurologists interested in analyzing relationship between the progression of stroke and the blood pressure change over time. In the following subsections, we explain the clinical research problems with the blood pressure data.

3.1 Dataset

A group of neurologists collected time-series data of blood pressure measurements for 1600 acute ischemic stroke patients at the Seoul National University Bun-dang Hospital in Korea. All patients in this dataset were hospitalized within 48 hours after the onset of stroke.

There are two sets of data. One includes stroke-related clinical information with 29 clinical variables such as age, gender, and medical history. Four important variables are summarized in Table 3.1. The other dataset includes systolic blood pressure (SBP) and diastolic blood pressure (DBP) values along with the time of measurements. This dataset is different from the usual time-series data handled in most conventional time-series data visualization tools. First, it is measured irregularly over time. Second, each patient has a unique hospitalization period, i.e. the total measurement period is different for each patient, ranging from 3 days to 60 days. Third, the first/last measurement time is different for each patient.

Table 3.1. Four Clinical Variables (selected). There are 29 clinical variables in the 1600 acute ischemic stroke patients' data that we used for the design and development of Stroscope. END (Early Neurological Deterioration) stands for neurological worsening within hospitalization period from stroke onset.

Variable	Value	Description
TOAST (Classification of types according to cause of stroke)	1	Cases in which the size of the lesion seen from the brain image is greater than 2cm
	2	Cases in which the size of the lesion seen from the brain image is less than 2cm
	3	Cases in which cerebral infarction occurs
	4	Cases in which it cannot be classified as remainder
	5	Cases in which there are more than 2 causes
	6	Cases in which causes are not identified after all necessary inspections
	7	Cases in which causes are not identified due to incomplete inspection
	8	Cases in which cerebral infarction occurs temporarily and disappears
END_Progression	1 or 0	Occurrence of symptom that gets worsening within 24 hours or not
END_Recur	1 or 0	Occurrence of symptom that gets worsening after 24 hours or not
END_Sym_ht	1 or 0	Case in which hemorrhage is observed in image, which is related to the END

3.2 A Scenario – Status Quo

At the beginning of our participatory design process, we observed how neurologists analyzed blood pressure data in their clinical practice. As a result, we came up with a primary persona, Dr. Lee - a neurologist with 20 years of experience, and a representative user scenario that explained the status quo of the data analysis process in the field.

Dr. Lee's goal is to examine if there are differences in blood pressure value and variability between the patients whose symptoms worsen within 24 hours from the onset of stroke and others. He needs to focus on the effect of the END_progression to achieve his goal. He first separated the patients into two groups: patients with END_progression value of 1 (group A) and other patients (group B). He consulted a statistician and decided to control two most important compounding variables – gender and age. They generated a new group B' by choosing the ones in the group B that matched the ones in the group A in terms of gender and age. They used the statistical software R for the matching task, and then they compared the two groups of A and B' in SPSS in terms of initial value, mean, standard deviation, maximum, and minimum of blood pressure. Although he wanted to see an overview of each group at once, descriptive statistics did not tell him an overall temporal trend in intuitive ways.

Carefully comparing the two groups, he formulated hypotheses. For example, he hypothesized that there would be a meaningful blood pressure change when an

END_progression occurs. To verify this hypothesis, he needed to examine how the measurements fluctuate around a clinically meaningful value and how the mean value changes every three- or six-hour period during 24 hours before and after END_progression occurs. These analyses were arduous and time consuming with conventional statistical tools because the numerous iterative filtering of patients and the quantization of time intervals were not efficiently supported in such tools. A bigger problem was that every patient's data in group B' was required to be aligned along a simulated event to be fairly compared with group A, where a simulated event could be defined as a virtual event occurring at the event time of the matched patient in group A.

3.3 Design Process

We have learned that the neurologists have never seen their data in a visualization tool. The fact that they could “see” the data in a more intuitive and informative way and interactively manipulate the data highly motivated them to participate in the design process.

We as information visualization designers collaborated with the real users – 3 neurologists to understand each other's work. We had met them at the hospital 6 times over a 6-month period. We alternated between observing users while they performed data analyses with their conventional tools and discussing what they did and why. We also showed them what is possible with interactive visualization tools to educate them about information visualization and to strengthen the partnership.

3.4 Design Rationale

As a result of our observations of and discussions with the real users, we have come up with the following design rationale of Stroscope using ripple graph representation.

(1) **Reveal measurement frequency/interval**

Blood pressure measurement frequency tells analysts a patient's condition, an occurrence of event, or a change in surroundings. Thus, we decided to design a new representation based on bar graph which is useful to grasp the measurement frequency.

(2) **Show fluctuation**

Neurologists observe changes in blood pressure around a clinically meaningful reference value possibly different for each patient, which can lead to finding patients with anomalous patterns. We decided to show the fluctuation of measurement values around a user-defined clinically meaningful value.

(3) **Use focus+context technique**

Systolic blood pressure values are concentrated in a narrow range around 120. It is necessary to enable users to interactively adjust their specific range of interest and see the detail while maintaining the context.

(4) **Keep familiarity**

We integrated well-known bar graph and line graph into a new visual representation to keep the learning curve steep for the new representation.

(5) **Provide unified interface**

Different representations are necessary for supporting different tasks for exploratory data analysis. We designed a unified interface framework into which we can seamlessly integrate multiple representations.

(6) Support multiple time-series

For visual exploration of multiple time-series, we have to support an intuitive overview and user interactions such as zooming, sorting, filtering, and grouping.

(7) Facilitate comparison task

One of the main tasks in the analytical exploration of the blood pressure data is to compare between two groups. To reduce bias and the effect of compounding variables in the comparison task, selecting well-matched entities from each group is important. Interactive matching and alignment of patients across two groups are important for accurate comparison.

(8) Integrate visualization and statistical methods

To facilitate exploratory data analysis, on-demand on-the-spot visualization of statistical summary measures is required. It enables users to perform a quick-and-dirty hypothesis testing on the spot.

Chapter 4

Ripple Graph: A Multi-scale Visualization Model for time-series data

We propose a multi-scale time-series data visualization model, “ripple graph,” to represent measurement frequency and uncertainty between measurement points as well as measurements of time-series data. We integrate line graph and bar graph into ripple graph (Figure 4.1c) to take the benefits of both, i.e. line graph (Figure 4.1a) for showing the overall temporal trend and bar graph (Figure 4.1b) for revealing measurement frequency. Furthermore, it also takes advantages of space-efficient representation techniques such as horizon graph [40] and heatmap-like graph [31] in a multi-scale model.

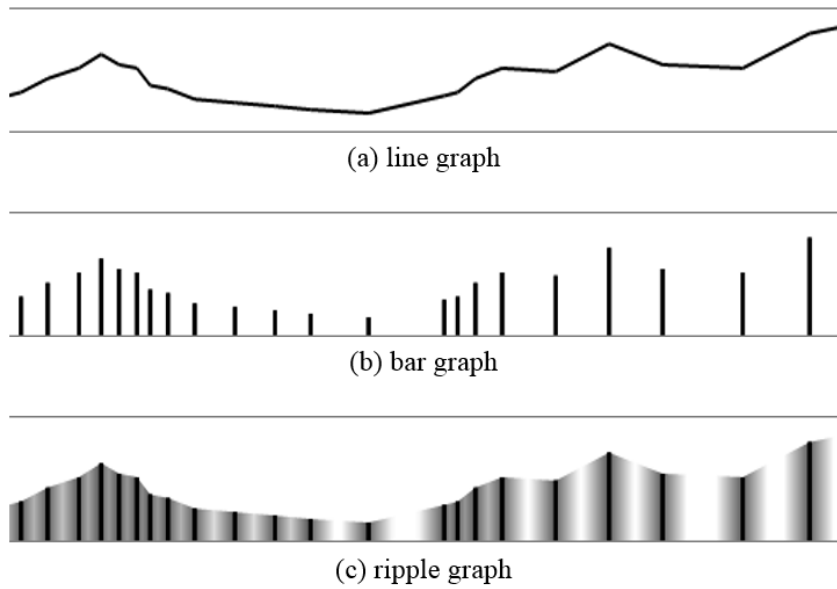


Figure 4.1. Concept of Ripple Graph. Ripple graph (c) takes the benefits of both line graph (a) and bar graph (b) while revealing the uncertainty of values between two adjacent real measurements: It effectively shows the overall temporal trend and the measurement frequency/interval while revealing the uncertainty of values between real measurements.

4.1 Visual Representation

The ripple graph aligns time points horizontally and measurements vertically (Figure 4.2a). The horizontal axis runs vertically in the middle of the given space by default. For example, in case of the blood pressure data, the horizontal temporal axis is positioned vertically at the midpoint of the blood pressure value range, and each bar anchored to the horizontal axis is displayed upward for the values over the midpoint value and downward for the values under the midpoint value.

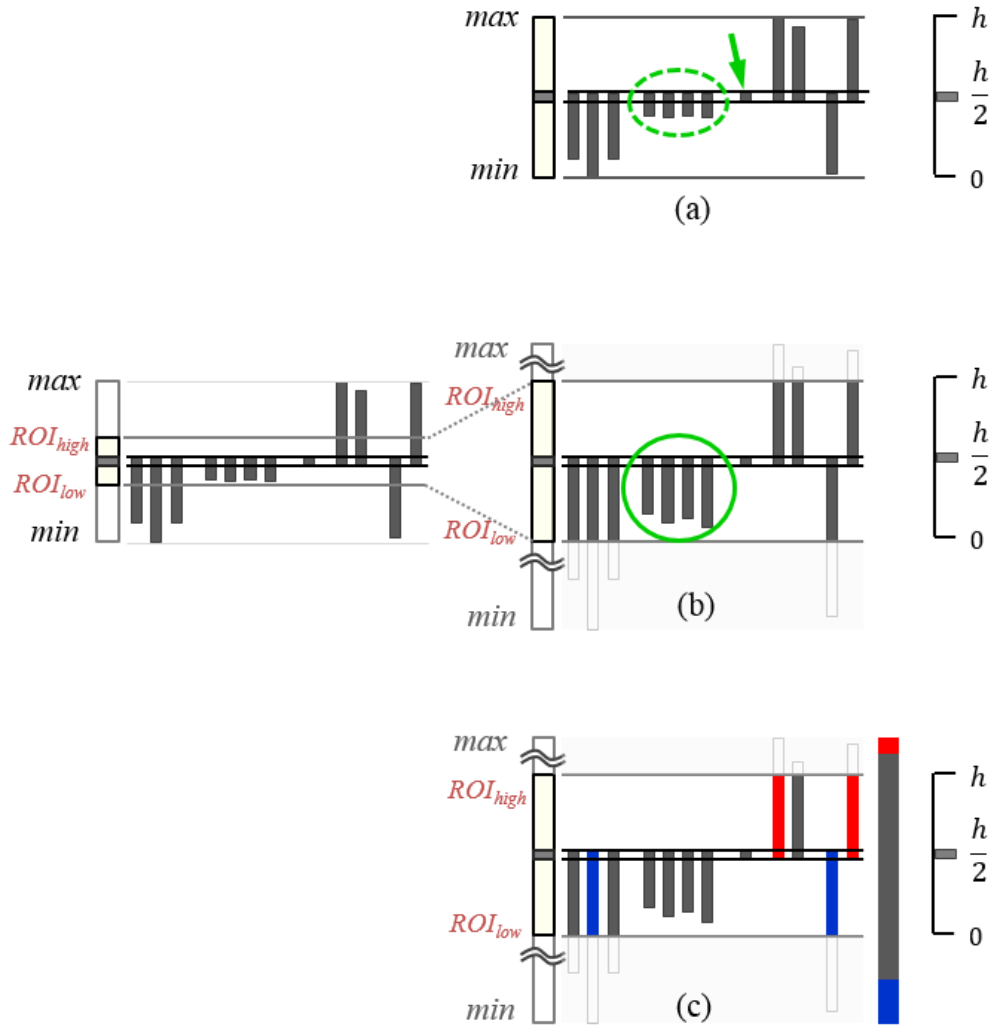


Figure 4.2. Dimension zooming and color mapping in ripple graph. The ripple graph represents a time-series with a sequence of 12 values ranging from \min to \max for the given height h . (a) General ripple graph. (b) Ripple graph after applying ROI parameters: Bars are more distinguishable from each other than bars in (a). (c) Ripple graph after assigning colors to measurement values: A color strip on the right is a user-defined color table.

In this way, it is easy to see how the blood pressure measurements fluctuate around a specific value which can be any user-defined meaningful value. The default position of the horizontal axis can be changed depending on the problem domain. For example, in case of stock data, the horizontal axis is better to be positioned at the bottom, i.e. at zero.

When one measurement is exactly equal to the user-defined value that the horizontal axis represents, the graph cannot show that measurement point because the height of its corresponding bar is zero, which gives a false impression to users that it was not even measured at the time point. To remedy this problem, we made the horizontal axis a tube-like dual line with the thickness of a small number of pixels. Then, the measurement point can be clearly shown as a blob in the tube as the eighth bar in Figure 4.2a (indicated by the green arrow).

4.2 Multi-scale Modeling

Quantitative comparison in a ripple graph becomes challenging when many graphs have to be shown on the screen. Then, the height of a ripple graph becomes too small to discriminate each value in the data. In order to resolve this problem, we developed three space-efficient methods in a multi-scale model.

4.2.1 Dimension zooming with range of interest (ROI)

The first method allows a user to zoom in to a specific range of values that the user

is interested in by adjusting the range of the vertical axis. It adjusts the height of a bar accordingly to the range of measurements of interest defined by the user, where the vertical axis spans from the lowest value (ROI_{low}) to the highest value (ROI_{high}) in the range. Then, the bars for any values out of the range have the same height, i.e. reach the top or the bottom of the given space. In this way, the user can zoom in to a specific range of interest to compare values in the range in detail, while maintaining the context, i.e. knowing the existence of the values outside the range. The four downward bars with similar height in Figure 4.2a (see the bars within the green dotted circle) can be more clearly distinguishable after setting ROI_{low} and ROI_{high} properly as shown in Figure 4.2b (see the bars within the green solid circle).

4.2.2 Color mapping to further distinguish bars

The second method allows a user to assign colors to measurements, through which each bar is painted in the corresponding color determined by a user-defined color table. This method was similarly used in [11] as “height-coded timelines,” but the two methods are different in that our method fills only the bars whereas the height-coded timelines fill the whole space including gaps between bars as well. Figure 4.2c shows that the first two bars with the same height are differentiated by color (gray for first bar and blue for second bar) and so are the third and fourth bars to the right end. The user-defined color mapping is shown in the vertical color strip on the right side of Figure 4.2c. To give users more flexible control over the visual encoding, the color mapping is independent of ROI as shown in Figure 4.2c.

4.2.3 Moving the horizontal axis

The last method allows a user to change the vertical location of the horizontal time axis. To see the bars with the values over a specific value in detail, a user can move the horizontal axis downwards. The horizontal axis can be even located at the bottom of the given space representing the value of ROI_{low} , and then the bars with the values less than ROI_{low} disappear. Figure 4.3a shows that the first three bars, which have values less than ROI_{low} , are filtered out. In this way, the vertically movable horizontal axis enables users to filter out some measurements.

When both ROI_{low} and ROI_{high} are set to the minimum in the whole measurements range and the horizontal time axis is located at the bottom (Figure 4.3b), all bars have the same height, i.e. the height of the given space for each series. The visualization then becomes a heatmap-like graph [31] where each measurement

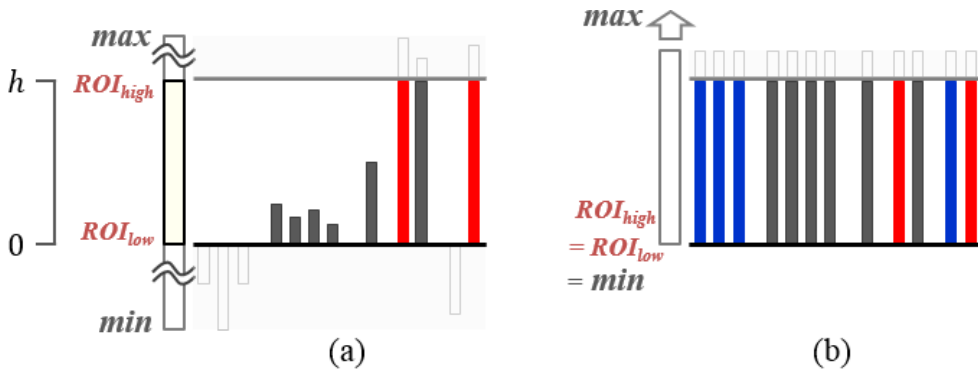


Figure 4.3. Moving the horizontal axis. The ripple graph represents the time-series used in Figure 4.2. The horizontal axis is located at the bottom. (a) Bars with the values less than ROI_{low} are filtered out. (b) All bars have the same height when ROI_{low} and ROI_{high} are set to the minimum of the measurement value.

is represented by a vertical strip with a specific color assigned by users. Since users can reduce the given height per series down to 1 pixel, this method provides one of more scalable representations regarding the number of series shown at once.

4.3 Visualizing degree of certainty between measurements

While the ripple graph emphasizes individual measurements, it can also show the overall pattern of change over time. Although the blood pressure changes continuously, it is measured discretely. A linear interpolation is conventionally employed as a simple way to fill the gap between discrete measurements. However, we note that the degree of certainty of an interpolated value between measurements decreases proportionally to the distance from a real measurement point. To the best of our knowledge, there has been no attempt to show the degree of certainty information in time-series data visualizations, which we believe contributes to enhancing the graphical integrity. The ripple graph fills the area between bars with the color of the closest bar while changing the alpha-channel value, where the color becomes more transparent as it gets farther from the closest real measurement point.

Let x be the distance between a real measurement point and a specific point between measurements. The degree of certainty (DOC) at the in-between point is determined as follows:

$$\text{DOC}(x) = -\frac{x}{c} + 1$$

, where c is a confidence interval, defined by users, which represents the temporal

range over which a real measurement holds its confidence. The alpha value at the in-between point is the product of $\text{DOC}(x)$ and a maximum alpha threshold (≤ 1). The maximum alpha threshold is empirically set to 0.8. Figure 4.4a and Figure 4.4b shows ripple graphs when the confidence interval is 3-hour and 6-hour, respectively. The upper graph in the figure shows how the alpha value changes over time.

The ripple graph has the following advantages with the degree of certainty representation: (1) it enhances the graphical integrity by showing predicted (or interpolated) values along with important context, i.e. the confidence of the predicted values; (2) the variability of measurement values along with measurement frequency is shown more clearly; (3) and it can effectively approximate the real continuous change over time for irregularly measured time-series data.

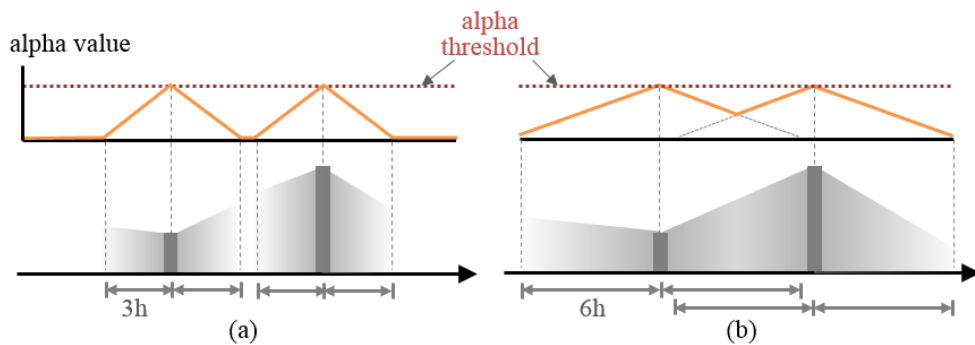


Figure 4.4. Ripple graph with confidence interval of (a) 3-hour and (b) 6-hour. The ripple graph fills the area between bars with changing the alpha value.

4.4 User interface for ripple graph manipulation

4.4.1 Control panel

We design an interface to enable users to flexibly adjust the parameters for a ripple graph (Figure 4.5). To adjust the range of the vertical axis for a ripple graph, user can drag two draggable vertical edges of the rectangle-shaped range slider on the ROI-widget histogram (Figure 4.5c) to set the range from ROI_{low} to ROI_{high} (e.g. 110 to 150 in Figure 4.5c). Users can also directly enter the desired value in an edit box on the top of each vertical line. A grey vertical line between two draggable vertical edges indicates the vertical location of the horizontal time axis. When ROI_{low} or ROI_{high} is changed, the grey vertical line moves to the midpoint between ROI_{low} and ROI_{high} .

A preview (Figure 4.5d) below the ROI widget shows the height and color of a bar depending on its value ranging from minimum to maximum. A horizontal line on the preview indicates the time axis. To examine the fluctuation of measurements around a specific meaningful value, a user can move the horizontal axis by dragging the line up and down to change the location of the axis between ROI_{low} and ROI_{high} . Then, the grey vertical line within the ROI widget (e.g. 130 in Figure 4.5c) also moves accordingly. Any changes of ROI values and the horizontal axis position cause immediate updates in the preview, the ROI widget, and the timeline view.

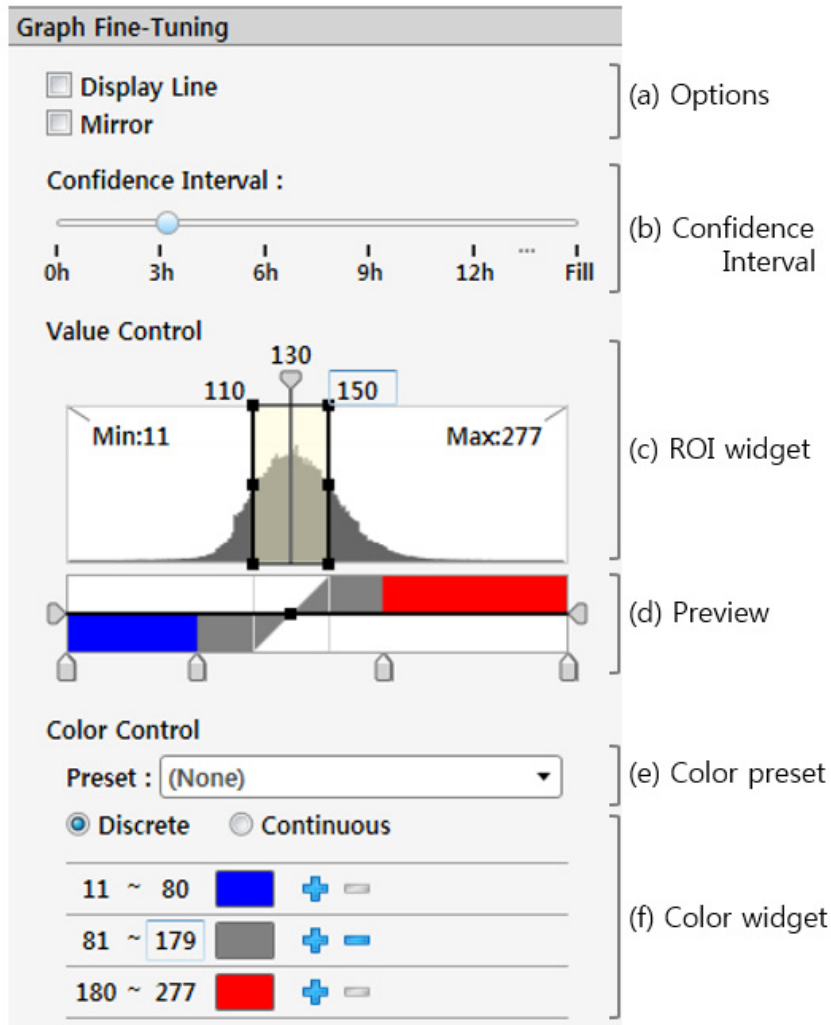


Figure 4.5. Control panel for ripple graph manipulation. Controls for (a) connecting bars and flipping bars downward around the horizontal time axis, (b) adjusting confidence interval, (c) adjusting ROI values, (d) showing a preview and changing a position of the time axis, (e) selecting a pre-defined color palette, and (f) manipulating color-related parameters.

4.4.2 Focus lens

When users know their range of interest of values in navigating time-series data, they can zoom in or assign colors to their specific value range using a control panel in Figure 4.5. But, if they want to see temporal pattern before and after a specific time or compare several time-series for multiple time-series, they might have a difficulty in assigning proper values to ripple graph parameters because they do not know the range of measurement value within the time range. To resolve this problem, we propose a new interaction technique, called focus lens, to interactively set ripple graph parameters according to regions defined by user on the ripple graph.

Users can designate several of regions of interest by using a rubber-band selection on the ripple graph while holding down the Shift key. Then, ripple graph automatically assign the minimum value and maximum value among measurement values within selected regions to ROI_{low} and ROI_{high} , respectively.

Figure 4.6 shows ripple graph to represent overall temporal pattern of change for a specific time-series using the focus lens. Because all measurement values are over ROI_{high} , all bars reach the top of the given space in Figure 4.6a. A user has a difficulty in seeing overall temporal pattern of change for the corresponding time-series. If a user designates the corresponding time-series using the focus lens, ripple graph assigns the minimum value and maximum value of the time-series to ROI_{low} (i.e. 147) and ROI_{high} (i.e. 205), respectively (Figure 4.6b). The user can see the overall temporal pattern of the time-series within orange rectangle.

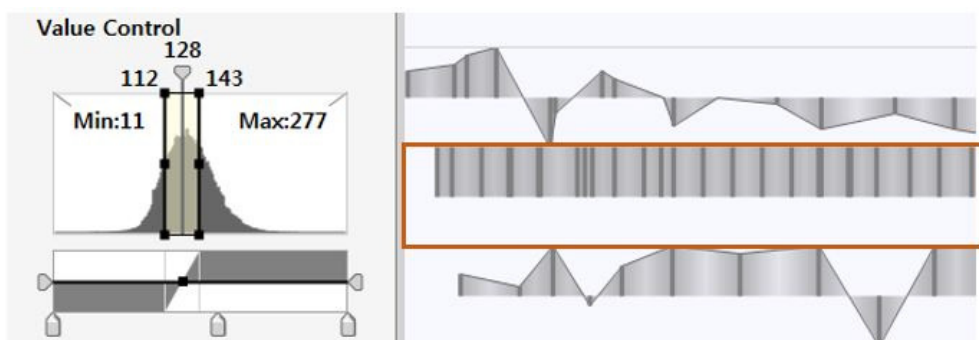


Figure 4.6. Ripple graph with the focus lens. (a) A user has a difficulty in seeing overall temporal pattern of change for the time-series within an orange rectangle. (b) After designating the corresponding time-series using the focus lens, the user can see the overall temporal pattern of the time-series while making the best use of the given space.

The ripple graph has the following advantages with the focus lens interaction:

(1) Focus lens prevents from splitting users' attention between controlling the ripple graph parameters and observing the graph; (2) Focus lens allows users to facilitate comparison task among time-series when they do not have a priori knowledge about measurement values; (3) Users can zoom in to measurement values within their specific regions of interest while maintaining the context.

Chapter 5

Usability Study

To verify the usefulness of ripple graph, we have conducted a usability study from usability and utility aspects [36]. For the usability aspect, we investigate whether users can easily and quickly become familiarized with the ripple graph. For the utility aspect, we check whether the ripple graph supports what users want for any problems they encounter.

5.1 Participants and materials

12 (5 females) volunteers from Seoul National University have participated in the study. The average age of participants was 27.6, ranging from 23 to 33 years of age. Four of them majored in computer science and engineering, and others were from four different majors (i.e., psychology, pharmacy, biology, and history). All participants were familiar with line graph or bar graph. The experiment took about

30 minutes and participants were rewarded about \$10. We prepared two time-series: a regional rainfall data and a stock market data. The rainfall data was a series of the amount of rainfall per rainy day in a specific region in 2011. Therefore, the interval of measurements of this data was irregular. The stock market data was a series of the prices of Google and Apple stocks from 1 January 2011 to 31 May 2012. Unlike the rainfall data, the interval of its measurements was regular. We included this data in our study to see if ripple graph works well with regular time-series data. Because all participants were familiar with the data, we gave only a brief description of the data.

5.2 Tasks

Participants performed 14 tasks of two different categories; usability category and utility category. For tasks in the usability category, participants had to make a graph the same as a given ripple graph. Participants were required to identify values of ripple graph parameters, such as ROI values, confidence interval, color, and position of the horizontal axis in the given ripple graph first. Because it was difficult for them to exactly identify and set these parameter values, we gave them some hints. For color, we made them select one of pre-defined color presets. For ROI values, we provided one of ROI_{low} , ROI_{high} , and position of the horizontal axis. Also, a position of mouse on graph defined by a participant was highlighted at the corresponding position on the given example graph. They performed one task at a time limit of 40 seconds not to spend too much time in fine-tuning the graph to make it perfectly same as the given ripple graph.

Table 5.1. Tasks of Utility Category. Elementary tasks to focus on an individual value for a time-series and Synoptic tasks to focus on the overall trend or relation with other series.

Class	Task	Question
Elementary	T7	Did it rain on July 3? If yes, how much rain fell?
	T8	What are the dates in which the rainfall exceeds 1000mm?
	T9	How many rainy days does A city have during June?
Synoptic	T10	What was the trend of Google stock?
	T11	Compare the rainfalls of A city and B city on June 22.
	T12	Find period in which the price of Apple stock higher than that of Google stock.
	T13	Find periods in which the price of Apple stocks is over \$620.
	T14	Compare the amount and frequency of precipitation in January and July.

For tasks in the utility category, the participants had to answer to a question about time-series. To cover most common tasks in analyzing time-series data, we selected representative tasks, i.e. questions based on the task model suggested by Andrienko and Andrienko [8]. The tasks were divided into two classes: (1) *Elementary tasks* to focus on an individual value for a time-series and (2) *Synoptic tasks* to focus on the overall trend or relation with other series. We list 3 elementary tasks and 5 synoptic tasks in Table 5.1.

5.3 Procedure

Before beginning a task, participants were given a tutorial on the ripple graph and the corresponding interface for about 15 minutes. Then they had time to try out the interface alone freely. After answering 4 practice questions correctly, participants performed 14 tasks (6 tasks in the Usability category and 8 tasks in the Utility category).

When performing a task, participants first had to understand the task appearing on the top pane in the usability testing software (Figure 5.1), and then they pressed the “Start” button when they were ready. After completing the task, they were asked to press the “Finish” button. They were asked to finish these tasks as fast and precisely as they could. At the end of a study session, we received comments on ripple graph.

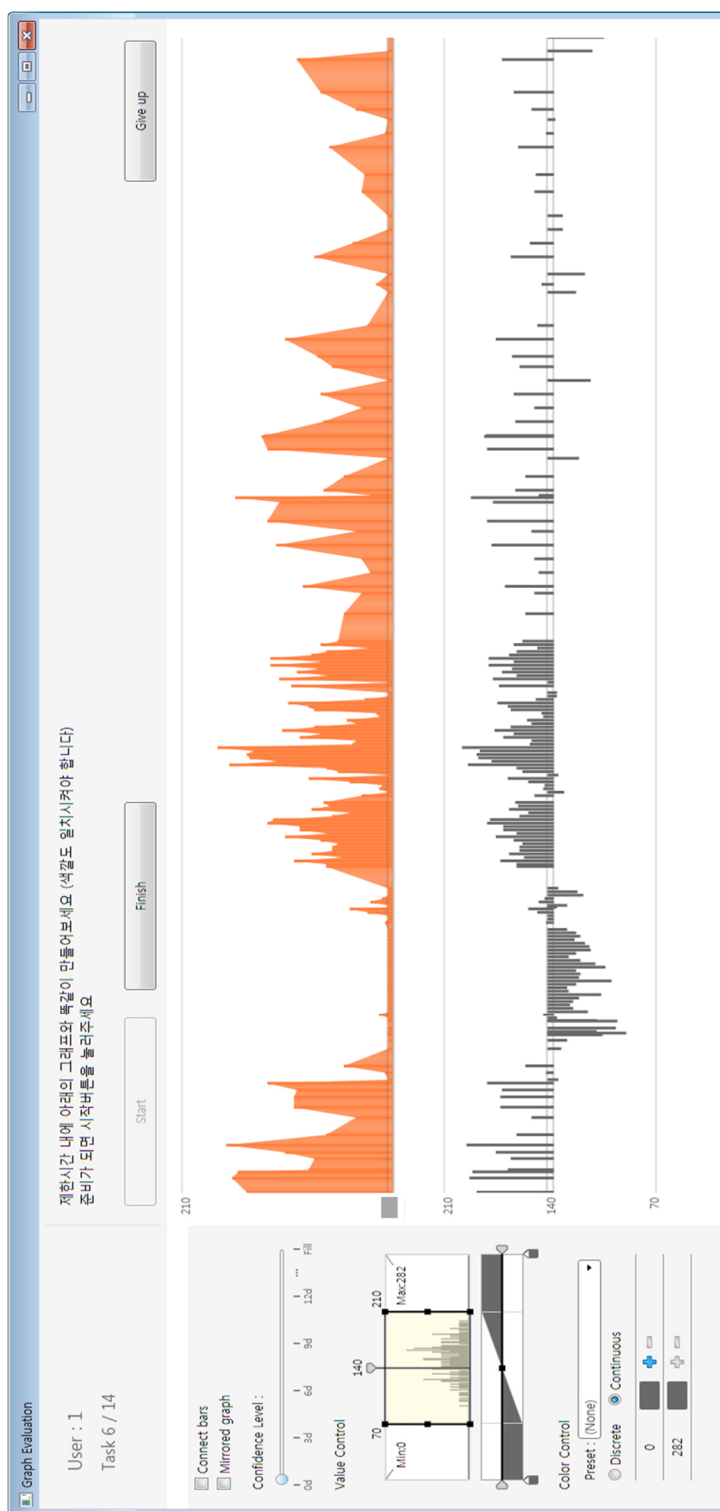


Figure 5.1. Usability testing software, showing the task in the Usability category. Participants had to make a graph (lower graph) the same as the given ripple graph (upper graph).

5.4 Results

For all tasks in the Usability category, most participants made a graph the same as a given graph and the average completion time was 11.08(± 7.16) seconds. There was a slight error of ROI values when no ROI values were provided as hint, but this error did not make a difference between two graphs. For accurate analysis, we defined *match rate* as a percentage of exactly matched pixels out of all pixels in a ripple graph. For tasks T1 through T4, *match rate* of all participants was 100%. For T5, *match rate* of a participant (P5) was 98.4% and that of all others was 100%. For T6, *match rate* was range from 99.8% to 100%.

For tasks in Utility category, most participants gave the correct answers. The average completion times for Elementary and Synoptic tasks were 43.5(± 14.8) and 44.7(± 7.1) seconds, respectively. Excluding T11, the completion time for Elementary tasks was 37.43(± 6.6) seconds, which means participants spent much more time completing T11 than other tasks. Because we stacked two graphs for Google stock and Apple stock vertically instead of overlapping two graphs in the same graph space for less visual clutter, it made comparison between two graphs difficult. Also, the price of Apple stock was higher than that of Google stock for only 5 days of the year, and the stock price difference was \$6 at most. So, participants spent much of their time adjusting ROI values and time range to find the answer. Table 5.2 shows the number of participants who failed and the average completion time for each task.

Table 5.2. Number of participants who failed and average completion time for each task.

Class	Task	number of participants failing task	average completion time (seconds)
Elementary	T7	0	29
	T8	2	35.6
	T9	1	43.6
Synoptic	T10	0	47.8
	T11	0	55.8
	T12	3	64.7
	T13	1	41.6
	T14	0	59.1

Participants were asked to answer the following 5 questions to collect subjective ratings about the ripple graph by using a 7 point Likert scale [Rating: 1=Strongly disagree; 7=Strongly agree].

Q1. This interface is easy to learn.

Q2. This interface is easy to use.

Q3. This interface is fun.

Q4. Overall, I like this interface.

Q5. I would like to use this interface again.

The average rating result was 5.7, 5.9, 6.6, 5.7 and 6.5, in order from Q1 to Q5.

The result shows that participants gave positive ratings for all questions, and

especially they enjoyed using the ripple graph.

Participants commented on the ripple graph as follows: “It’s nice to analyze data after focusing on the my range of interest”, “It’s very useful in identifying whether a value at a certain time is larger or smaller than a specific value”, “The preview helped me to understand the resulting graph”, “It is likely that I’ll be able to make various types of graphs once I get more used to it” and “I was satisfied with representation using confidence interval because it was easy to grasp the frequency for rainfall data while reducing visual clutter from a series of bars.”

5.5 Discussion

We analyzed how participants resolved problems for tasks in the Utility category and summarized their strategies as follows.

(1) Grasp of bar density

For a frequency-related question, all participants easily answered T9 by just counting bars and T14 by comparing density of bars.

(2) Maximization / Minimization of bar height

Some participants used a strategy to maximize the height of the bars for measurements with the range of interest. For example, in case of T8, a participant assigned ROI_{low} and ROI_{high} to 0 and 1000, respectively and moved the horizontal axis to the bottom. As a result, the bars with values greater than 1000 reached the top, and then he/she was able to answer the question just by counting the bars that met the top (Figure 5.2a).



Figure 5.2. Participant's Strategies for Question T8 "What are the dates in which the rainfall exceeds 1000mm?" (a) Maximization of the height of the bars for measurements greater than 1000 (b) Differentiation by the horizontal axis with measurement 1000 (c) Assigning a red color to measurements greater than 1000. (d) Minimization of the height of the bars for measurements less than 1000; This strategy caused a problem that the bars with values slightly greater than 1000 are indistinguishable from minimized bars.



Figure 5.3. Participant's Strategies for Question T13 "Find periods in which the price of Apple stocks is over \$620." (a) Minimization in the height of the bars for measurements less than 620 (b) Differentiation by the horizontal axis with measurement 620.

On the contrary, some participants used a complement strategy to minimize the height of the bars for measurements out of range of interest. For example, in case of T13, a participant moved the horizontal axis to the bottom and changed ROI_{low} to 620 so the height of bars with values less than 620 was minimized to the thickness of the horizontal axis. He/she could easily find the satisfying bars by just excluding the minimized bars (Figure 5.3a).

However, one participant made an error using this strategy for T8. He/she minimized the height of corresponding bars for values less than 1000, and made the height of bar for the values over 1000 to be proportional to the corresponding values. Consequently, it was difficult for him/her to distinguish bars with a value slightly greater than 1000 (e.g. 1020) from other minimized bars (note the red

arrow in Figure 5.2d). Although the correct answer for T8 was August 7 and November 18, where the amount of rainfall was 3002 and 1020 respectively, he/she answered only August 7.

(3) Differentiation by the horizontal axis

Some participants differentiated the range of interest and the range out of interest by the horizontal axis. For T8, a participant assigned ROI_{low} and ROI_{high} to the same value 1000. Then, the bars for the values less than 1000 were maximized downward from the axis. For the values more than 1000, the corresponding bars were maximized upward from the axis (Figure 5.2b). For T13, a participant moved the horizontal axis until its value became 620. He/she could get the right answer by just noting upward bars while adjusting the time range (Figure 5.3b).

(4) Utilization of color

For questions regarding a specific value, some participants assigned different colors to two ranges separated by a specific value. For example, for T8, a participant assigned gray to values less than 1000 and red to values over 1000. Then, he/she could easily perform the task by just identifying bars with red color (Figure 5.2c)

(5) Utilization of confidence interval

For a trend-related question, most participants analyzed the trend after filling area between bars while changing the transparency according to the confidence

interval. For T10, 8 out of 12 participants manipulated the slider bar for the confidence interval control without adjusting ROI values or adding colors.

Participants found an appropriate representation for given problems by making full use of an interactive widget. For a frequency-related question, their strategy (i.e. grasp of bar density) and very low incorrect answer rate implies that ripple graph based on bar graph is appropriate for visual analysis, especially frequency-related analysis, of irregularly measured time-series data.

Through the usability study, we found that participants preferred visualizing degree of certainty between measurements. They could see the variability clearly because this approach enabled them to perceive individual bars for a time-series as a gestalt (i.e. unified whole) instead of a series of scattered bars. Though ripple graph fills the area between bars according to degree of certainty, there was nothing wrong with grasping the measurement frequency because they had only to identify opaque bars (i.e. real measured values).

In spite of participants' efficient utilization and positive comments, it is difficult to conclude that our ripple graph is generally applicable visualization model for time-series because we focused quantitative values at only time points, not time intervals.

Chapter 6

Controlled User Study

We conducted a controlled user study to show the efficacy of the ripple graph by comparing it to existing representations (i.e. line graph, bar graph, and interactive horizon graph).

6.1 Participants and materials

We recruited 14 (5 females) volunteers from Seoul National University for the study. The average age of participants was 28.4, ranging from 23 to 35 years of age. Five of them majored in computer science and engineering, and the others were from four different majors, i.e. chemical and biological engineering, pharmacy, economics, and communication studies. All participants were familiar with the line graph and bar graph. The experiment took about 40 minutes and they were rewarded about \$5.

We prepared a time-series dataset of blood pressure measurements for 1600

stroke patients. We only used the measurements during the first 3 days after hospitalization to keep participants from distinguishing records by their different measurement periods.

6.2 Visualization techniques

We compared the following four visualization techniques for time-series data. The first three were among the most representative techniques.

(1) Line graph (LG)

We actually used a filled line graph with gray color to ease identification [28] (Figure 6.1a). To perform comparison tasks on different line graphs, they share the same range of values.

(2) Bar graph (BG)

Each bar with 3 pixels width was filled with gray color (Figure 6.1b). Like line graph, different bar graphs share the same range of values.

(3) Interactive horizon graph (IHG)

We implemented the horizon graph with zoom and pan interaction introduced in [39] (Figure 6.1d). The pan and zoom were provided using the mouse by dragging vertically anywhere on the screen with one of the two mouse buttons pressed. The left button triggered the zoom and the right button the pan as introduced in [39].

(4) Ripple graph (RG)

For a fair comparison with other three techniques, participants were only allowed

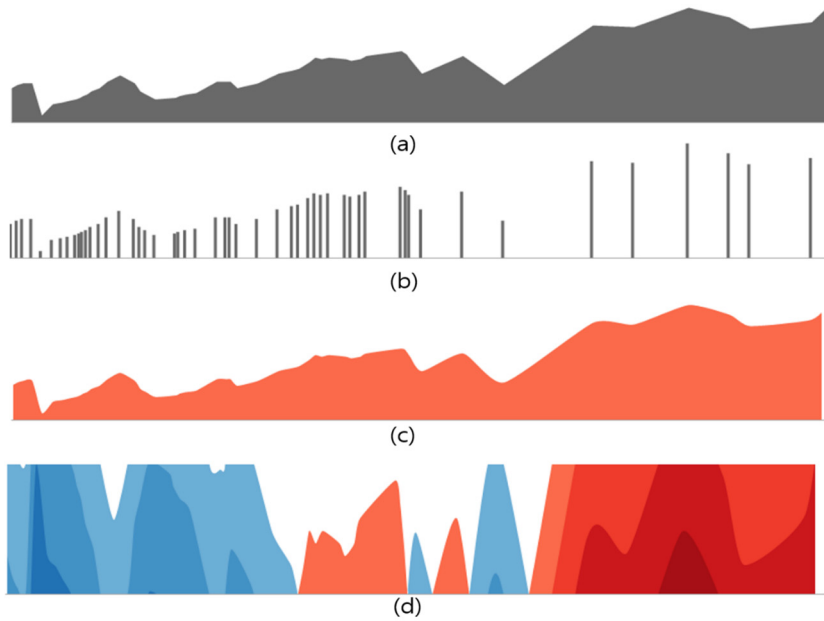


Figure 6.1 Visualization techniques for time-series. (a) filled line graph. (b) bar graph. (c) horizon graph by default. (d) horizon graph after pan and zoom interaction

to change the *ROI* values and move the horizontal axis from a control interface for the ripple graph.

We used a split-space technique where each time-series is shown in a row of the same height [28]. We fixed the height for each time-series at 24 pixels to facilitate comparison between our results and previous studies in [25] and [39].

6.3 Tasks

Based on the task model suggested by Andrienko and Andrienko [8] and user studies on the graphical perception of multiple time-series [25][39], we chose the following three types of tasks.

(1) **Max**

Selecting a time-series with the highest value across all records.

(2) **Same**

Selecting a time-series which is exactly the same as a given record.

(3) **Frequency**

Selecting the most frequently measured time-series.

(4) **Confidence**

Determining the subjective confidence in the value at a given time point. The same number of time points was selected from real measurement points and those in-between. The value for a point between two adjacent measurement points was linearly interpolated. Because BG represents only real measurement points, it was excluded from the confidence task.

6.4 Study design and procedure

We ran the study as a within-subjects design, with each participant performing all the tasks using all the visualization techniques. We ran the experiment as a 4 (Visualization technique: LG, BG, IHG, and RG) \times 4 (Task type: Max, Same, Frequency, and Confidence) \times 2 (Number of time-series: 20 and 40) \times 2 (trials) design while counterbalancing the order of visualization techniques. Performance time and correctness of answers were the dependent variables of this study. For each trial, we randomly selected a small number (20 or 40) of time-series from the pool of 1600 time-series to avoid a learning effect, keeping the following constraints to maintain

the same level of task difficulties across the four techniques. For all measurements of selected time-series

- (1) The mean of all measurements for selected time-series: $142(\pm 5)$
- (2) The standard deviation of all measurements for selected time-series
: $17(\pm 2)$
- (3) The difference between the largest and the fifth largest measurements
: 14 occupying 3 pixels on screen

Before beginning real tasks, we gave participants a tutorial on a visualization technique and showed them how to perform 4 types of tasks with an example. They also had enough time to try out each technique by themselves. Then, they performed 12 ($3 \times 2 \times 2$) tasks for BG or 16 ($4 \times 2 \times 2$) tasks for others. They were asked to finish tasks as fast and precisely as they could. We measured the task time and correctness. At the end of a study session, participants filled in a questionnaire for subjective evaluation of each visualization technique.

We established three hypotheses for this study.

- (1) For the Max and Same tasks, IHG and RG will outperform LG and BG in both the task time and correctness.
- (2) For the Frequency task, BG and RG will outperform LG and IHG in both the task time and correctness.
- (3) For the Confidence task, RG will outperform LG and IHG in distinguishing between real measurement values and interpolated values.

6.5 Results

We analyzed the task time and correctness with a 4 (Visualization technique) \times 3 (Task type excluding Confidence) \times 2 (Number of time-series) repeated-measures analysis of variance (RM-ANOVA). Regarding the task time, we found a significant main effect of Visualization technique ($F_{3,312} = 3.53, p < .05$).

Figure 6.2a and Figure 6.2b show the mean task time and correctness for each

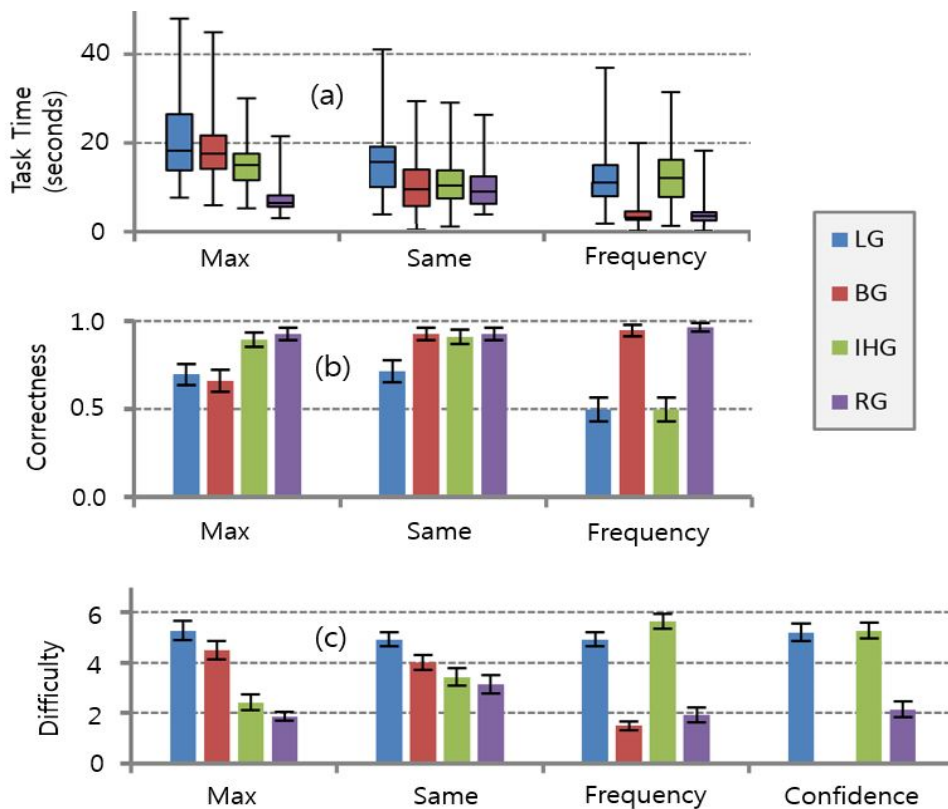


Figure 6.2. (a) Task time, (b) Correctness, and (c) Difficulty for each visualization technique by task type.

technique by task type, respectively. To analyze differences among the visualization techniques, we also conducted Tukey's HSD post-hoc test for each task type (Table 6.1, Table 6.2, and Table 6.3).

Table 6.1. Results of post-hoc test by each task type. The < sign represents the inequality relation with a statistical difference ($p < .05$).

Records	Task	Pairwise comparisons
Task completion time	Max	RG < IHG < LG & BG
	Same	RG & BG & IHG < LG
	Frequency	RG & BG < LG & IHG
Correctness	Max	LG & BG < IHG & RG
	Same	LG < BG & IHG & RG
	Frequency	LG & IHG < BG & RG
Difficulty	Max	RG & IHG < LG & BG
	Same	RG & IHG < LG
	Frequency	RG & BG < LG & IHG
	Confidence	RG < LG & IHG

Table 6.2 Results of Post-hoc test by each task type for the task time. * means that there is a significant difference between the mean task times of two techniques.

Task	Technique	Mean Task Time (s)	p values with Tukey's HSD post-hoc		
			BG	IHG	RG
Max	LG	20.86	0.657	0.003*	< 0.001*
	BG	19.12	-	0.046*	< 0.001*
	IHG	15.45		-	< 0.001*
	RG	7.28			-
Same	LG	16.13	< 0.001*	< 0.001*	< 0.001*
	BG	10.46	-	0.967	0.965
	IHG	10.98		-	0.786
	RG	9.93			-
Frequency	LG	11.73	< 0.001*	0.965	< 0.001*
	BG	4.21	-	< 0.001*	0.998
	IHG	14.37		-	< 0.001*
	RG	3.91			-

Table 6.3 Results of Post-hoc test by each task type for the correctness. * means that there is a significant difference between the mean correctness of two techniques.

Task	Technique	Mean Correctness	p values with Tukey's HSD post-hoc		
			BG	IHG	RG
Max	LG	0.70	0.955	0.027*	0.006*
	BG	0.66	-	0.006*	0.001*
	IHG	0.89		-	0.955
	RG	0.93			-
Same	LG	0.71	0.005*	0.011*	0.005*
	BG	0.93	-	0.992	1.000
	IHG	0.91		-	0.992
	RG	0.93			-
Frequency	LG	0.50	< 0.001*	1.000	< 0.001*
	BG	0.95	-	< 0.001*	0.995
	IHG	0.50		-	< 0.001*
	RG	0.96			-

The results supported our first and second hypotheses except for the Same task with BG. Participants completed the Same task in a significantly less time and with significantly more correct answers with BG than with LG. The reason might be that BG enabled participants to quickly filter the target time-series by preattentively perceiving the measurement frequency of time-series.

Participants usually spent more time in completing tasks with 40 time-series than with 20 time-series as shown in Figure 6.3. However, there were exceptions where participants showed similar performance for both cases. The likely reasons for the exceptions are as follows.

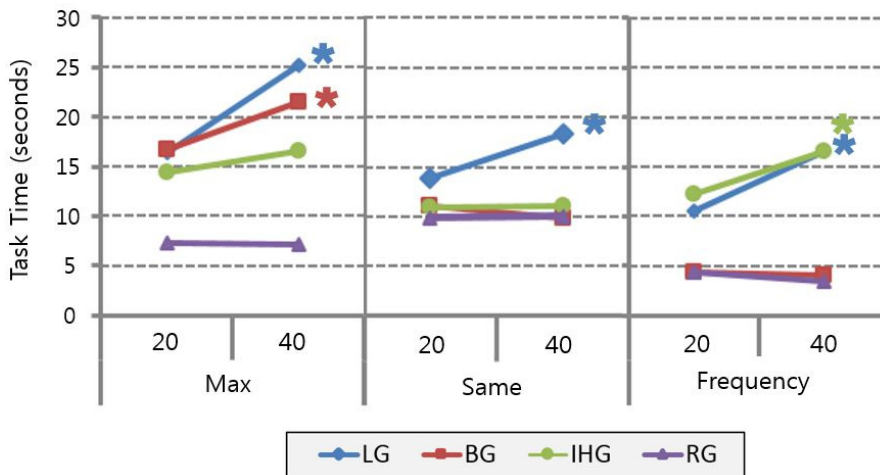


Figure 6.3. Task time for each visualization technique by number of time-series (i.e. 20 and 40). * indicates a significant difference in task time between 20 and 40 time-series ($p < .05$).

For the Max task with RG, participants could identify the target time-series at once just by narrowing the range of interest of values down to maximum value.

For the Same task, participants could identify the target time-series at once by preattentively perceiving a different color of a band with IHG and by grasping the distinctive bar occurrence frequency with BG and RG. In contrast, with LG, they had to use their elementary perceptual skills to find the target.

For the Frequency task, participants could grasp the frequency at a glance with BG and RG by just perceiving the overall occurrence pattern of bars.

For the Confidence task, we analyzed the results after dividing selected time points into three categories: (1) real measurement points with extreme values, (2) real measurement points with non-extreme values, and (3) interpolated points between two adjacent real measurement points (Figure 6.4).

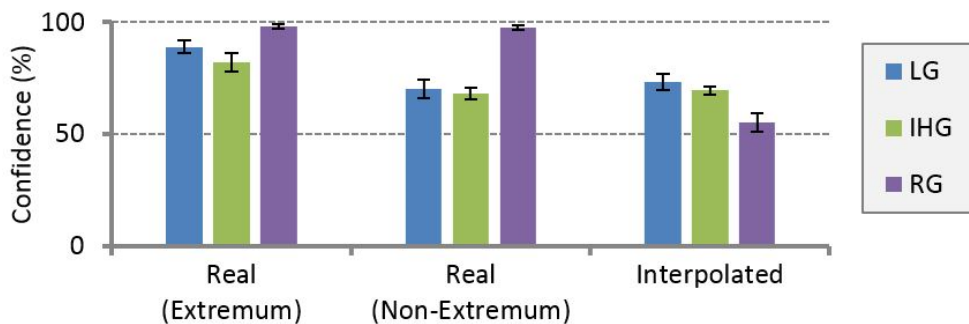


Figure 6.4. Average of subjective confidence results for each visualization technique by time point type

For real measurement points, the confidences with LG and IHG were lower than those with RG. Especially, for real measurement points with non-extreme values, the confidences of LG and IHG were as low as those for interpolated points. It is likely because connected lines or filled areas in LG and IHG made it difficult to tell if such measurement points are real or not. Participants also answered that they actually felt a difficulty in performing the Confidence task with LG and IHG (Figure 6.2c). In contrast, participants easily performed the Confidence task with more reasonable rating of the confidence values with RG (Figure 6.2c and Figure 6.4).

We asked each participant to rate how difficult each task was by using a 7 point Likert scale [Rating: 1 = Very easy; 7 = Very difficult] (Figure 6.2c). We analyzed the subjective rating results using Friedman test with Bonferroni correction. We found a significant main effect of Visualization technique ($X^2(3) = 53.2, p < .001$). Pairwise comparisons among visualization techniques for each task show that participants had less or same difficulty in performing a task with RG than others across all task types (Table 6.1 and Table 6.4).

Table 6.4 Results of Bonferroni correction by each task type for the difficulty.

* means that there is a significant difference between the mean difficulties of two techniques.

Task	Technique	Mean Difficulty	p values with Bonferroni correction		
			BG	IHG	RG
Max	LG	5.29	0.333	< 0.001*	< 0.001*
	BG	4.50	-	< 0.001*	< 0.001*
	IHG	2.43		-	0.606
	RG	1.86			-
Same	LG	4.93	0.190	0.010*	0.002*
	BG	4.00	-	0.598	0.251
	IHG	3.43		-	0.924
	RG	3.14			-
Frequency	LG	4.93	< 0.001*	0.249	< 0.001*
	BG	1.50	-	< 0.001*	0.674
	IHG	5.64		-	< 0.001*
	RG	1.93			-
Confidence	LG	5.21	-	0.987	< 0.001*
	BG	-	-	-	-
	IHG	5.29		-	< 0.001*
	RG	2.14			-

6.6 Discussion

Our controlled user study results supported the three hypotheses, suggesting that the ripple graph was overall the best performing in terms of task time, correctness, and subjective satisfaction. Our results also suggested that the interactive horizon graph was not an appropriate technique for measurement frequency-related analysis of irregularly measured time-series data even though it was known to be good for quantitative comparison of multiple time-series [39]. Although there was no difference in performance for the Frequency task between BG and RG, participants commented that the bar graph caused confusion and eye fatigue when multiple time-series were displayed.

We found in several participants' comments that it was helpful to see the degree of certainty between real measurements. They could see the variability clearly in the ripple graph because visualizing the degree of certainty enabled them to perceive a time-series with many individual bars as a gestalt (i.e. unified whole) instead of a series of scattered bars.

Chapter 7

Stroscope

Our 6 month-long participatory design with the neurologists leads us to implementing an interactive visualization tool for time-series data, entitled “Stroscope,” where the ripple graph is the main visualization component. In this section, we explain the user interface and interaction models of Stroscope along with related analytical features.

7.1 Layout

Stroscope consists of three main areas (Figure 7.1): Control panel, timeline view, and detail view. The control panel in the left area has four tabs: (1) the control tab for sorting, filtering, and aligning time-series, (2) the fine-tuning tab for adjusting various parameters of the ripple graph, (3) the grouping and layout tab for changing the view layout and dividing records into groups, and (4) the variable tab for creating a custom variable out of existing variables.

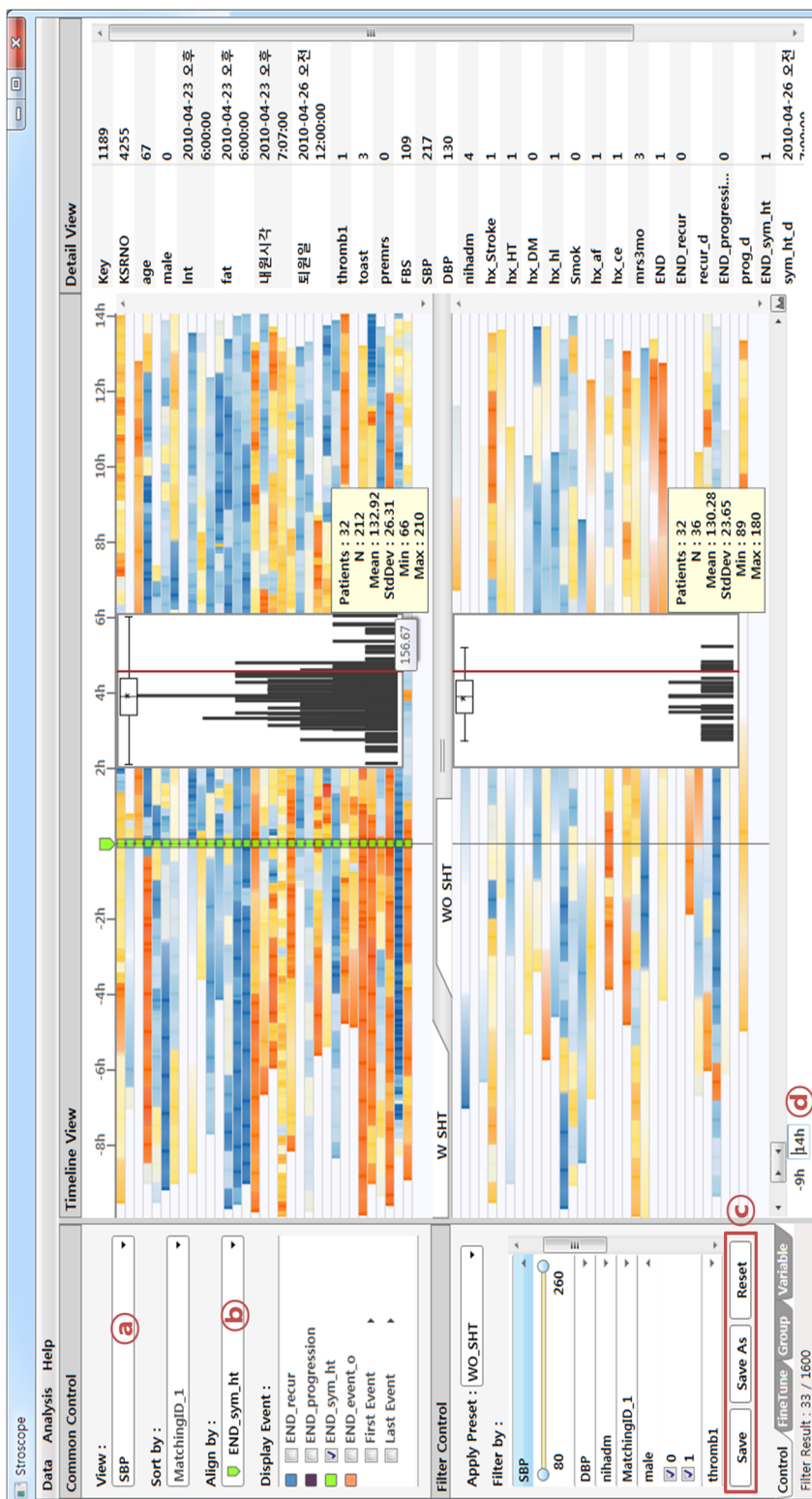


Figure 7.1. Overview of Stroscope.

The timeline view in the middle area visualizes multiple time-series using the ripple graph, where each series is vertically stacked with the same height. All series are aligned by the first measurement time by default. The alignment indicator (i.e. the green marks in the upper timeline view and the vertical line in the lower time-line view in Figure 7.1) highlights the alignment time, where the relative time is zero.

The detail view in the right area shows the information of the selected series in the timeline view, and this information is immediately updated upon any selection.

7.2 User Interaction

Stroscope provides effective means to help users efficiently explore a large time-series data. They include common operations that many existing time-series visualization tools support, e.g. ARF framework in [60] for alignment, ranking, and filtering.

(1) Select a measurement variable

Time-series data can have more than two contemporaneous variables changing over time, i.e. multivariate time-series. In blood pressure data, there are two such variables: SBP and DBP. Furthermore, Stroscope automatically generates a delta measurement variable that is defined as a sequence of differences between consecutive measurements (\triangle SBP and \triangle DBP) to help users easily examine variability. Users can select one of these measurement variables using a combo box (Figure 7.1(a)).

(2) **Align**

An alignment operation enables users to effectively compare temporal patterns before and after a specific event across multiple records. It helps users to predict prognosis and determine a treatment for a patient. Stroscope allows a user to align all records by a clinical variable (such as onset time and hospital arrival time) or by a specific event (such as END_progression). In addition, all records can be aligned by an absolute time, e.g. the oldest time among timestamps. Then, the time label is shown in the form of “yyyy-mm-dd.” Users can select one of these variables for alignment in a combo box (Figure 7.1⑥).

The alignment indicator at the top of the timeline view is positioned at the top left end by default and filled with the corresponding color to the selected alignment variable. Users can drag the indicator left or right within the entire time range to check the trend before and after the alignment point in detail.

(3) **Event display**

Stroscope displays events with a colored rectangle at the corresponding location. Users can select the events to see by checking a check box and change the color for the event by clicking the colored rectangle in a check box.

(4) **Filter/preset**

The user can filter out the unnecessary records by adjusting the range slider bar for a continuous variable and by unchecking the check box for a categorical variable. It is also possible to save the filtering conditions as a preset and reuse

it later.

(5) **Zoom**

Our multi-scale visualization model, the ripple graph allows users to dynamically change the visual representation depending upon the available display space. To maximize space utilization, Stroscope provides a vertical zooming to adjust the height of the ripple graph for each record through wheeling the mouse on the timeline view. A zooming on the horizontal axis is also provided to help users focus on a temporal period of interest (Figure 7.1④).

(6) **Hierarchical grouping**

To detect differences among groups classified by categorical variables (e.g. gender or age group) is one of the routine analytic tasks for neurologists. Stroscope facilitates this task by enabling hierarchical groupings. The result of a grouping is summarized in an enhanced treeview control (Figure 8.9) where each node has both the number of corresponding records and a check box to show/hide the corresponding group. The control panel in Figure 7.1 shows the result of a hierarchical grouping where all records are divided into groups by “Cluster” first and then by “END” categorical variable.

(7) **Side-by-side comparison**

One of the main tasks in the analytical exploration of the blood pressure data is to compare between groups. To facilitate such comparison task, a horizontal/vertical juxtaposition of views for the two groups is necessary. A user

can define new groups by making filter presets (Figure 7.1©) and apply a different preset to each view after splitting the timeline view horizontally or vertically into two. The zoom factor and the position of the alignment indicator are synchronized between two views. Figure 7.1 shows two horizontally split views.

7.3 Analytical Features

(1) Statistical summary measures on demand

Providing statistical summary measures on users' demand is important for gaining insights and performing a quick-and-dirty hypothesis testing in an interactive visualization tool. In Stroscope, a user can designate a region of interest by using a rubber-band (rectangle) selection on the timeline view, and check the descriptive statistical summary measures of the region. The summary measures include the number of records, number of measurements, average, standard deviation, minimum value, and maximum value. Stroscope provides the summary measures not only numerically, but also graphically in a histogram and a box plot together within the selection rectangle (grey rectangle on timeline view in Figure 7.1).

Using the side-by-side comparison feature, users can instantly compare two groups in terms of the statistical summary measures. The linking and brushing technique implemented in Stroscope enables a user to select a rectangular region

in one timeline view and see the same region selected in the other timeline view. This feature could encourage users to interactively perform their routine analytic task without cognitive overload.

(2) Accommodating individual differences

In the blood pressure data, there exist individual differences among patients in terms of age, weight, medical history, and so on. A value of 150 is generally considered a slightly higher value, but the value can be a critical value for a patient with relatively lower blood pressure. However, for all patients, the value of 150 was represented by the same height and color in the ripple graph. To resolve this issue, Stroscope allows a user to define a reference variable with a different basis value for each patient. For example, let us consider that a user defines the average blood pressure during 3 days before discharge as a reference variable with “SBPMean” name. Then Stroscope calculates the basis value for every patient according to the definition of the reference variable, and generates a new measurement variable that is defined as the difference between the real blood pressure value and the basis value for each patient. Consequently, a user can accommodate individual differences among patients using the reference variable.

(3) Generate evenly spaced dataset

In blood pressure data, the measurement intervals are irregular and the number of measurements is different for each patient. For example, when a user

compares group A that TOAST value is 1 and group B that TOAST value is 2, the number of blood pressure measurements exceeding a specific threshold can be larger in group B. This could be because patients in group B actually have more blood pressure measurements exceeding the threshold because of a risky factor, e.g. size of lesion. But it might be in part due to the fact that the blood pressure was measured more frequently in group B than group A. To address this issue, Stroscope allows users to generate a time-series data with a regular interval. If a user designates Interval and Aggregated value (mean/maximum/nearest neighbor value), Stroscope generates a sequence of aggregated values with the regular interval.

In our case study, this feature was used as secondary method for gaining insights, because this method causes losing the measurement frequency and may result in loss or duplication of the measurements according to the sampling interval.

(4) **Matching**

Users are commonly interested in identifying factors that may contribute to a clinical outcome such as recurrence of END by comparing patients who have the condition (case) with patients who do not have the outcome (control) but are otherwise similar [13]. To help users conduct this kind of case-control study, Stroscope supports the matching of the case to the control. For example, users can match ‘patients with the END_recur event’ to ‘patients without the

END_recur event’ according to various variables (e.g. gender should be exactly matched and the age should not be different by more than 5). Then, Stroscope generates a new variable with a user-defined name, where one record in the case and one or more matched record(s) in the control have the same value for the variable.

(5) **Data-space clustering and image-space clustering**

Users can gain insights by finding any regularities or anomalies through clustering. We used the I-kMeans algorithm [55], an interactive k-Means clustering method taking advantage of the multi-resolution property of wavelets. We enhanced this algorithm by making it applicable to irregularly measured time-series data because the algorithm assumes that each time-series data is measured regularly over time.

Stroscope provides two kinds of clustering techniques: data-space clustering and image-space clustering. For the data-space clustering, records that have similar measurement values are grouped together, which always results in the same clusters for the same dataset. However, a coarse color mapping could result in a visual inconsistency problem that the resulting visualization does not look well-clustered. For example, a neurologist assigned any measurements of 180 or greater to a red color because those are considered as critical values. However, data-space clustering could separate two measurements in the same red, e.g. 180 and 240 because they were numerically very different in the data space. This

result could confuse users due to the inconsistency between users' mental model expressed in color mapping and clustering results based on actual data values.

The neurologist suggested an image-space clustering during the participatory design process in order to resolve this visual inconsistency problem of the data-space clustering. In the image-space clustering, records with a similar color pattern are clustered together, where the clustering results could vary according to the color table defined by the user, but the results make more sense to the user who expresses his intention in his color mapping choice.

7.4 Implementation

Stroscope is implemented in C# with Windows Presentation Foundation (WPF). We used an observer pattern, one of software design patterns [22]. This pattern supports that for any state change of an object called subject, other objects called observers are automatically notified, where subject maintains a list of its dependents, i.e. observers. This pattern is efficient in implementing the linking and brushing.

Stroscope can display more than two different time-series datasets in a multi-monitor environment. It is possible to individualize or synchronize parameters of ripple graph among different monitors. In a 24-inch 1920 x 1080 resolution monitor, Stroscope can display up to about 950 (1900 in the side-by-side comparison mode) time-series interactively on screen at once.

Chapter 8

Case Study

We designed and developed Stroscope involving three neurologists in the design process. After completing an initial prototype, we have conducted two case studies with neurologists dealing with the blood pressure data for four months to show the efficacy of Stroscope in the real field.

8.1 Procedure

We had a meeting with two participants (female and male doctors enrolled in a stroke fellowship program) together for about 90 minutes every 2 or 3 weeks for the first two months for exchanging ideas and sharing findings while improving the prototype as well if necessary. We had a 1-hour meeting with each participant every 2 weeks for the rest two months. We used the following procedure for each meeting: (1) Participants gave us feedback on Stroscope. (2) We installed an improved version and

explained the improvements. (3) We let participants try Stroscope to confirm whether they understood the new features. In this stage, we employed a modified pair analytics method, where assistance is provided only when participants asked for help. (4) After the meeting, we updated Stroscope following the feedback and maintained contact with the participants by answering their questions via e-mails or phone calls.

When the two participants first tried Stroscope, they were impressed by its visual and interactive nature because they had never used such a visualization tool before. They were also excited that they could find significant patterns in a specific group by comparing different groups using matching, aligning, and clustering.

8.2 Participant 1 (P1)

P1 was interested in the relationships between the variability in blood pressure and the occurrence of symptomatic hemorrhagic transformation (SHT) of acute ischemic stroke. SHT is one of the important factors that influence the outcome of stroke treatment. Previous studies have shown that the occurrence of SHT relates to high variability in blood pressure. But statistical summary measures did not bring him an intuitive understanding and it was always elusive to examine each record separately in detail using conventional statistical tools.

P1 decided to conduct a case-control study with Stroscope. He first defined two groups: cases are the patients who have had a SHT and controls are the patients who have not had a SHT. He used the matching function of Stroscope to match each patient in the cases to the patients in the controls according to the initial SBP ($\pm 5\text{mmHg}$) and

age (± 5 years). After he split the timeline view into two (up and down), he assigned the cases to the upper view and the controls to the lower view. He immediately noticed without any additional manipulation that the blood pressure was measured more frequently and the hospitalization period was longer for the patients in the cases. For a detailed analysis, he aligned patients by the SHT onset time, while aligning patients in the controls by the SHT onset time of the patient matched in the *cases*. High or low values are observed more frequently near the SHT event in the cases than in the controls.

To analyze the variability of SBP, he first selected Δ SBP as a measurement variable and manipulated the color palette to make positive values red and negative values blue. It was difficult to see the difference of variability between cases and controls by just checking the occurrence of dark blue or dark red regions. So, he dragged the horizontal axis of the ripple graph to the bottom to see only the increasing periods with positive Δ SBP values (Figure 8.1).

In the same manner, he dragged the horizontal axis to the top to see only the decreasing periods with negative Δ SBPs (Figure 8.2). Then he could clearly see the difference between cases and controls, e.g. a rapid change in the blood pressure was observed more frequently in the cases near the occurrence of SHT event. During 6 hours before and after the SHT onset event in Figure 8.2, blue regions representing the periods with decreasing SBP by more than 20 appear more frequently in the cases (upper timeline view) than in the controls (lower timeline view).

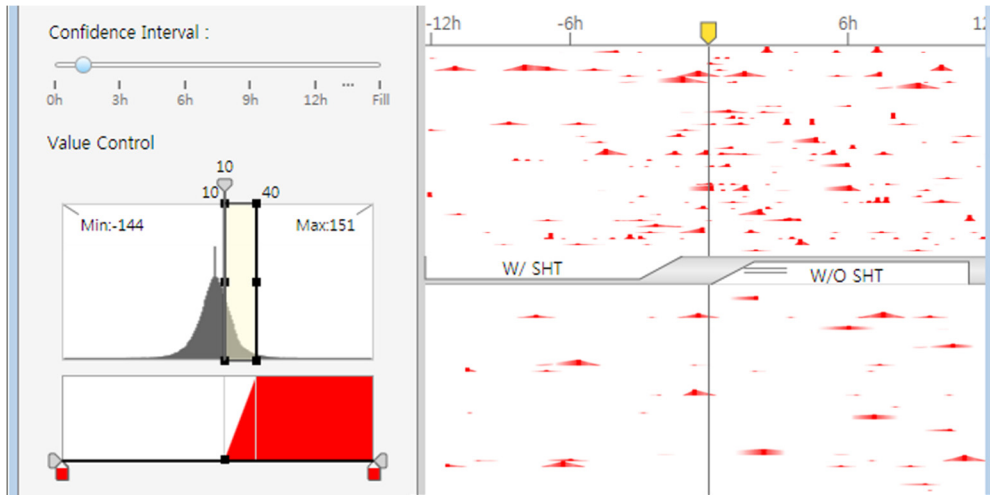


Figure 8.1. Stroscope showing only SBP increasing periods. The yellow mark indicates when an SHT onset event occurred.

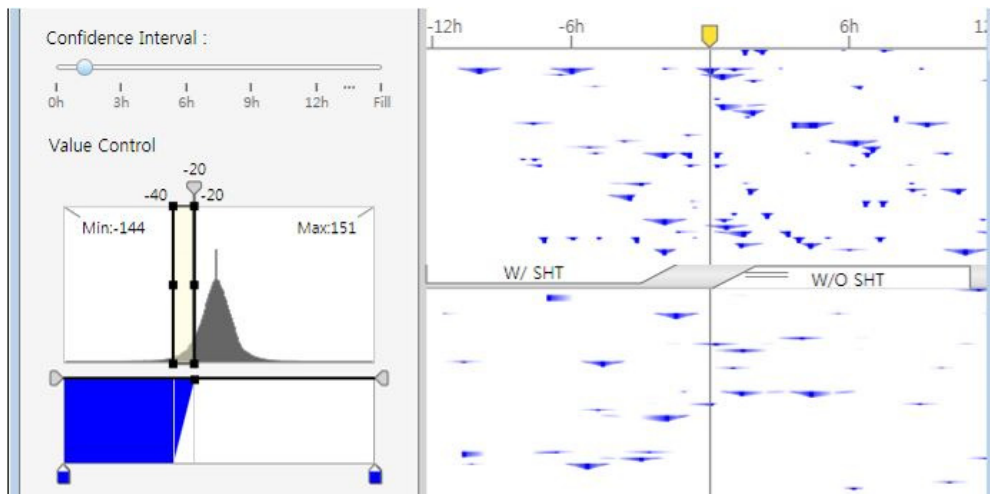


Figure 8.2. Stroscope showing only SBP decreasing periods. After selecting Δ SBP (a difference between consecutive SBP values) as a measurement variable, P1 can observe Δ SBP values less than -20. The yellow mark indicates when an SHT onset event occurred.

To minimize the influence of individual differences among patients, he decided to examine the deviation of SBP values. He first defined a reference variable named “SBPMean” as the average SBP during 3 days before discharge. And then, he changed the measurement variable to a new reference variable defined by $SBP - SBP_{Mean}$. He easily found that there were many extreme values, especially higher values in dark red in the cases, before the occurrence of SHT (Figure 8.3). In addition, dark red and dark blue colors were observed *more frequently* in the cases, indicating that the variability of SBP was high.

While observing the patients in the cases, he found one patient with an anomalous pattern: the patient exhibited a dramatic decrease of the “ $SBP - SBP_{Mean}$ ” value about seven hours before the occurrence of SHT (see the black arrow in Figure 8.3). After zooming in to the patient further, he found that the “ $SBP - SBP_{Mean}$ ” value decreased approximately from 25 to -50 (see the black arrow in Figure 8.4).

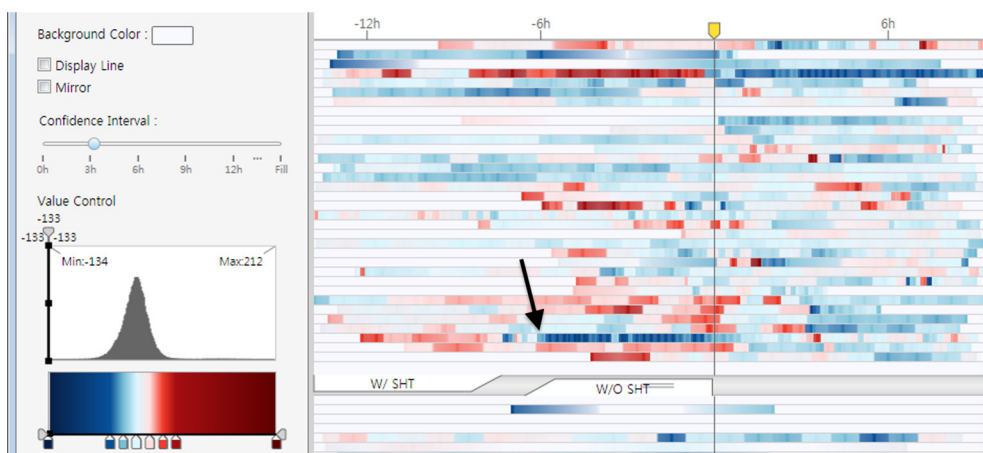


Figure 8.3. Stroscope showing a “SBP-SBPMean” variable. A black arrow indicates a patient with an anomalous pattern. Dark blue represents a sudden decrease in SBP (a great negative value of “SBP-SBPMean”). The yellow mark is the indicator of SHT onset event.

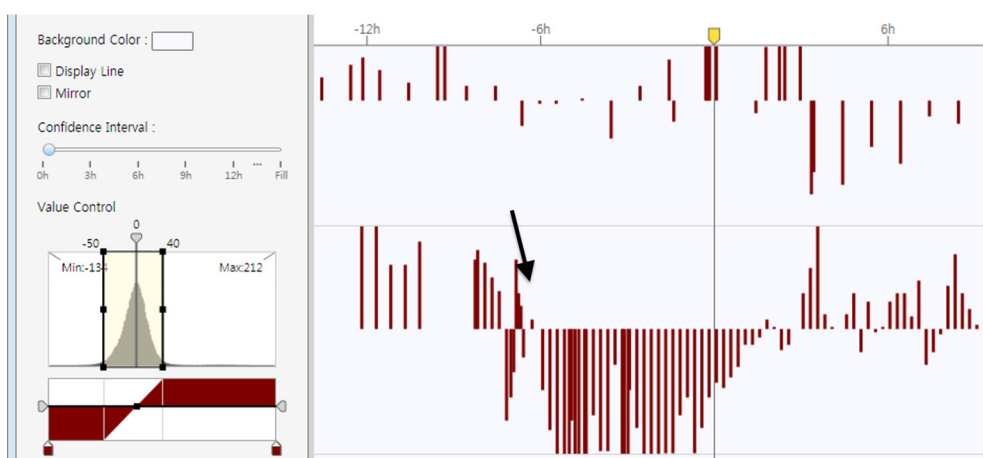


Figure 8.4. Stroscope showing the detail of the patient with anomalous pattern in Figure 8.3. P1 identified that the “SBP-SBPMean” value decreased approximately from 25 to -50 about 7 hours before the SHT onset. The yellow mark is the indicator of SHT onset event.

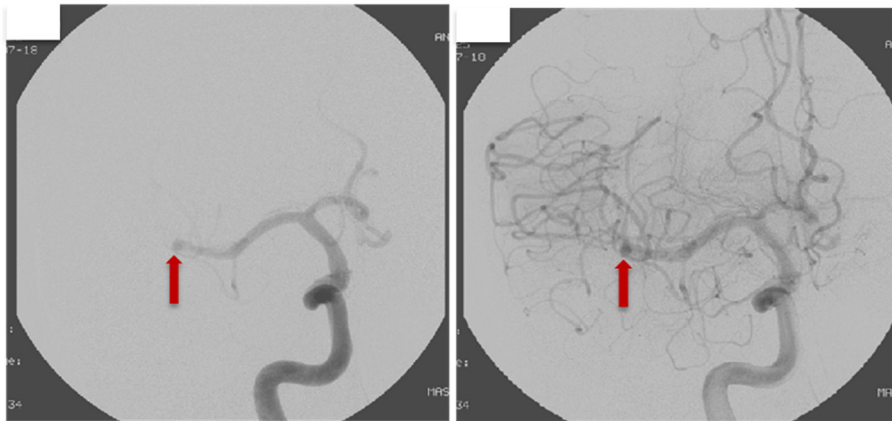


Figure 8.5. MRI scans before (left) and after (right) the treatment to remove the offending thrombus.

He wondered why the SBP value suddenly decreased. He checked the medical history of the patient in the EMR (electronic medical record) system. He found that the patient received a treatment called mechanical thrombectomy for an occlusion in the sphenoid segment of the middle cerebral artery (a part of blood vessel in the brain). He reconfirmed the sudden decline in the SBP value through two MRI scans before and after the treatment to remove the offending thrombus (Figure 8.5).

8.3 Participant 2 (P2)

In P2's clinical research, she often found that patients' blood pressure increased or decreased rapidly when they got worse. But there are previous studies that have shown conflicting results because most of these studies were cross-sectional which compared only statistical summary measures between groups without taking into

account the temporal aspect of blood pressure change. P2 wanted to go beyond the statistical summary measures by visually exploring individual blood pressure values and their changes over time using Stroscope.

As soon as she loaded her dataset and sorted patients by TOAST (a classification stroke according to the causes of stroke), P2 started to make her discoveries in her dataset: (1) Overall, SBP value was decreasing during a day after the first measurement; (2) patients in ‘TOAST 2’ and ‘TOAST 6’ groups were hospitalized for a shorter period of time; and (3) SBP values of patients in ‘TOAST 3’ and ‘TOAST 4’ groups tended to be relatively low. P2 also found patients with anomalous patterns differentiated by color (Figure 8.6), frequency (Figure 8.7), or fluctuation (Figure 8.8).

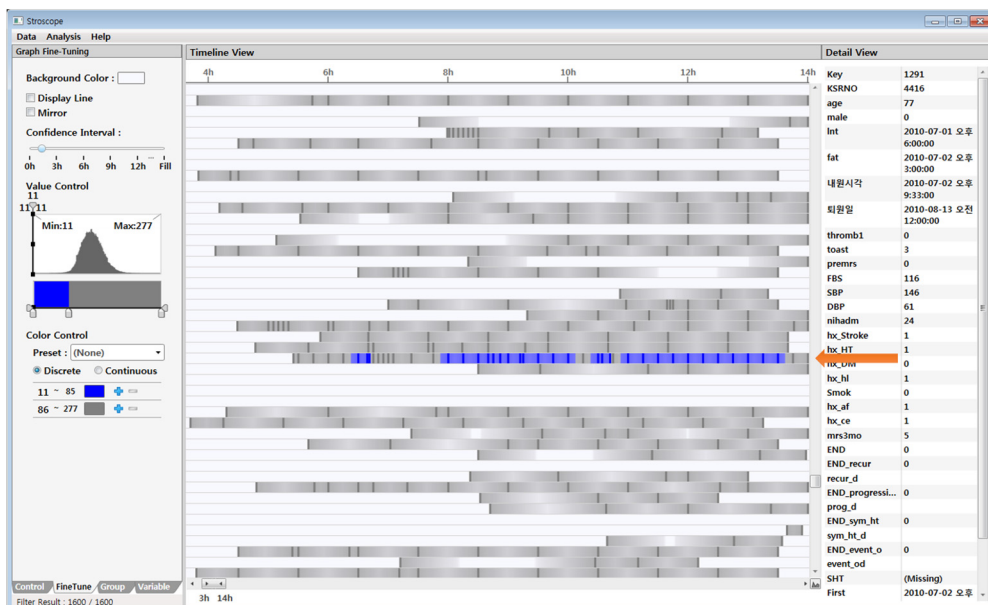


Figure 8.6. Stroscope showing values under 85 in blue color. An orange arrow indicates a patient with an anomalous pattern because low values (i.e. values under 85) are observed frequently for only the patient.

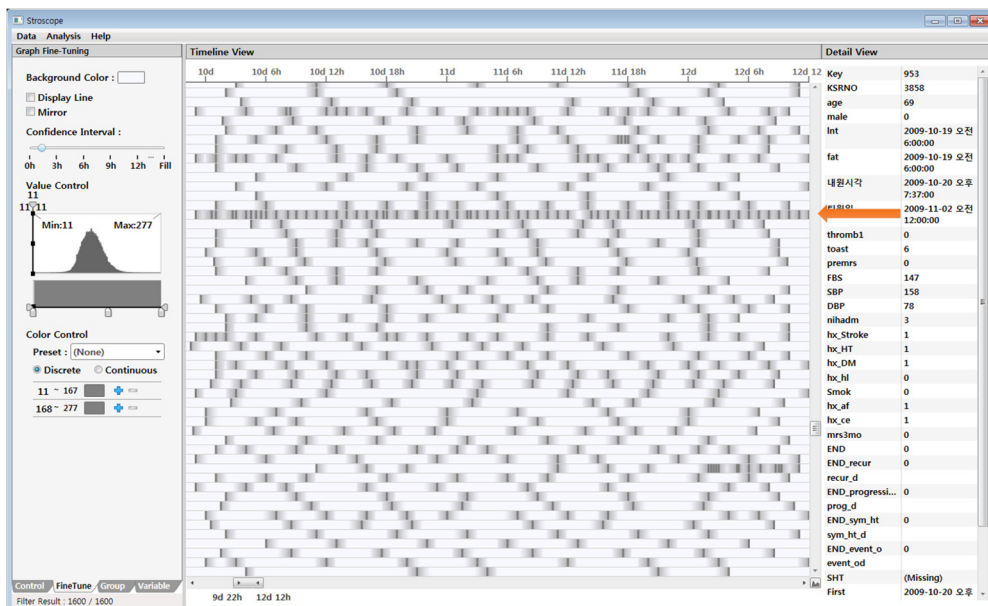


Figure 8.7. Stroscope that all values are represented by the same height and color. An orange arrow indicates a patient with an anomalous pattern because measurement frequency is higher for the patient than others.

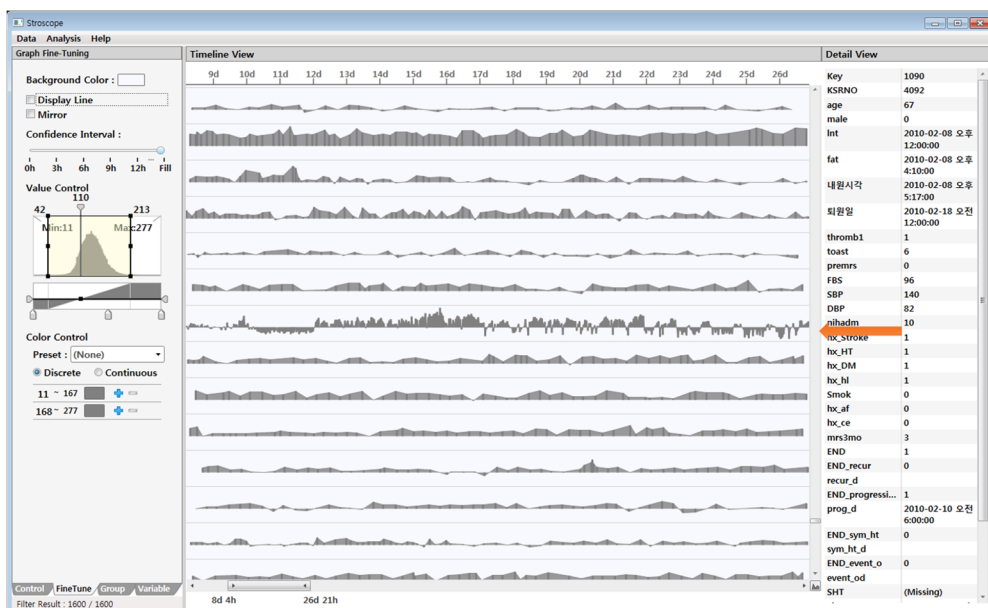


Figure 8.8. Stroscope showing the fluctuation around the value of 110. An orange arrow indicates a patient with an anomalous pattern because of a sharp fluctuation.

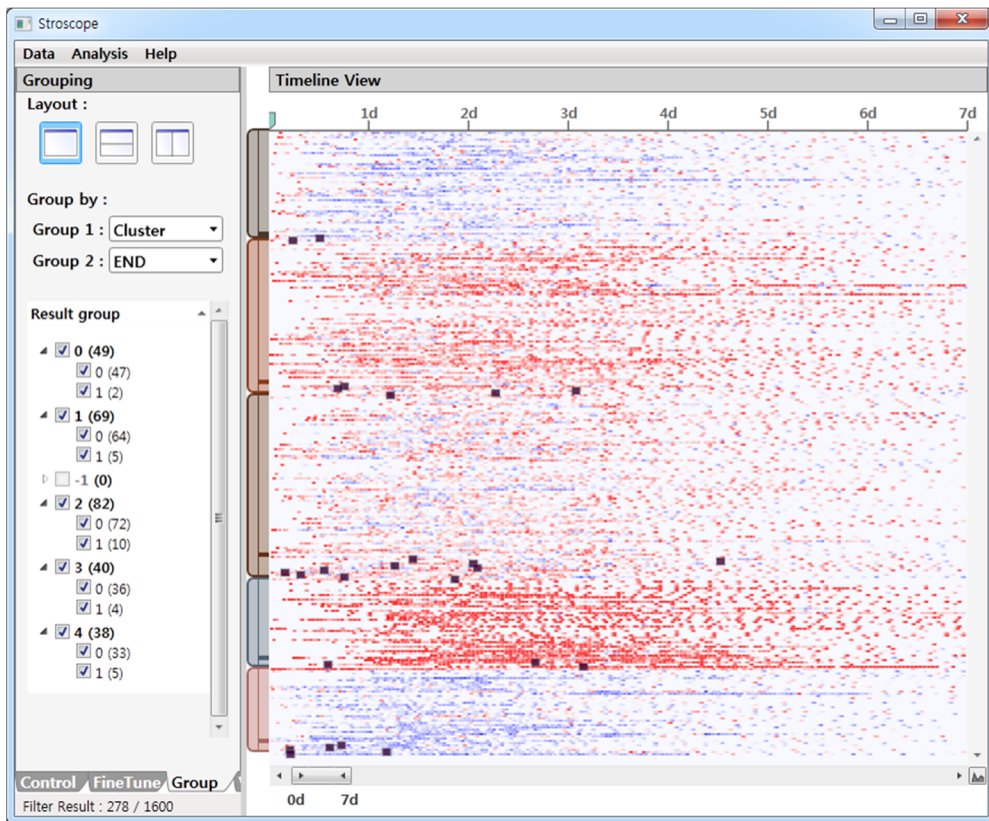


Figure 8.9. Stroscope showing hierarchical grouping result. P2 aligned patients by onset time of stroke and grouped patients according to Cluster and END variables after performing a clustering function. Black rectangular spots in the timeline view represent the END_progression event. In the two clusters with low blood pressure (the first and last clusters), the event tends to occur within about 30 hours from onset.

Using the matching and alignment functions in Stroscope, P2 also figured out that the blood pressure of patients with lacunar infarcts in ‘TOAST 2’ was increasing or decreasing rapidly before and after an END_progression event. Then, she partitioned patients with lacunar infarcts into 5 clusters by performing a clustering

using our enhanced I-kMeans algorithm (section 7.3): two clusters with high blood pressure, two clusters with low blood pressure, and a cluster with slightly high blood pressure. She aligned records by the onset time to check whether there exists any difference among the clusters in the elapsed time from onset to the END_progression event. She observed that the END_progression event occurred within about 30 hours from onset in the clusters with low blood pressure (Figure 8.9). But, the number of patients with the event was not enough to conclude that her observation was meaningful. So she decided to explore in the same manner after collecting more time-series for patients with lacunar infarcts.

8.4 Discussion

Two long-term case studies helped us test the efficacy and utility of Stroscope. Although Stroscope was the two participants' first visualization tool for time-series data, they became rapidly proficient in using it. We allowed them to ask us for help whenever they were faced with any problems. But, they used Stroscope skillfully without any help after the first one month of the case study.

They used sorting, aligning, and matching functions for comparison of two groups, which was one of the main tasks in the analytical exploration of the blood pressure data. They could easily find differences in measurement frequency and variability as well as measurements, especially before and after a specific event. These findings confirmed what they already knew and also yielded the results contradicting previous studies.

They changed the visual representation from a familiar graph such as the line graph or bar graph to their own ripple graph using the interactive widget depending upon their goals and the available display space. To see an intuitive overview of multiple records, they often made all bars have the same height of 1 pixel and then adjusted the color palette and confidence interval (Figure 8.9). For participant 1, to see only the increasing/decreasing periods, he adjusted the position of horizontal axis and ROI values. He was satisfied with that he created his own graph to show the peak only. He could also observe one record in detail by adjusting ROI values after increasing the height of the ripple graph (Figure 8.4). He commented on our visualization model and interactive widget as follows: “It is very nice that I can progressively narrow down to a range of values of my interest after understanding the context.”

Although our multi-scale visualization model enables users to choose the best representation for a given display space, the scalability issue still remains. It can only scale up to a point where each time-series takes a pixel height. It is possible to scale up further by employing aggregation or data reduction techniques, but then we may lead to a more aggregated overview, thus inducing information loss. Such information loss is in general unacceptable in the medical field since it could complicate or mislead medical decision-making.

Another limitation is that we conducted two case studies in only one domain, i.e. medical domain with a blood pressure data. Further case studies are required to show

that Stroscope based on the ripple graph is not a domain-specific tool. Thus, more case studies in other domains can be meaningful future work.

Chapter 9

Conclusion

In this dissertation, we presented a multi-scale visualization model, i.e. a ripple graph for irregularly measured time-series data, concerned with measurement frequency and confidence in values between measurements. For large time-series data, we developed three space-efficient methods in a multi-scale model (i.e. Dimension zooming with range of interest, Color mapping, and moving the horizontal axis). We then evaluated the ripple graph by conducting a usability to verify the usefulness of ripple graph. We also conducted a controlled user study to evaluate the ripple graph in terms of how well people could learn and use it in performing graphical perception tasks, compared to other visualization techniques for time-series data. Results showed that the ripple graph was overall the best performing in terms of task time, correctness, and subjective satisfaction.

To investigate the efficacy and potential of the ripple graph and to promote interactive information visualization techniques in the medical domain, we implemented an interactive visualization tool, called Stroscope, through a collaborative participatory design with neurologists. Stroscope provides an interactive widget to enable intuitive control of the ripple graph and several analytical functions. We then evaluated Stroscope by conducting case studies with two neurologists. They could efficiently exploit the visualization model and the analytical functions of Stroscope throughout their exploratory analysis processes.

In summary, this work's contributions are three-fold: (1) introducing a new representation to enhance the graphical integrity in visual analysis of irregularly measured time-series, (2) seamlessly integrating different existing time-series representations into a unified interface framework for supporting different tasks for exploratory data analysis, and (3) presenting neurologists with an interactive visualization tool for analyzing blood pressure measurement data for stroke study.

Bibliography

- [1] C. Ahlberg, "Spotfire: an information exploration environment," *ACM SIGMOD Record*, vol.25, no.4, pp.25–29, 1996
- [2] C. Ahlberg, B. Shneiderman, "The alphaslider: A compact and rapid selector," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'92)*, pp.365–371, 1994
- [3] C. Ahlberg, C. Williamson, and B. Shneiderman, "Dynamic queries for information exploration: An implementation and evaluation," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'92)*, pp.619–626, 1992
- [4] W. Aigner, S. Miksch, W. Mueller, H. Schumann, and C. Tominski, "Visualizing Time-Oriented Data: A Systematic View," *Computers & Graphics*, vol.31, no.3, pp.401–409, 2007
- [5] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, "Visualization of Time-Oriented Data," *New York, Springer*, 2011
- [6] W. Aigner, C. Kainz, R. Ma, and S. Miksch, "Bertin was Right: An Empirical Evaluation of Indexing to Compare Multivariate Time-Series Data Using Line Plots," *Computer Graphics Forum*, vol.30, no.1, pp.215–228, 2011
- [7] P. Andre, M. Wilson, A. Russell, D.A. Smith, A. Owens, and M.C. Schraefel,

- “Continuum: Designing Timelines for Hierarchies, Relationships and Scale,”
Proc. ACM Symp. User Interface Software and Technology, 2007
- [8] N. Andrienko and G. Andrienko, "Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach," *New York, Springer*, 2006
 - [9] A. Aris, B. Shneiderman, C. Plaisant, G. Shmueli, and W. Jank, "Representing Unevenly-Spaced Time Series Data for Visualization and Interactive Exploration", *Proc. IFIP TC13 international conf. Human-Computer Interaction (INTERACT'05)*, pp.835–846, 2005
 - [10] R. Arias-Hernandez, L.T. Kaastra, T.M. Green, and B. Fisher, "Pair Analytics: Capturing Reasoning Processes in Collaborative Visual Analytics," *Proc. 44th Hawaii International Conf. System Sciences*, 2011
 - [11] R. Bade, S. Schlechtweg, and S. Miksch, "Connecting Time-Oriented Data and Information to a Coherent Interactive Visualization," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'04)*, pp.105–112, 2004
 - [12] R. Baeza-Yates and B. Ribeiro-Neto, “Modern information retrieval,” *New York, ACM press*, 1999
 - [13] E.J. Bergstralh and J.L. Kosanke, "Computerized Matching of Cases to Controls," *Technical Report Serial No. 56, Rochester, MN: Mayo Clinic, Section of Biostatistics*, 1995
 - [14] L. Berry, and T. Munzner, "BinX: Dynamic Exploration of Time Series Datasets across Aggregation Levels," *Proc. IEEE Symp. Information Visualization, Posters Compendium* , pp.215–216, 2004
 - [15] E. Bier, M. Stone, K. Pier, W. Buxton and T. DeRose, “Toolglass and Magic Lenses: the See-Through Interface,” *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'93)*, pp.73–80, 1993

- [16] D. Brodbeck, R. Gasser, M. Degen, S. Reichlin and J. Luthiger, “Enabling Large-Scale Telemedical Disease Management through Interactive Visualization,” *European Notes in Medical Informatics*, vol.1, no.1, pp.1172–1177, 2005
- [17] S.-M. Chan, L. Xiao, J. Gerth, and P. Hanrahan, “Maintaining Interactivity While Exploring Massive Time Series,” *IEEE Symp. Visual Analytics Science and Technology (VAST'08)*, 2008
- [18] ChronoViz at <http://chronoviz.com/>
- [19] C. Dunne, B. Shneiderman, R. Gove, J. Klavans and B. Dorr, “Rapid understanding of scientific paper collections: Integrating statistics, text analytics, and visualization,” *Journal of the American Society for Information Science and Technology*, vol.63, no.12, pp.2351–2369, 2012
- [20] K. Fishkin and M.C. Stone, “Enhanced dynamic queries via movable filters,” *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'95)*, pp.415–420, 1995
- [21] M. Freire, C. Plaisant, B. Shneiderman and J. Golbeck, “ManyNets: an interface for multiple network analysis and visualization,” *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'10)*, pp.213–222, 2010
- [22] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, “Design Patterns: Elements of Reusable Object-Oriented Software,” *Addison-Wesley*, 1994
- [23] T. Gschwandtner, W. Aigner, K. Kaiser, S. Miksch and A. Seyfang, “CareCruiser: Exploring and Visualizing Plans, Events, and Effects Interactively,” *Proc. IEEE Pacific Visualization Symposium*, 2011
- [24] H. Hauser, F. Ledermann, and H. Doleisch, “Angular brushing of extended parallel coordinates”. *Proc. IEEE Symp. Information Visualization*, 2002

- [25] 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," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'09)*, pp.1303–1312, 2009
- [26] H. Hochheiser and B. Shneiderman, "Dynamic query tools for time series data sets: timebox widgets for interactive exploration," *Information Visualization*, vol.3, no.1, pp.1–18, 2004
- [27] W. Javed and N. Elmqvist, "Stack Zooming for Multi-focus Interaction in Time-series Data Visualization," *Proc. IEEE Pacific Visualization Symposium*, 2010
- [28] W. Javed, B. McDonnel, and N. Elmqvist, "Graphical Perception of Multiple Time Series," *IEEE Trans. Visualization and Computer Graphics*, vol.16, no.6, pp.927–934, 2010
- [29] H. Kang, B. Shneiderman, "Visualization methods for personal photo collections: Browsing and searching in the PhotoFinder," *Proc. IEEE International Conf. on Multimedia and Expo 2000*, pp.1539–1542, 2000
- [30] R. Kincaid, "SignalLens: Focus+Context Applied to Electronic Time Series," *IEEE Trans. Visualization and Computer Graphics*, vol.16, no.6, pp.900–907, 2010
- [31] R. Kincaid and H. Lam, "Line Graph Explorer: Scalable Display of Line Graphs Using Focus+Context," *Proc. Working Conf. Advanced Visual Interface (AVI'06)*, pp.404–411, 2006
- [32] M. Krstajic, E. Bertini, D. Keim, "CloudLines: Compact Display of Event Episodes in Multiple Time-Series," *IEEE Trans. Visualization and Computer Graphics*, vol.17, no.12, pp.2432–2439, 2011
- [33] B.C Kwon, B. Fisher, and J.S. Yi, "Visual Analytic Roadblocks for Novice

- Investigators," *IEEE Symp. Visual Analytics Science and Technology (VAST'11)*, 2011
- [34] J. Lin, E. Keogh, and S. Lonardi, "Visualizing and Discovering Non-Trivial Patterns in Large Time Series Databases," *Information Visualization*, vol.4, no.2, pp.61–82, 2005
 - [35] P. McLachlan, T. Munzner, E. Koutsofios, and S. North, "LiveRAC: Interactive Visual Exploration of System Management Time-Series Data," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'08)*, pp.1483–1492, 2008
 - [36] J. Nielson, "Usability 101: Introduction to Usability," at <http://www.useit.com/alertbox/20030825.html>
 - [37] M. Osada, H. Liao, and B. Shneiderman, "AlphaSlider: Searching textual lists with sliders," *Proc. the Ninth Annual Japanese Conf. on Human Interface 1993*, 1993
 - [38] A. Perer, and B. Shneiderman, "Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'08)*. pp.265–274, 2008
 - [39] C. Perin, F. Vernier, and J. Fekete, "Interactive Horizon Graphs: Improving the Compact Visualization of Multiple Time Series," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'13)*, pp.3217–3226, 2013
 - [40] H. Reijner, "The Development of the Horizon Graph," *Proc. Vis08 Workshop From Theory to Practice: Design, Vision and Visualization*, 2008
 - [41] C. Plaisant, V. Jain, "Dynamaps: Dynamic queries on a health statistics atlas," *Conf. on Human Factors in Computing Systems*, pp.439–440, 1994

- [42] A. Rind, W. Aigner, S. Miksch, S. Wiltner, M. Pohl, F. Drexler, B. Neubauer and N. Suchy, "Visually Exploring Multivariate Trends in Patient Cohorts Using Animated Scatter Plots," *Proc. the International Conf. Ergonomics and health aspects of work with computers (EHAWC '11)*, pp.139–148, 2011
- [43] A. Rind, W. Aigner, S. Miksch, S. Wiltner, M. Pohl, T. Turic and F. Drexler, "Visual Exploration of Time-oriented Patient Data for Chronic Diseases: Design Study and Evaluation," *Proc. USAB 2011: Information Quality in e-Health*, pp.301–320, 2011
- [44] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda, "Two-Tone Pseudo Coloring: Compact Visualization for One Dimensional Data," *Proc. IEEE Symp. Information Visualization*, pp.173–180, 2005
- [45] J. Sanyal, S. Zhang, G. Bhattacharya, P. Amburn, and R. J. Moorhead, "A User Study to Compare Four Uncertainty Visualization Methods for 1D and 2D Datasets," *IEEE Trans. Visualization and Computer Graphics*, vol.15, no.6, pp.1209–1218, 2009
- [46] M. Sedlmair, P. Isenberg, D. Baur, M. Mauerer, C. Pigorsch, and A. Butz, "Cardiogram: Visual Analytics for Automotive Engineers," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'11)*, pp.1727–1736, 2011
- [47] J. Seo and B. Shneiderman, "Knowledge Discovery in High-Dimensional Data: Case Studies and a User Survey for the Rank-by-Feature Framework," *IEEE Trans. Visualization and Computer Graphics*, vol.12, no.3, pp.311–322, 2006
- [48] B. Shneiderman, "Dynamic queries for visual information seeking," *IEEE Software*, vol.11, no.6, pp.70–77, 1994
- [49] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," *Proc. IEEE Symp. on Visual Languages 1996*,

pp.336–343, 1996

- [50] B. Shneiderman, C. Ahlberg, “Visual information seeking: Tight coupling of dynamic query filters with starfield displays,” *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'94)*, pp.313–317, 1994
- [51] B. Shneiderman and C. Plaisant. "Strategies for Evaluating Information Visualization Tools: Multi-Dimensional In-Depth Long-Term Case Studies," *Proc. AVI Workshop on Beyond time and errors: novel evaluation methods for information visualization (BELIV'06)*, pp.1–7, 2006
- [52] E.R. Tufte. “The Visual Display of Quantitative Information,” *Graphics Press*, 2001
- [53] S. Thakur and T. Rhyne, "Data Vases : 2D and 3D Plots for Visualization Multiple Time Series," *Proc. International Symp. Visual Computing (ISVC'09)*, pp.929–938, 2009
- [54] J. J. Thomas and K. A. Cook, “Illuminating the Path: The Research and Development Agenda for Visual Analytics,” *IEEE Computer Society Press*, 2005.
- [55] M. Vlachos, J. Lin, E. Keogh, and D. Gunopulos, "A Wavelet-Based Anytime Algorithm for K-Means Clustering of Time Series," *Proc. Workshop on Clustering High Dimensionality Data and Its Applications*, pp.23–30, 2003
- [56] K. Vrotsou, K. Ellegard, and M. Cooper, "Everyday Life Discoveries: Mining and Visualizing Activity Patterns in Social Science Diary Data," *Proc. 11th International Conf. Information Visualization (IV'07)*, pp.130–138, 2007
- [57] K. Vrotsou, J. Johansson, and M Cooper, "ActiviTree: Interactive Visual Exploration of Sequences in Event-Based Data Using Graph Similarity," *IEEE Trans. Visualization and Computer Graphics*, vol.15, no.6, pp.945–952, 2009

- [58] M. Wattenberg, "Sketching a graph to query a time-series database," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems Extended Abstracts*, pp.381–382, 2001
- [59] T.D. Wang, C. Plaisant, B. Shneiderman, N. Spring, D. Roseman, G. Marchand, V. Mukherjee, and M. Smith, "Temporal Summaries: Supporting Temporal Categorical Searching, Aggregation and Comparison," *IEEE Trans. Visualization and Computer Graphics*, vol.15, no.6, pp.1049–1056, 2009
- [60] T.D. Wang, C. Plaisant, A. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman, "Aligning Temporal Data by Sentinel Events: Discovering Patterns in Electronic Health Records," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'08)*, pp.457–466, 2008
- [61] T.D. Wang, K. Wongsuphasawat, C. Plaisant and B. Shneiderman, "Visual information seeking in multiple electronic health records: design recommendations and a process model". *Proc. ACM International Health Informatics Symposium*, pp.46–55, 2010
- [62] C. Williamson, B. Shneiderman, "The dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system," *Proc. the 15th annual international ACM SIGIR Conf. on Research and development in information retrieval*, pp.338–346, 1992
- [63] K. Wongsuphasawat, and B. Shneiderman, "Finding Comparable Temporal Categorical Records: A Similarity Measure with an Interactive Visualization," *IEEE Symp. Visual Analytics Science and Technology (VAST'09)*, 2009
- [64] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman, "LifeFlow: Visualizing an Overview of Event Sequences," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'11)*, pp.1747–1756, 2011

- [65] K. Wongsuphasawat, C. Plaisant, M. Taieb-Maimon and B. Shneiderman, "Querying event sequences by exact match or similarity search: Design and empirical evaluation.," *Interacting with computers*, vol.24, no.2, pp.55–68, 2012
- [66] J. S. Yi, N. Elmqvist and S. Lee, "TimeMatrix: analyzing temporal social networks using interactive matrix-based visualizations", *Intl. Journal of Human–Computer Interaction*, vol.26, no.11–12, pp.1031–1051, 2010
- [67] D. Young and B. Shneiderman, "A graphical filter/flow representation of boolean queries: A prototype implementation and evaluation," *Journal of the American Society for Information Science*, vol.44, no.6, pp.327–339, 1993
- [68] J. Zhao, F. Chevalier, and R. Balakrishnan, "KronoMiner: Using Multi-Foci Navigation for the Visual Exploration of Time-Series Data," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI'11)*, pp.1737–1746, 2011
- [69] H. Ziegler, M. Jenny, T. Gruse, and D. A. Keim, "Visual market sector analysis for financial time series data," *IEEE Symp. Visual Analytics Science and Technology (VAST'10)*, 2010

요약

본 논문에서는 일정한 시간 간격으로 측정된 시계열 데이터뿐만 아니라 불규칙하게 측정된 시계열 데이터를 가시화하기 위한 리플 그래프(ripple graph)를 제안한다. 리플 그래프는 시간의 흐름에 따른 변화를 잘 나타내주는 선 그래프와 데이터의 측정 빈도를 잘 나타내주는 막대 그래프의 장점을 취하여 고안되었다. 또한, 리플 그래프는 실제 측정값은 불투명하게, 측정값 사이의 선형 보간 영역은 값에 대한 확신 정도에 따라 불투명 정도를 서로 다르게 표현함으로써 데이터를 왜곡 없이 표현한다. 더 나아가 많은 수의 시계열 데이터를 한정된 공간에 효율적으로 시각화 하기 위하여 세가지 기법인 (1)관심 영역 설정, (2)칼라 맵핑, (3)시간 축의 이동을 개발하여 리플 그래프에 적용하였다. 또한, 사용자가 이 기법을 유연하게 사용하고 기존의 시계열 데이터의 시각적 표현 방법들을 통합 하기 위해 대화형의 사용자 인터페이스를 고안하였다. 리플 그래프의 효율성을 평가하기 위하여 기존의 대표적인 시계열 데이터 표현 방법인 선 그래프, 막대 그래프, 호라이즌(horizon) 그래프와의 비교 실험을 진행하였다. 그 결과 리플 그래프는 실험에 사용된 모든 태스크에 대하여 수행 시간, 정확도, 만족도 측면에서 가장 좋은 결과를 얻었다.

본 논문에서는 리플 그래프의 효율성과 잠재성을 살펴봄과 동시에 시계열 데이터의 시각적 탐색을 위한 가시화 툴인 Stroscope을 제안한다. Stroscope은 시간적인 관점에서 데이터를 탐색하기 위한 다양한 사용자 인터랙션과 특정 작업을 수행하기 위한 통계적인 분석 방법을 제공한다. Stroscope의 효율성을 보여주기 위해 두 명의 신경과 전문의들과 함께 뇌졸중으로 입원한 환자 1600명의 혈압데이터에 대한 사례 연구를 실시하였다. 두 명의 참가자들은 다수의 환자에 대해 혈압의 변화 패턴을

눈으로 확인하고 관심 영역을 쉽게 탐색해 봄으로써, 기존의 통계 틀을 이용하여 데이터를 분석하는 것에 비해 시간과 노력을 덜 수 있었다. 사례 연구를 통해 참가자들은 (1) 이상이 있는 패턴을 가진 환자 발견, (2) 혈압 값, 혈압 측정 빈도, 혈압의 변동 정도에 따른 특정 두 그룹간의 비교, (3) 이미 알고 있었던 사실에 대한 확인, (4) 새로운 가설의 설정 등의 작업을 용이하게 수행하였음을 발견할 수 있었다.

주요어 : 불규칙적으로 측정된 시계열 데이터, 측정 빈도 가시화, 불확실성 가시화, 사용자 스터디, 장기 사례 연구

학 번 : 2008-30243