A 3D Visualization Technique for Large Scale Time-Varying Data

Maiko Imoto, Takayuki Itoh
Ochanomizu University
{maiko_i, itot}@itolab.is.ocha.ac.jp

Abstract

We represent time-varying data as polyline charts very often. At the same time, we often need to observe hundreds or even thousands of time-varying values in one chart. However, it is often difficult to understand such large-scale time-varying if all the values are drawn in a single polyline chart. This paper presents a polyline-based 3D time-varying data visualization technique. The technique places a set of polylines in the 3D space, where the X-axis denotes time, the Y-axis denotes values, and the polylines are arranged along the Z-axis. It provides two views: the first viewpoint has a view direction along Y-axis, and the second viewpoint has a view direction along Z-axis. The technique displays the overview of the data from the first viewpoint, and the detail of the specific parts of the data from the second viewpoint. It also detects frequent or outlier patterns by applying SAX (Symbolic Aggregate approXimation), and indicates them so that users can discover such characteristic patterns. This paper shows several interesting visualization examples to demonstrate the effectiveness of the presented technique.

1 Introduction

Observation and analysis of time-varying data is important in various scientific, technical, and social fields. We believe information visualization techniques should be useful for observation and analysis of such data. Science researchers often observe time-varying data to clarify the mechanism of natural phenomena. For example, many scientific simulation results construct time-varying volume data, and its visualization is very useful to clarify the mechanism of dynamics or other natural phenomena. Another example is time-varying expression values of genes. Functionality forecasting of genes is an important topic of bioinformatics researchers, and observation of time-varying expression values is a key to clarify their functionality. Many engineers monitor time-varying values of systems on their operations. For example, engineers of atomic plants always look the time-varying measurement values to quickly find accidents of the systems. Similarly, administrators of computers often watch multiple time-varying statistical values so that they can quickly find serious errors or malicious attacks. Moreover, such observation and analysis of time-varying data is important in various social science fields, such as time change of stock prices, and approval ratio of politicians.

We represent time-varying data as polyline charts very often. Also, we commonly draw multiple time-varying values in a single polyline chart space so that users can compare the time-varying values. On the other hand, we often deal with hundreds or even thousands of time-varying values in the above mentioned various fields. It is usually difficult to read if we draw hundreds or thousands of polylines in a single space. Several recent works have addressed the visualization of such large-scale time-varying data, including techniques that enable interactive query of polylines [6] [9] [10], similarity-based discovery of frequent or non-trivial patterns [2] [3] [8], and linked view with a geo-visualization technique [1].

The paper presents a 3D time-varying data visualization technique, for all-in-one display and extraction of similar values. The technique firstly orders the time-varying data based on their similarity, and places the polylines in the order. It then provides two views, while one view briefly represents the whole time-varying data, and the other represents the partial data extracted by click operations. Users can look over the time-varying data, extract groups of interested similar values, and carefully observe the extracted values, by using this technique. Also, the technique detects frequent or outlier patterns by applying SAX (Symbolic Aggregate approXimation), and indicates them by enclosing the corresponding parts of the polylines. This functionality makes users easier to select groups of the polylines which have typical patterns and carefully observe.

This paper introduces our case study using temperature data measured at 913 points in January 2006, and discusses the effectiveness of the presented technique.

2 Related Work

There have been several works on novel polyline chart tools for visualization of time series data sets. Some works focused on interactive query of polylines. Wattenberg et al. presented a sketch-based query interface to search for specific shapes of polylines [10]. Hochheiser et al. presented

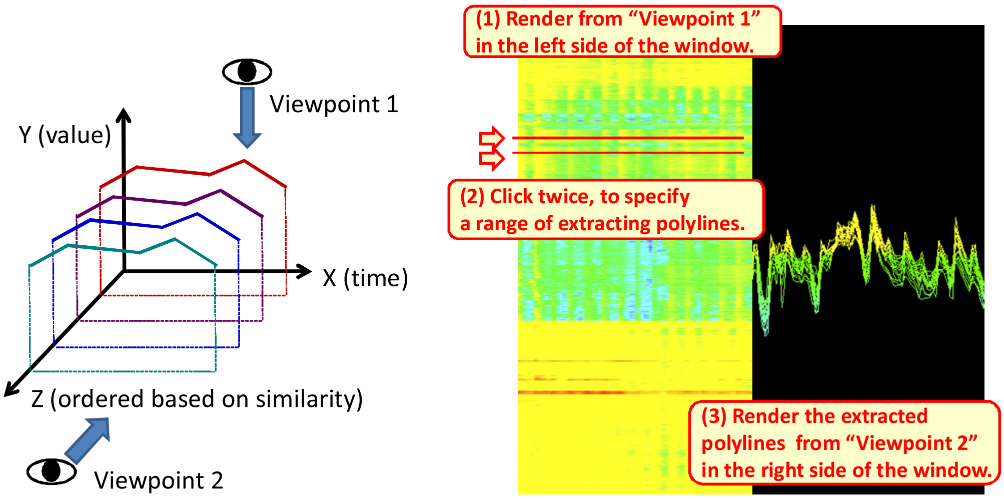

Viewpoint 1
Y (value)
X (time)
Z (ordered based on similarity)
Viewpoint 2
(1) Render from "Viewpoint 1" in the left side of the window.
(2) Click twice, to specify a range of extracting polylines.
(3) Render the extracted polylines from "Viewpoint 2" in the right side of the window.

Figure 1: Outline of the presented technique. (Left) 3D placement of polylines. The technique assigns z-coordinate values to the polylines based on the order. (Right) Processing and operation flow of the presented technique. Left side of the window provides the overview of the data, and user-specified portion of the data is then displayed in the right side of the window.

Timeboxes and TimeSearcher, a gradient- and range-based query interface for polyline-based time-varying data visualization [6]. Some works focused on similarity-based pattern and outlier discovery. Buono et al. presented a technique to interactively search for similar pattern, as an extension of TimeSearcher [2]. They also presented a similarity-based forecasting technique [3], which forecasts the future pattern of time-varying values by searching for similar past patterns. Lin et al. presented a technique to discover non-trivial patterns [8], by clustering a set of time-varying values and searching for outliers. As discussed in Section 1, such interactive query and pattern discovery techniques are useful to zoom into the interested parts or filter the uninterested parts of the time-varying data.

We have presented a polyline-based time-varying data visualization technique [9], which is close to combination and extension of the above existing techniques. The technique applies polylines clustering to reduce of drawing polylines for overview and level of detail control. It also applies sketch interface for query of specific polylines and revival of polylines skipped by the level of detail control.

Several application-specific polyline-based visualization techniques also focused on representation of time-varying values. For example, ThemeRiver [5] visualizes frequency of words in a set of documents as a metaphor of a river.

The presented technique applies SAX [4] [7] for pattern discovery from time-varying data. It divides time-varying values into a set of segments, and replaces the segments into constant values (e.g. mean values). It then replaces the segments into symbolic characters, so that users can easily discover frequent or outlier patterns, and query them. The presented technique appliers the symbolic representation to indicate such patterns so that users can easily discover.

3 Proposed Technique

Most of existing polyline-based time-varying data visualization place sets of polylines onto 2D (xy-place) display spaces. On the other hand, the technique presented in this paper places the polylines onto a 3D space (xyz-space). Figure 1 shows an illustration and an example of the presented technique. It firstly orders the polylines based on the similarity of their feature values, and then assigns z-coordinate values to the polylines according to the order. The placement is useful to extract groups of similar polylines. It then provides two views, while "Viewpoint 1" has a ray perpendicular to xz-plane, and "Viewpoint 2" has a ray perpendicular to xy-plane. The first view provides an overview of the whole time-varying data and a user interface to extract interested polylines, and the second view provides zoomed display of the extracted polylines. The technique also displays rectangles corresponding to frequent or outlier patterns detected by SAX, in the left side of the window.

The technique consists of the following three components:

1. Preprocess: Polyline ordering and pattern detection.

2. Overview of the whole time-varying data in the left side of the window.

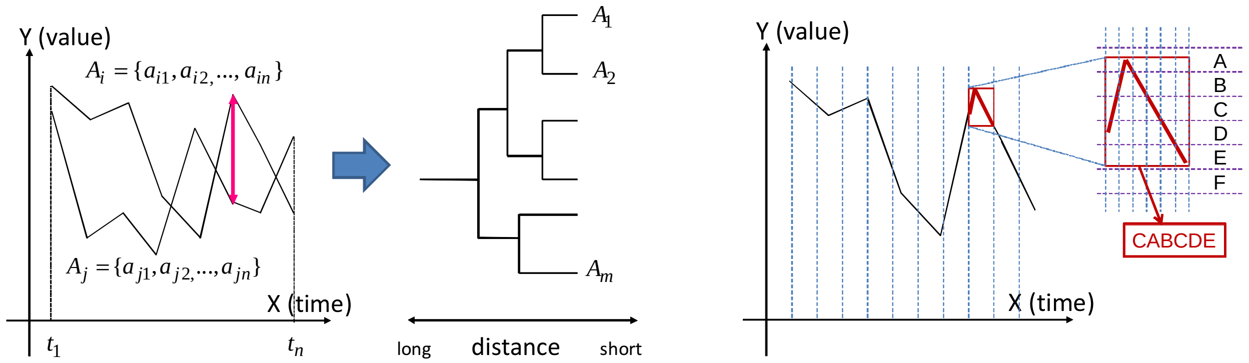


Figure 2: (Left) Construction of dendrogram based on similarity of polylines. The dendrogram is used to order the polylines. (Right) Symbolic character representation of the pieces of time-varying values by SAX.

3. Display of the extracted polylines in the right side of the window.

3.1 Polyline Ordering

This section describes the j-th value at the i-th time as a_{ij}, and the j-th time series values as $A_j = \{a_{j1}, .., a_{jn}\}$, while the data has n time steps. Here, the technique calculates distances between arbitrary pairs of polylines, and constructs a dendrogram of the polyline to order them. Our current implementation prepares several distance calculation schemes so that users can select one of them according to their use scenario. The paper introduces a scheme that calculates distances based on positions of polyline-vertices. The scheme calculates the distance between the j-th and k-th polylines $D(j,k)$ by the following equation:

$$D(j,k) = \sum_{i=1}^{n} \sqrt{(a_{ij} - a_{ik})^2} \qquad (1)$$

After calculating the distances between all the possible pairs of polylines, the scheme constructs a dendrogram. Initially the scheme treats a polyline as a cluster, and merges the closest clusters into a single cluster. The merge process is repeated until all the polylines are connected by a single dendrogram. Finally, the scheme orders all the polylines by using the dendrogram. Figure 2 (Left) shows an illustration of the above mentioned process.

3.2 Overview

The technique prepares two viewpoints as above mentioned. Figure 1 (Left) shows an illustration of the polyline placement in a 3D space by the technique.

These are used to global and local visualization of the time-varying data. The technique firstly displays the whole data from ”Viewpoint 1”, in the left side of the window, as shown in Figure 1(Right)(1). Here all the polylines are drawn as ribbons, while they are shaded after setting several light sources. The polylines are colored according to y-coordinate values of the polyline-vertices: our current implementation assigns warm colors (e.g. red) to higher y-coordinate values and cool colors (e.g. blue) to lower y-coordinate values. The technique initially adjusts ”Viewpoint 1” to display the whole data in one space; however, users can control the viewpoint and freely focus on specific parts of the data.

3.3 Polyline Selection

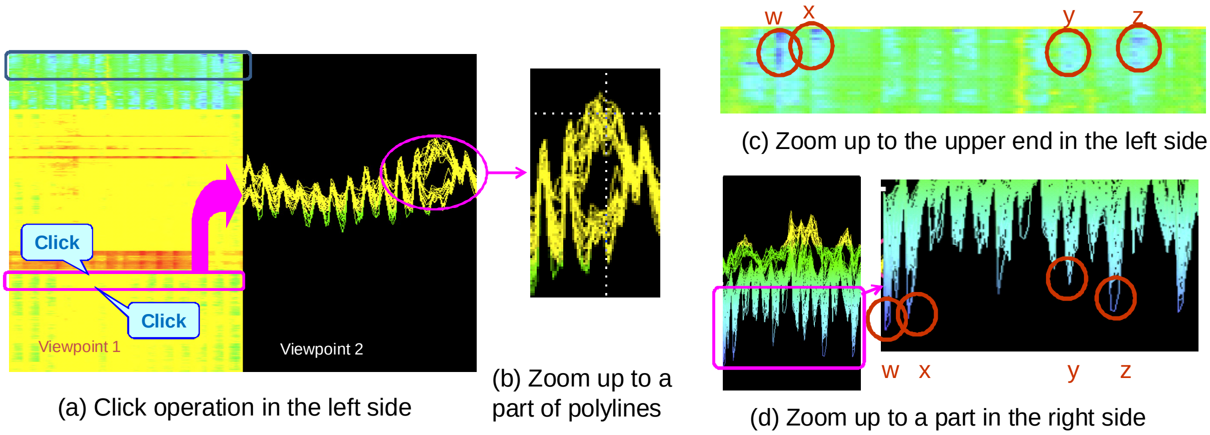
The technique also provides a user interface to extract specific similar polylines on the left side of the window. Our implementation supports two-click operation to specify a range of extracting polylines, as shown in Figure 1(Right)(2). When a user specifies a range, the technique renders the extracted polylines from ”Viewpoint 2” and displays on the right side of the window, as shown in Figure 1(Right)(3). Again, the technique initially adjusts the viewpoint to display the whole part of the extracted polylines; however, users can freely control the viewpoint to focus on specific parts of the extracted polylines.

As above mentioned, the technique supports dual display of the time-varying data, so that users can initially overview the whole data, extract interesting similar polylines, and focus on the extracted polylines to carefully observe their coherency or differences.

It is often effective to assign another color in the right side, by calculating from other values or metadata of the polylines. We support the coloring according to the gradients of polylines as one of the variety of color calculation. It can effectively emphasize suddenly changing values.

3.4 Pattern Extraction and Indication

The technique is generally useful to extract a set of totally similar time-varying values in large-scale datasets, because they are ordered based on their global similarity. On the other hand, we are often interested in extraction of partially characteristic values. We apply SAX [4] [7] for symbolic character representation of time-varying values,



(a) Click operation in the left side

(b) Zoom up to a part of polylines

(c) Zoom up to the upper end in the left side

(d) Zoom up to a part in the right side

Figure 3: Example (1). Users can understand the overview of the time-varying data by looking at the left side, and the detailed trends of specific parts of the data by looking at the right side.

which is useful for various purposes, including discovery of frequent or outlier patterns, and query of specific patterns.

Figure 2(Right) is an illustration of our implementation of SAX. The implementation divides time-varying values into pieces by a constant interval N_c, as

$$A_j = \{(a_{j1}, ..., a_{jN_c}), (a_{j(N_c+1)}, ..., a_{j(2N_c)}), \\ ..., (a_{j(n-N_c+1)}, ..., a_{jn})\} \qquad (2)$$

At the same time, the implementation assigns characters to ranges of the values. Figure 2(Right) shows that it assigns A, B, C, D, E, and F to the illustrated ranges. It then divides each interval into smaller pieces by a smaller interval, calculates average values of the piece, and finally assigns characters to the pieces. Figure 2(Right) zooms in an interval $(a_{j(kN_c+1)}, ..., a_{j((k+1)N_c)})$, where the implementation divides the interval into six pieces and assigns characters "CABCDE". This section calls such sequences of characters "words".

After dividing all the time-varying values and constructing words, the implementation counts the frequency of the words. The technique can indicate time-varying values corresponding to the specific words in the left side of the window. Our current implementation supports to display:

- One or more words which has extremely high frequency, to indicate major patterns.
- One or more words which has extremely low frequency, to indicate outlier patterns.

4 Experiment

This section introduces a case study that visualized temperature data recorded in January 2006 at 913 points in Japan by a weather measurement system AMeDAS. We implemented the technique with Java JDK 1.6.0, and tested on an Apple MacBook Pro (CPU 2.2GHz, RAM 4GB) running Mac OS 10.4.

4.1 Operation Flow

Figure 3 shows an example of the operation flow. We clicked a part of the data in the left side, as marked by a red rectangle in Figure 3(a). We focused this part because its color changed between yellow and green very frequently. We then observed the specified polylines in the right side of the window. The temperatures were totally less-varying between daytime and night, but actually changed between yellow and green almost every day. Most of the polyline looked totally very similar; however, they were very different in a few days, as marked by a pink circle. It was difficult to recognize such local difference in the left side, but we found it by carefully observing in the right side, as shown in Figure 3(b). The example denotes that the mechanism of our technique is effective, to overview the whole data first, and then extracts parts of the data to carefully observe. Another interesting point is that the extracted polylines corresponded to temperatures at different regions of Japan, including Ibaraki prefecture (along Pacific Ocean), Ishikawa prefecture (along Japan Sea), and Nara prefecture (inland). Usually we do not think such different regions had similar variation of temperature. The example also denotes that we discovered interesting and unexpected phenomena by the presented technique.

We clicked another part of the data in the left side, as marked by a blue rectangle. Before clicking the part, we observed several deep blue parts marked by brown circles in Figure 3(c). We clicked this part and observed the polylines as shown in Figure 3(d), and found the differences

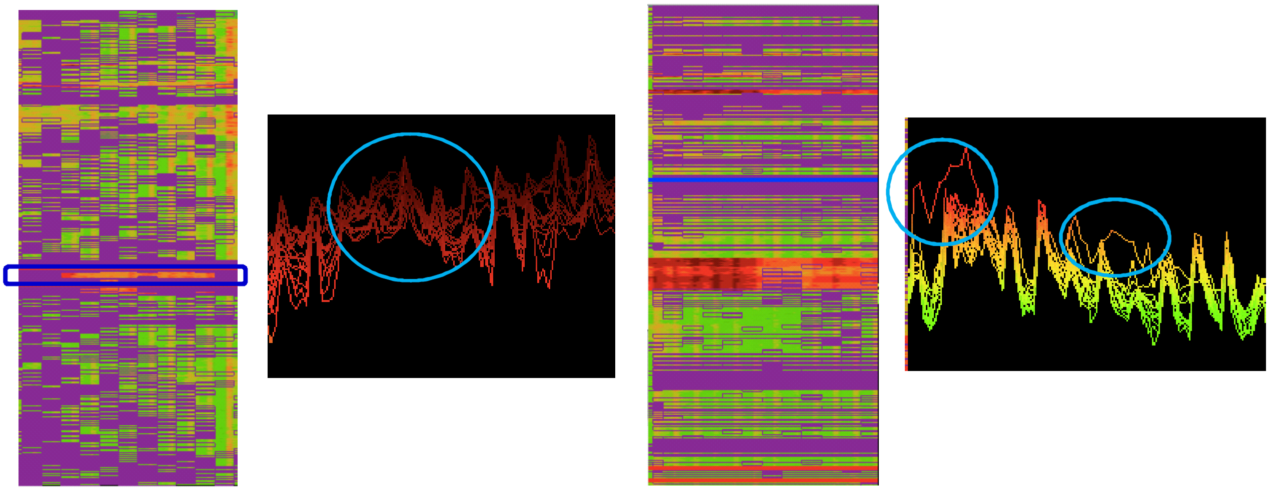


Figure 4: Example (2). (Left) Frequent patterns are enclosed by purple rectangles. (Right) Outlier patterns are enclosed by purple rectangles.

among the marked parts. This result shows that truly low temperature was observed in many points at the time ”w”. On the other hand, such low temperature was observed in a few points at the time ”z”. We did not think such differences before the click operation because we are not enough sensitive to the slight differences of colors; however, we could understand such slight differences after the click operation.

4.2 Frequent or Outlier Patterns

Figure 4 shows examples of frequent or outlier pattern discovery applying SAX. We divided the temperature data in January 2006 into 31 pieces (day-by-day), and consequently dealt with $931 \times 31 = 28861$ words by assigning 7 characters.

Figure 4(Left) shows frequent patterns by enclosing corresponding words by purple rectangles in the left side of the window. Our implementation drew rectangles if the words appeared 300 times or more. Here, we observed a part marked by a blue rectangle, where the purple rectangles were not drawn for several days. We clicked the part and observed the polylines in the right side, and found that the temperature was relatively high and flat all the several days, as marked by a sky blue circle. We understood that temperatures are usually more fluctuating in these points, and such flat temperature marked by the sky blue circle is not a frequent phenomenon. We could not understand this kind of numerical features only by looking at colored polylines, but we could discover them by marking frequent patterns.

Figure 4(Right) shows outlier patterns by enclosing corresponding words by purple rectangles in the left side of the window. Our implementation drew rectangles if the words appeared less than 10 times. We clicked a part that many purple rectangles were drawn, and discovered outlier polylines marked by sky blue circles. They are truly rare patterns that were not synchronized with temperatures at other points. This example demonstrates that our SAX implementation contributed to discover such outlier patterns easily.

Figure 5 shows examples that just one of the frequent patterns is indicated by purple rectangles. Here, we clicked two parts enclosed by blue and red rectangles in the left side of the window, and observed the polylines. We observed that many polylines were enclosed by purple rectangles twice, as marked by two yellow rectangles. We displayed the polylines in the two modes: the former mode colored polylines in blue and red based on the click operations, and the latter mode colored polylines based on temperature values. We found that polylines in red did not range so much in the period marked by the second yellow rectangle, though blue polylines ranged so much at that time. This example denotes that we discovered two groups of partially similar polylines which are separately placed in the left side of the window, and understood how the two groups were similar and different thanks to the specific pattern indication.

5 Conclusion and Future Work

The paper presented a polyline-based 3D time-varying data visualization technique. The technique first orders time-varying values based on their similarity. It then places the time-varying values as polylines in the 3D space, where the X-axis denotes time, the Y-axis denotes values, and the polylines are arranged along the Z-axis. It provides two views: one view displays the overview of the data, and the

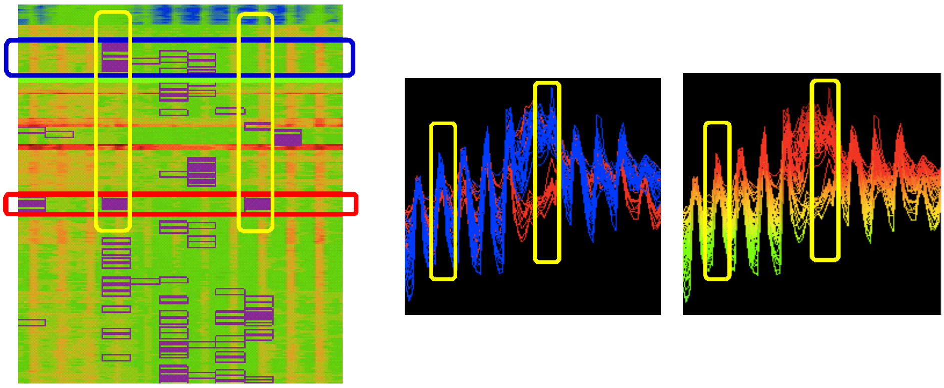


Figure 5: Example (3). Just one of the frequent patterns is indicated in the left side of the window.

other view displays finer representation of the specified set of time-varying values. It also displays frequent or outlier patterns so that users can easily discover major or minor trends of the data. This paper presented an interesting case study with the temperature data recorded by Japanese weather measurement system.

Our potential future work includes the following issues:

Validation of the polyline ordering algorithm is an important issue. We aim to gather similar polylines in a single region in the left side of the visualization, but it often causes that similar polylines separately place and form multiple clusters. We would like to solve the issue.

We would like to have scalability tests soon. Our test data contained 913 sets of time-varying values. We did not have any difficulty while observing and operating the data by the technique, but we are not sure if it is still comfortable to use the technique with ten-thousands of time-varying values. We would like to test the technique with larger datasets. We think it is also important to evaluate with experimental users, and have experiments with other time-varying data.

This work has been partially supported by Japan Society of the Promotion of Science under Grant-in-Aid fir Scientific Research.

References

[1] G. Andrienko, N. Andrienko, Visual Exploration of the Spatial Distribution of Temporal Behaviors, *9th International Conference on Information Visualization*, 799-806, 2005.

[2] P. Buono, A. Aris, C. Plaisant, A. Khella, B. Shneiderman, Interactive Pattern Search in Time Series, *Conference on Visualization and Data Analysis '05*, 175-186, 2005.

[3] P. Buono, C. Plaisant, A. Simeone, A. Aris, B. Shneiderman, G. Shmueli, W. Jank, Similarity-Based Forecasting with Simultaneous Previews: A River Plot Interface for Time Series Forecasting, *11th International Conference on Information Visualization*, 191-196. 2007.

[4] S. Gandhi, S. Nath, S. Suri, J. Liu, GAMPS: Compressing Multi Sensor Data by Grouping and Amplitude Scaling, *35th SIGMOD International Conference on Management of Data*, pp. 771-784, 2009.

[5] S. Havre, E. Hetzler, P. Whitney, L. Nowell, ThemeRiver: Visualizing Thematic Changes in Large Document Collections, *IEEE Transactions on Visualization and Computer Graphics*, 8(1), 9-20, 2002.

[6] H. Hochheiser, B. Shneiderman, Dynamic Query Tools for Time Series Data Sets: Timebox Widgets for Interactive Exploration, *Information Visualization*, 3(1), 1-18, 2004.

[7] J. Lin, E. Keogh, S. Lonardi, B. Chiu, A Symbolic Representation of Time Series, with Implications for Streaming Algorithms, *8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003.

[8] J. Lin, E. Keogh, S. Lonardi, Visualizing and Discovering Non-trivial Patterns in Large Time Series Databases, *Information Visualization*, 4(2), 61-82, 2005.

[9] Y. Uchida, T. Itoh, A Visualization and Level-of-Detail Control Technique for Large Scale Time Series Data, *13th International Conference on Information Visualisation (IV09)*, pp. 80-85, 2009.

[10] M. Wattenberg, D. Jones, Sketching a Graph to Query a Time-Series Database, *SIGCHI Conference on Human Factors in Computing Systems Extended Abstract (CHI2001)*, 381-382, 2001.