A Visualization and Level-of-Detail Control Technique for Large Scale Time Series Data

Yumiko Uchida, Takayuki Itoh
Ochanomizu University
{yumi-ko, itot}@itolab.is.ocha.ac.jp

Abstract

We have various interesting time series data in our daily life, such as weather data (e.g., temperature and air pressure) and stock prices. Polyline chart is one of the most common ways to represent such time series data. We often draw multiple polylines in one space to compare the time variation of multiple values. However, it is often difficult to read the values if the number of polylines gets larger. This paper presents a technique for visualization and level-of-detail control of large number of time series data. The technique generates clusters of time series values, and selects representative values for each cluster, as a preprocessing. The technique then draws the representative values as polylines. It also provides a user interface so that users can interactively select interesting representatives, and explore the time series values which belong to the clusters of the representatives.

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.

We represent time-varying data as polyline charts almost always. 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 [8] [14], similarity-based discovery of frequent or non-trivial patterns [2] [3] [11], and linked view with a geo-visualization technique [1].

Visualization Information Seeking Mantra [13] is a famous guideline for various information visualization techniques, and we believe it should be also useful as a guideline for polyline-based time-varying data visualization techniques. Aforementioned interactive query and pattern discovery techniques are useful to zoom into the interested parts or filter the uninterested parts of the time-varying data. Linked view techniques are useful for detail-on-demand; for example, geographic visualization techniques can represent the geographic features of time-varying data, when a user specifies an interesting time from the data. On the other hand, we think that overview of time-varying data should be still improved. It is desirable that we can clearly visualize all typical patterns of the time-varying values in one display space.

This paper proposes a polyline-based time-varying data visualization technique which realizes the operations based on the mantra. It consists of the following components:

Overview: The technique selects representative polylines to be drawn, so that it improves the readability by reducing the number of drawing polylines, while it does not miss features of the data. It is useful to overview the data. The technique flexibly controls the number of representative polylines for the level of detail control.

Zoom and filter: The technique provides a sketch interface to interactively query the polylines. It highlights the specified polylines, and inconspicuously draws others. The interface also works to revive the polylines skipped by the representative polyline selection. It is useful to filter unnecessary parts of the data, and zoom and look all the polylines which have interested features.

Detail on demand with linked view: The technique provides a user interface for linked view with other visualization techniques. Currently we support the linked view with our own hierarchical data visualization technique [10] to represent hierarchical values at the specific time. It is useful for detail on demand of the specified polylines or times.

One of the main features of the presented technique is reduction of displayed polylines for overview, and the revival of the skipped polylines for zoom and filter. The mechanism is semantically similar to polygon reduction for level of detail control, used in realistic or interactive computer graphics techniques. It initially draws the reduced number of polygons to represent overall of the scene, and revives the simplified polygons according to zoom in operations. Similarly, the presented technique initially draws the reduced number of polylines, while it improves the readability and frame rates to look overall the data. The technique then revives the skipped polylines according to the sketch operations, which works to zoom into the interested parts, and filter the uninterested parts.

2 Related Work

This section introduces existing time-varying data visualization. Since our proposed technique focuses on cluttering reduction of jammed polylines, this section also introduces existing readability improvement techniques for jammed lines. Finally, this section introduces linked view with time-varying data visualization techniques.

2.1 Time-varying Data Visualization

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 [14]. Hochheiser et al. presented Timeboxes and TimeSearcher, a gradient- and range-based query interface for polyline-based time-varying data visualization [8].

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 [11], 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.

Our technique is close to combination and extension of the above existing techniques. The presented technique applies clustering of polylines; however, the purpose is not search of specific patterns but reduction of drawing polylines for overview and level of detail control. It also applies sketch interface for query of specific polylines; however, the purpose includes revival of polylines skipped by the level of detail control.

Several application-specific visualization techniques also focused on representation of time-varying values. For example, ThemeRiver [7] visualizes frequency of words in a set of documents as a metaphor of a river.

2.2 Jammed Line Problem for Visualization

Readability of jammed lines is a common problem of information visualization techniques, and several works have addressed the problem. We are inspired by such works while developing the technique presented in this paper.

Several Parallel Coordinates techniques resolved the problem by applying clustering algorithms. Fua et al. presented a technique to cluster polylines of parallel coordinates [6], by simply applying hierarchical clustering algorithm to filter uninterested polylines. The technique also provided brushing interface to interactively filter the polylines. Ellis et al. presented a series of works on cluttering reduction for parallel coordinates. Firstly they proved that polylines reduction can be effective for cluttering reduction even if they are randomly sampled [4], and then they proposed a technique to measure cluttering and density of polylines for automatic and adequate sampling [5].

In graph drawing area, readability of jammed edges is an important problem. Usually graph drawing algorithms consider of positions of nodes so that they can reduce edge-crossing; however, they do not flexibly control the positions if nodes have constrains such as orders or hierarchy. Holten's technique addressed this problem, by bundling edges of hierarchical graph data [9]. The technique draws edges as Spline curves; their end control points are the nodes, and intermediate control points are common parent nodes. It looks that the curved edges concentrate at their centers, and fan out at their ends.

2.3 Linked View with Time-Varying Data

We may need to develop a linked view system with the time-varying data visualization techniques, if the time-varying values have additional information, such as tree or graph structure, and geographic data.

There are several recent works focusing on linked or integrated view of time-varying and other data visualization techniques. Andrienko et al. presented a visualization system for spatio-temporal data [1] which links geographic and time-varying visualization techniques. Saraiya et al. presented a visualization technique that maps time-varying data onto graph visualization [12].

This paper presents a linked view between the presented technique and our own hierarchical data visualization technique [10]. The hierarchical data visualization technique represents a hierarchy as nested rectangles, and leaf-nodes as painted icons, satisfying the following conditions: 1) never overlaps the leaf-nodes and branch-nodes in a single hierarchy of other nodes, 2) attempts to minimize the display area requirement, 3) draws all leaf-nodes by equally shaped and sized icons, and 4) attempts to minimize aspect ratio and area of rectangular subspaces.

3 Proposed Technique

3.1 Technical Overview

This paper supposes the following time series data, consisting of a set of values $P = (p_1, p_2, ..., p_n)$ represented as n polylines. We describe the values of a polyline as $p_i = (p_{i1}, p_{i2}, ..., p_{im})$; p_{ij} denotes the value at the j-th

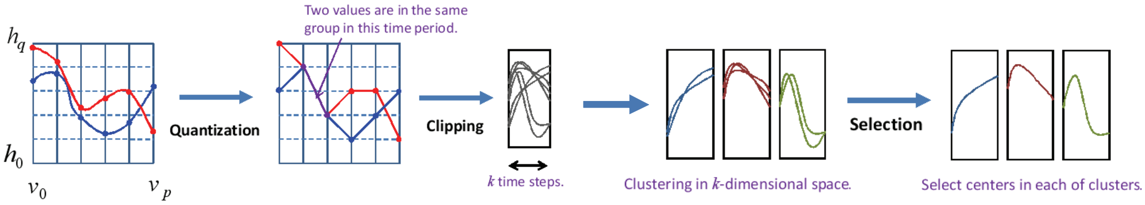

Figure 1: Overview of the preprocessing. (Left) Grouping of polylines according to the quantized values, for each interval of the sampled times. (Right) Clustering of the grouped polylines, and selection of representative polylines for each cluster.

time of i-th polyline. We draw the set of values as a polyline chart, while the horizontal axis denotes the 1st to the m-th time, and the vertical axis denote the magnitude of the values.

The technique consists of the following components:

1. Preprocessing: This component constructs clusters of polylines, and then draws representative polylines of the clusters.

2. Interactive display: This component provides a sketch interface to interactively query the polylines.

3. Linked view: This component provides an interface to deliver the specified set of polylines or times to other views. In this paper we show an example of linked view with our own hierarchical data visualization technique.

The presented technique first generates a grid surrounding the drawing area, and clusters the polylines using the grid. It then selects representative polylines for each cluster, and draws the selected representatives. This approach drastically eases up the jam of polylines and provides the simplified view of the time series data, without missing its interesting features. The presented technique also provides a user interface so that users can interactively select interesting representative polylines, by directly clicking, or drawing their shapes. Specifying the representative polylines, the technique draws non-representative polylines belonging to the clusters of the specified representatives.

3.2 Preprocessing: Clustering and Representative Polyline Selection

Our technique reduces the number of polylines to improve the readability, while it does not miss features of the time-varying data. The technique selects adequate number of polylines as representatives, and skips others, satisfying the following strategies:

Strategy 1: The technique clips the polylines interval by interval, and then generates clusters of clipped polylines for each interval, so that it skips to draw similar patterns.

Strategy 2: The technique adequately selects a polyline for each cluster, so that it can reduce the total number of drawing representative polylines.

Figure 1 shows the overview of the preprocessing for selection of representative polylines. The technique first generates a grid covering the drawing area, and then divides into $p \times q$ subspaces. Here this paper formalizes the grid as follows:

- h_i is the i-th horizontal line of the grid ($0 \leq i \leq q$).
- v_i is the i-th vertical line of the grid ($0 \leq i \leq p$).
- t_i is the time at v_i.
- q_i is the value at h_i.

The technique first samples P at t_0 to t_p, and temporarily quantizes the sampled values at q_0 to q_q. The technique then generates groups of polylines, if the polylines have the same quantized values both at t_{i-1} and t_i. Consequently, the technique generates $q(p+1)^2$ groups, while it may generate empty groups.

Procedure for polyline clustering is as follows. The technique first clips the polylines by t_{i-1} and t_i, and then simply calculates the distances of all the possible pairs of the polylines in a group. If the clipped polylines contain k time steps between t_{i-1} and t_i, the technique regards them as k-dimensional vectors, and calculates Euclidian distances among them. The technique then constructs a dendrogram from the polylines in a group according to their distances. Finally, it constructs clusters of polylines in a group by dividing based on predefined thresholds.

Next, the technique selects representative polylines, for each cluster. Our current implementation simply extracts a polyline as the representative, which is the closest to the center of a cluster in a k-dimensional vector space. This strategy is basically good because it selects average polylines. Here, if one or more polylines in a cluster have been already selected as the representative polylines of other clusters, the technique does not select any new representative from the current cluster, so that we can reduce the total number of representative polylines.

The representative selection result depends on the order of process of clusters. Our implementation applies the following orders: 1) in the order of importance of times, and 2) in the order of number of polylines in the clusters.

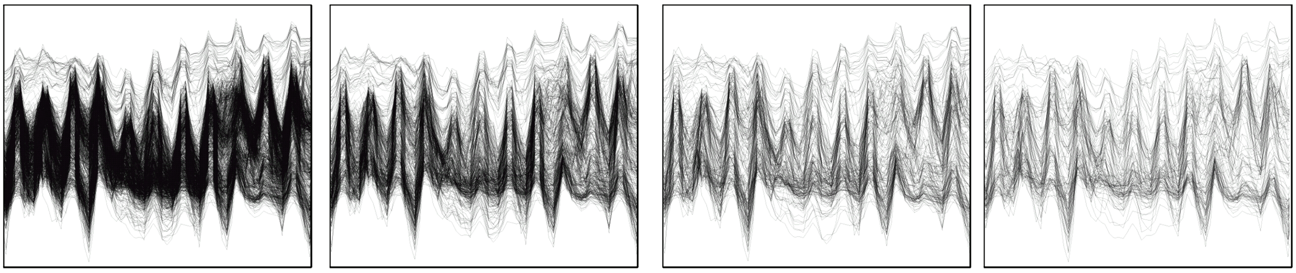

Figure 2: Level-of-detail control. The left figure shows the all polylines, and other three figures show the different numbers of representative polylines.

3.3 Visualization and Level-of-Detail Control

Initially our technique draws only the representative polylines. Here, number of representatives depends on the resolution of the grid and the threshold of the clustering process. Our current implementation generates several clustering results, with several configurations of the grid and the clustering process. Smoothly replacing the clustering results, our technique seamlessly displays several levels of numbers of representatives, as if typical polygon-based computer graphics techniques switch several levels of polygon models for the level-of-detail control. Figure 2 shows an example of level-of-detail control.

3.4 Click and Sketch Interface

The technique provides a user interface to interactively select the interesting representatives. Figure 3 shows examples of the visualization results using the click/sketch interface of the presented technique.

The technique provides a click interface, so that users can specify interesting representatives by directly clicking. When a user clicks a point on the display, the technique calculates distances between the point and all segments of the drawn polylines. If at least one of the segments of a polyline is enough close to the clicked point, the technique highlights the current polyline.

The technique also provides a sketch interface, so that users can specify interesting representatives which have partial shapes similar to the sketched curves. When a user draws a curve on the display, the technique samples several points on the curve, and calculates distances between the sampled points and all segments of the drawn polylines. If at least one of the segments of a polyline is enough close to each of the sampled points, the technique highlights the current polyline.

While polyline reduction in our technique improves the readability of the data, users may want to look all the polylines that have the interested features. To satisfy such requirement, the technique can reactivate the non-representative polylines, which belong to the clusters of the representative polylines specified by click/sketch operations. Figure 3(3) shows the examples.

Users may want to combine multiple click/sketch operations. For example, users may want to visually compare multiple polylines specified by multiple click/sketch operations. Or, users may want to specify polylines by logical operation of the polylines specified by multiple click/sketch operations. To satisfy such requirements, the technique can assign different colors to the polylines specified by the different operations. Also, the technique can assign independent colors to the polylines satisfying the logical operation of the polylines specified by multiple click/sketch operations. Figure 3(4) shows examples of the visualization results using multiple colors.

3.5 Linked View

We can assume that a set of time-varying values forms a structure such as tree or graph. In this case it is useful to develop a linked view between time-varying and other visualization techniques. Our presented technique therefore supports an interface to link it with other visualization techniques. The interface outputs a list of highlighted polylines when a user clicks or sketches them, and calls a redraw function of the linked visualization system. In this paper we suppose that the sets of time-varying values forms trees, and show a linked view which integrates with our own hierarchical data visualization technique [10].

4 Results

We developed the presented technique with JDK (Java Development Kit) 1.6, and executed on a personal computer (CPU 2.2GHz, RAM 2.0GB) with Windows Vista. This section discusses the processing time of the technique, and an application of the technique for the visualization of Japanese weather data.

Processing time of the technique is estimated as follows. The time for preprocessing is $O(n(a+s))$ and for sketch interface is $O(rq)$. Here, n is the total number of polylines, a is the number of sampled time steps, s is the total number of time steps, r is the number of representative polylines,

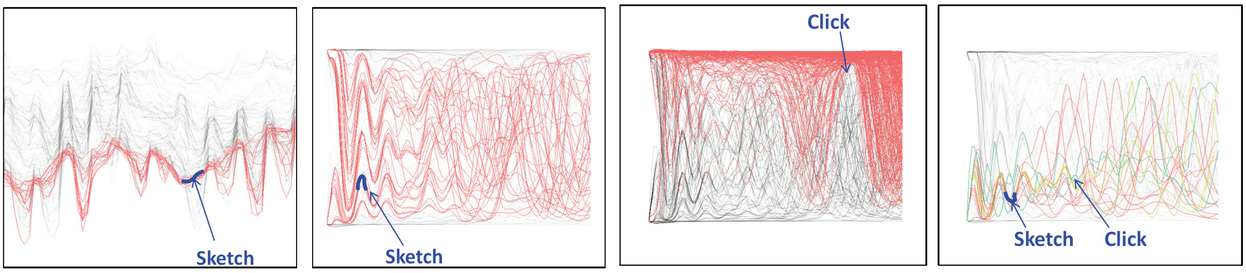

Figure 3: Sketch interface. (1) Time-varying values which have partial shapes similar to the sketched curve, and pass near the sketched region, are highlighted. (2) Time-varying values which have partial shapes similar to the sketched curve are highlighted, even if they are far away from the sketched region. (3) Non-representative polylines which belong to the clusters of sketched representative polylines are also highlighted. (4) Highlighting by multiple colors. In this example, values specified by a sketch operation are highlighted in red, and ones specified by a click operation are in green. Values specified by the both operations are in yellow.

and q is the number of vertices of a sketched curve. In our measurement, the time for preprocessing was 250 milliseconds, and the average of sketch process was 14 milliseconds, which n=913, a=15, s=372, r=419, and q=30. This result denotes that the technique is enough fast for the interactive visualization of our test data.

We applied Japanese weather data recorded by AMeDAS (Automated Meteorological Data Acquisition System) to the presented technique. We extracted time-varying temperature data observed at 913 points in every 2 hours, and clustered them based on the characteristics of the weather. Figure 4(1) denotes the variation of the temperature at 913 points on several days in March 2006. We applied the level-of-detail control as shown in Figure 4(2). A region in the figure indicated by a red circle denotes that temperatures widely vary at many points, and slightly vary at other points. We can suppose that weather was fine at the former points, because the temperature rose in the daytime, and declined in the night. We can also suppose that weather was bad at the latter points, because the temperature did not rise in the daytime. We can discover such feature by the effort of level-of-detail control.

Here we sketched the polylines which denote that temperature slightly changed. We sketched three parts of such polylines: lower, intermediate, and higher, as shown in Figure 4(3)(4)(5). We then confirmed the corresponding regions by highlighting bar charts in the linked hierarchical data view. From the hierarchical data view, we found that the three sets of the sketched polylines were temperatures of the points along Japan Sea. Similarly, we sketched the polylines which denote that temperature widely changed, as shown in Figure 4(6), and confirmed that the corresponding regions that the sketched polylines were temperatures of the points along Pacific Ocean. We checked the weather report in March 2006, and confirmed that the above observation was correct. At that time there was a cold front in the Japan Sea, and therefore weather was bad there. On the other hand, weather was fine along Pacific Ocean, and therefore temperatures widely rose in the daytime and declined in the night there.

5 Conclusion and Future Works

This paper presented a technique for polyline-based visualization of large number of time series values. It adaptively controls the number of displaying representative polylines by a clustering algorithm, so that it can improve the readability without losing interesting features. The paper also presented a sketch interface and linked view with the visualization technique, and introduced the effectiveness with the application to the visualization of weather data in Japan.

The following issues will be our future works:

- Applications to various time series data.
- Subjective evaluation by user experiments.
- Integration and linkage with other visualization techniques, such as time-varying volume visualization techniques.

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.

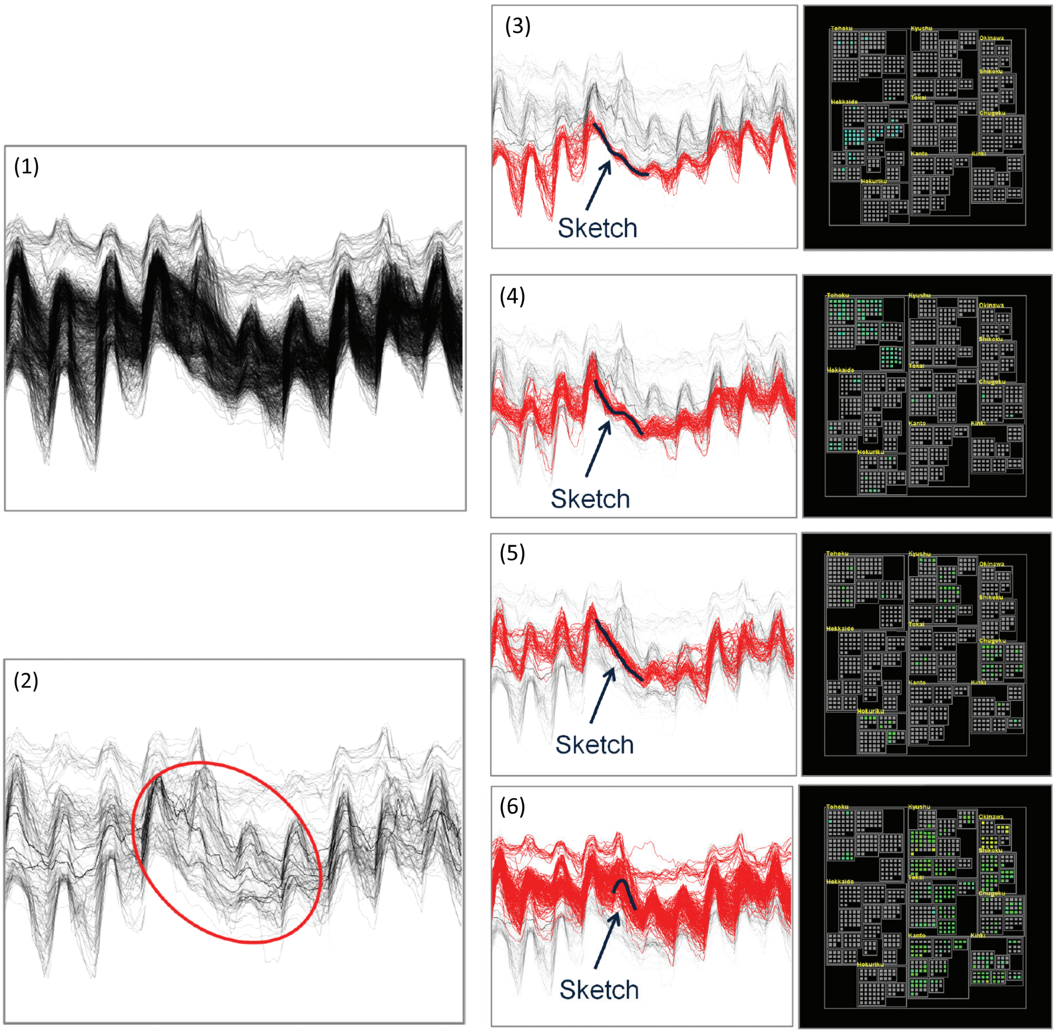

Figure 4: Visualization of weather data in Japan.

[4] G. Ellis, A. Dix, Density Control through Random Sampling: An Architectural Approach, *6th International Conference on Information Visualization*, 82-90, 2002.

[5] G. Ellis, A. Dix, Enabling Automatic Clutter Reduction in Parallel Coordinate Plots, *IEEE Transaction on Visualization and Computer Graphics*, 12(5), 717-723, 2006.

[6] Y.-H. Fua, M. O. Ward, E. A. Rundensteiner, Hierarchical Parallel Coordinates for Visualizing Large Multivariate Data Sets, *IEEE Visualization '99*, 43-50, 1999.

[7] 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.

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

[9] D. Holten, Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data, *IEEE Transaction on Visualization and Computer Graphics*, 12(5), 741-748, 2006.

[10] T. Itoh, H. Takakura, A. Sawada, K. Koyamada, Hierarchical Visualization of Network Intrusion Detection Data in the IP Address Space, *IEEE Computer Graphics and Applications*, 26(2), 40-47, 2006

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

[12] P. Saraiya, P. Lee, C. North, Visualization of Graphs with Associated Timeseries Data, *IEEE Symposium on Information Visualization*, 225-232, 2005.

[13] B. Shneiderman, Dynamic Queries for Visual Information Seeking, *IEEE Software*, 11(6), 70-77, 1994.

[14] 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.