A Polyline-Based Visualization Technique for Tagged Time-Varying Data

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

Abstract

We have various interesting time-varying data in our daily life, such as weather data (e.g., temperature and air pressure) and stock prices. Such time-varying data is often associated with other information: for example, temperatures can be associated with weather, and stock prices can be associated with social or economic incidents. Meanwhile, we often draw large-scale time-varying data by multiple polylines in one space to compare the time variation of multiple values. We think it should be interesting if such time-varying data is effectively visualized with their associated information. This paper presents a technique for polyline-based visualization and level-of-detail control of tagged time-varying data. Supposing the associated information is attached as tags of the time-varying values, the technique generates clusters of the time-varying values grouped by the tags, 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 values which belong to the clusters of the representatives.

Keywords— **Information visualization, tagged time-varying data, level-of-detail control.**

1 Introduction

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. Wattenberg et al. presented a sketch-based query interface to search for specific shapes of polylines [8]. Hochheiser et al. presented Timeboxes and TimeSearcher [3], a gradient- and range-based query interface for polyline-based time-varying data visualization Some works focused on similarity-based pattern and outlier discovery. Buono et al. presented a technique to interactively search for similar pattern [1] as an extension of TimeSearcher, and a similarity-based forecasting technique [2] for the forecasts the future patterns. Lin et al. presented a technique to discover non-trivial patterns [5], by clustering a set of time-varying values and searching for outliers. Wang et al. presented a technique for important polyline selection [7]. Recently we presented two time-varying data visualization techniques, featuring sketch query on the clustered view [6], and pattern display on the heatmap [4].

Meanwhile, such time-varying data is often associated with other information: for example, temperatures can be associated with weather, and stock prices can be associated with social or economic incidents. Such information may be tightly correlated with the time-varying values, and therefore, it should be useful if time-varying data visualization techniques simultaneously display such associated information.

This paper presents a new time-varying data visualization technique which supposes time series values have tags for each time. We suppose that tags consist of a set of predefined terms: for example, sunny, cloudy, and rainy for weather data, or exercising, eating, or sleeping for health care data. As a preprocessing, the technique clusters polylines based on their shapes and tags, and then selects representative polylines from the clusters. It realizes smooth level-of-detail control by interactively controlling the number of polylines to be displayed. Also, the technique features click and sketch interfaces so that users can interactively select particular polylines which are tagged with the user-interested terms. This paper presents the effectiveness of the presented technique with Japanese weather data recorded by AMeDAS (Automated Meteorological Data Acquisition System).

2 Level-of-detail control for time varying data visualization

This section briefly introduces a level-of-detail control and sketch interface [6] for time-varying data visualization previously we presented.

The technique supposes the following time series data, consisting of a set of values $P = (p_1, p_2, ..., p_n)$ repre-

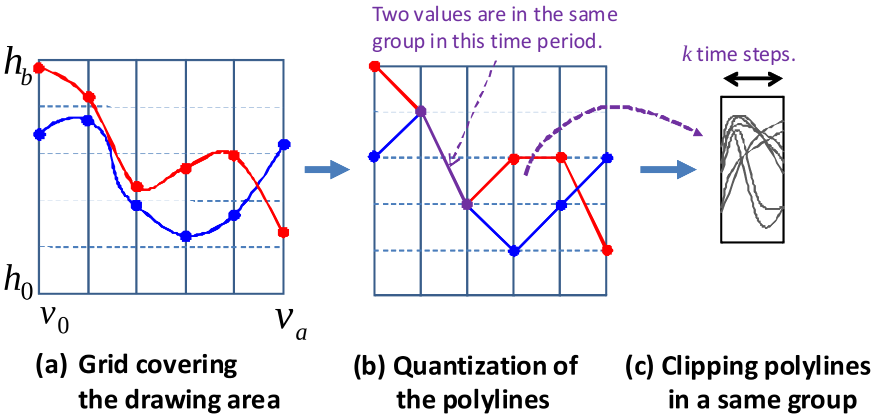

Figure 1: Quantization of polylines.

sented 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 time of the 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.

As a preprocessing, the technique temporarily quantizes polylines, generates clusters of them, and selects representative polylines from the clusters. While the quantization step, the technique generates a grid surrounding all polylines, and calculates intersections between the polylines and grid-lines. It then generates rough polylines by connecting the intersections, and uses them for the clustering. Number of clusters can be controlled by the resolution of the grid as well as similarity threshold values, and our implementation prepares several clustering results so that the number of representative polylines smoothly varies.

The technique then initially displays representative polylines. Smoothly switches clustering results, it can seamlessly change the number of representative polylines to be displayed. Also, the technique provides a click interface, so that users can specify interesting representatives by directly clicking. It also provides a sketch interface, so that users can specify interesting representatives which have partial shapes similar to the sketched curves.

3 Extension to the tagged time-varying data visualization

This section proposes an extended visualization technique for tagged time-varying data. This paper extends the aforementioned time series data as follows: we describe the tags of the i-th polyline as $w_i = (w_{i1}, w_{i2}, ..., w_{im})$; w_{ij} denotes the tag at the j-th time of the i-th polyline, as well as p_{ij} denotes the value at the j-th time of the i-th polyline.

3.1 Clustering and Representative Polyline Selection

The extended technique displays adequate number of representative polylines to reduce the cluttering among the polylines and improve the readability. The technique clips the polylines interval by interval, and then generates clusters of clipped polylines for each interval, where clipped polylines in a cluster is similarly shaped and tagged.

The technique first generates a grid covering the drawing area, and then divides into $a \times b$ subspaces, as shown in Figure 1(a). Here this paper formalizes the grid as follows: h_i is the i-th horizontal line of the grid ($0 \leq i \leq b$), v_i is the i-th vertical line of the grid ($0 \leq i \leq a$), t_i is the time at v_i, and b_i is the value at h_i.

The technique first samples P at t_0 to t_a, and temporarily quantizes the sampled values at b_0 to b_b. The technique then generates groups of polylines, if the polylines have the same quantized values both at t_{i-1} and t_i, as shown in Figure 1(b). It then clips polylines of a group by t_{i-1} and t_i, as shown in Figure 1(c), and generates clusters of the clipped polygons.

Procedure for clustering of tagged polylines is as follows. Figure 2 denotes the procedure, while the colors of polylines denote their dominant tags. Here, the technique regards the clipped polylines as k-dimensional vectors, while they contain k time steps between t_{i-1} and t_i. This step firstly divides the clipped polylines according to tags, as shown in Figure 2(b), using a dendrogram from the polylines constructed according to similarities of their k-dimensional vectors $(w_{jt_{i-1}}, ..., w_{jt_i})$. The technique then applies a non-hierarchical clustering (e.g. k-means) to the polylines in each cluster, using their k-dimensional vectors $(p_{jt_{i-1}}, ..., p_{jt_i})$. Consequently, it generates clusters consisting of similarly tagged and shaped fragments of polylines, as shown in Figure 2(c).

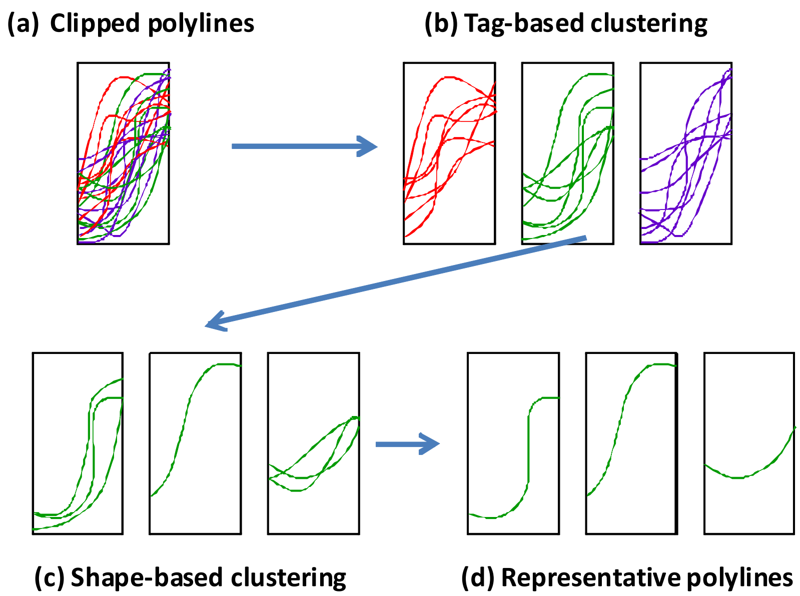

Figure 2: Clustering of polylines.

Next, the technique selects representative polylines, for each cluster, as shown in Figure 2(d). 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.

3.2 Interactive Visualization

The extended technique represents the time-varying data as colored polylines. It assigns colors to the tags (e.g. red to sunny, and gray to cloudy, to the tags of weather data), and draws the polyline in the assigned colors. If the vertices of a segment of a polyline have different tags, it interpolates the colors along the segments. The technique also features selective polyline display based on the tags. Users can select particular tags so that the technique can draw only corresponding parts of the polylines. They can also select particular tags to be filtered from the display of the polylines.

Initially our technique draws only the representative polylines. 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.

The technique provides click and sketch interfaces, so that users can specify interesting representatives by directly clicking or sketching particular shapes. 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. 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.

Users can specify particular tags to be extracted by the above query operations. It can highlight only the parts of the polylines corresponding to the specified tags while the click or sketch operations. Also, it can reactivate only the parts of the non-representative polylines corresponding to the specified tags.

4 Examples

We developed the presented technique with JDK (Java Development Kit) 1.6, and executed on a personal com-

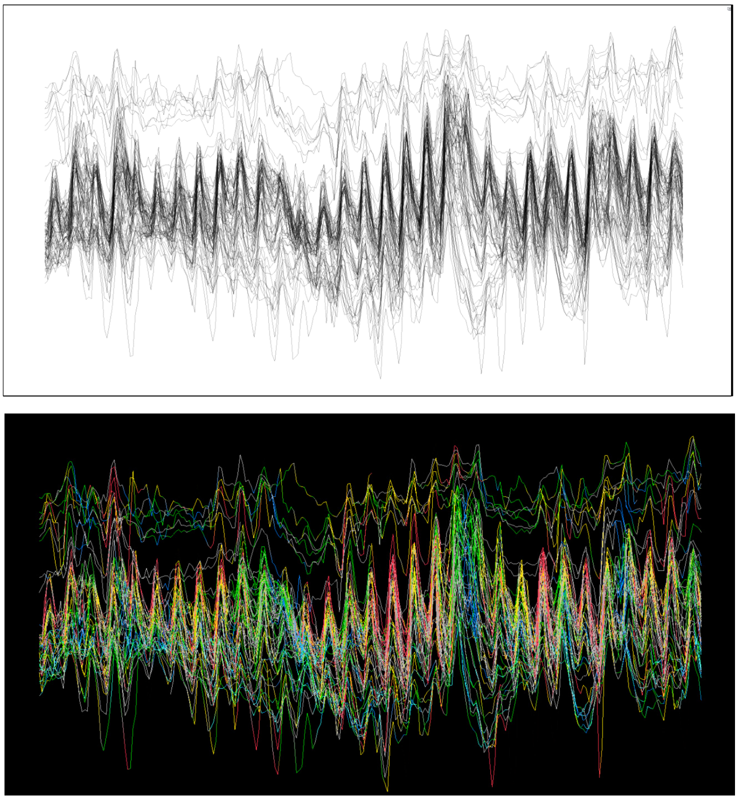

Figure 3: (a) Overview without tags. (b) Overview with tags.

puter (CPU 2.7GHz Dual Core, RAM 8.0GB) with Windows 7 (64bit). 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 83 points in every 2 hours. We then assigned weather tags including ”Clear”, ”Sunny”, ”Cloudy”, ”Rainy”, and ”Snowy” to temperature value of each time of each point.

Our implementation draws the segments of ”Clear” polylines in red, ”Sunny” in yellow, ”Cloudy” in green, ”Rainy” in blue, and ”Snowy” in cyan, interpolating the colors if tags of two ends of a segment is different. Exceptionally it may draw the segments in gray if we cannot obtain the weather data. While using click or sketch interfaces, our implementation draws selected polylines brightly, and others in gray.

Figure 3 shows an overview of the temperature data. While Figure 3(a) displays just major and outlier variations of temperature, Figure 3(b) demonstrates much more information. Colors in the dense parts denote major weather of the days. Red or yellow (”Clear” or ”Sunny”) is mainly observed at the higher peak times of the temperature. The variation looks more complicated when less points were ”Clear” or ”Sunny”. This result well demonstrates the effectiveness of the visualization of tagged time-varying data, since it is difficult to obtain such knowledge from Figure 3(a).

Figure 4 shows a zoom up view of temperature variation during three days. Figure 4(a) shows the original view of the tagged temperature data before applying level-of-detail control, and Figure 4(b) shows after the level-of-detail control. We can observe that temperatures drawn as blue or cyan polylines continued decreasing for almost 24 hours, while temperatures drawn as yellow or green polylines increased in the day times. Figure 4(a) looks somewhat crowded, while Figure 4(b) represents the features of the data more clearly. Again, it is difficult to obtain such knowledge from the visualization results without tags, as shown in Figure 4(c).

Figure 5 shows interactive polyline selection by using click and sketch interfaces. Figure 5(Upper) shows the result which we clicked middle-range points while selecting two tags ”Clear” and ”Sunny”; consequently polylines tagged as ”Clear” or ”Sunny” at the clicked points were highlighted. It represents that range of temperature between the daytime and the night is relatively large, while

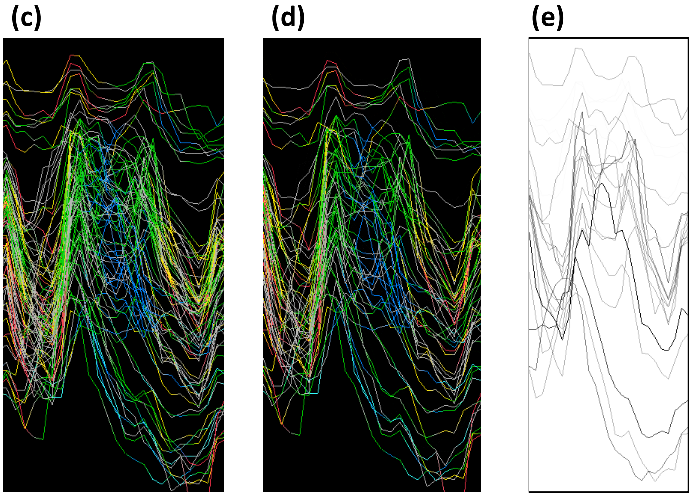

Figure 4: (a) Zoom up view with tags, without level-of-detail control. (b) Zoom up view with tags, with level-of-detail control. (c) Zoom up view without tags.

the variation is quite stable. Figure 5(Center) shows the result which we clicked low-range points while selecting two tags ”Rainy” and ”Snowy”; consequently polylines tagged as ”Rainy” or ”Snowy” at the clicked points were highlighted. It represents that range of temperature between the daytime and the night is relatively small, and the average is a bit lower than temperatures of ”Clear” and ”Sunny”. Also, we can find that extremely lower temperature can be often observed from ”Clear” or ”Sunny” points. Similar knowledge can be brought from Figure 5(Lower), which we clicked low-range points while not selecting any tags, and consequently polylines which have extremely lower values at clicked points are highlighted. Again, we can find that extremely lower temperature can be often observed from ”Clear” or ”Sunny” points, while temperature variation is relatively smaller at ”Rainy” or ”Snowy” points at that times.

5 Conclusion

This paper presented a polyline-based visualization technique for tagged time-varying data. The paper first described the definition of the tagged time-varying data, and presented techniques for level-of-detail control and interactive polyline selection. It also demonstrated effectiveness of the technique by applying temperature data with weather tags.

Our potential future work includes the following:

[Many kinds of tags:] Since our current implementation represents tags as colors, it may be difficult to visually distinguish if we have many kinds (e.g. more than 10) of tags. We would like to discuss what kinds of visual metaphor can be more effective for the representation of more kinds of tags.

[Multiple tags at a point:] It is also difficult for our current implementation to represent if multiple tags are simultaneously assigned to a particular time of a particular polyline. Again, we would like to discuss what kinds of visual metaphor can be more effective for the representation of multiple tags.

[Observation of tag-change:] It is interesting for several kinds of data to observe how time series values vary when the assigned tags change. We would like to add features to the technique so that we can focus on time series value variation with particular patterns of tag changes.

[More applications and tests:] We would like to apply more various data to the technique, including medical measurement datasets, system measurement datasets, and stock price datasets. Also, we would like to have experiments for subjective and objective evaluations of the technique.

References

[1] 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.

[2] 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 Visualisation*, 191-196, 2007.

[3] H. Hochheiser, B. Shneiderman, Dynamic query tools for time series data sets: Timeboxwidgets

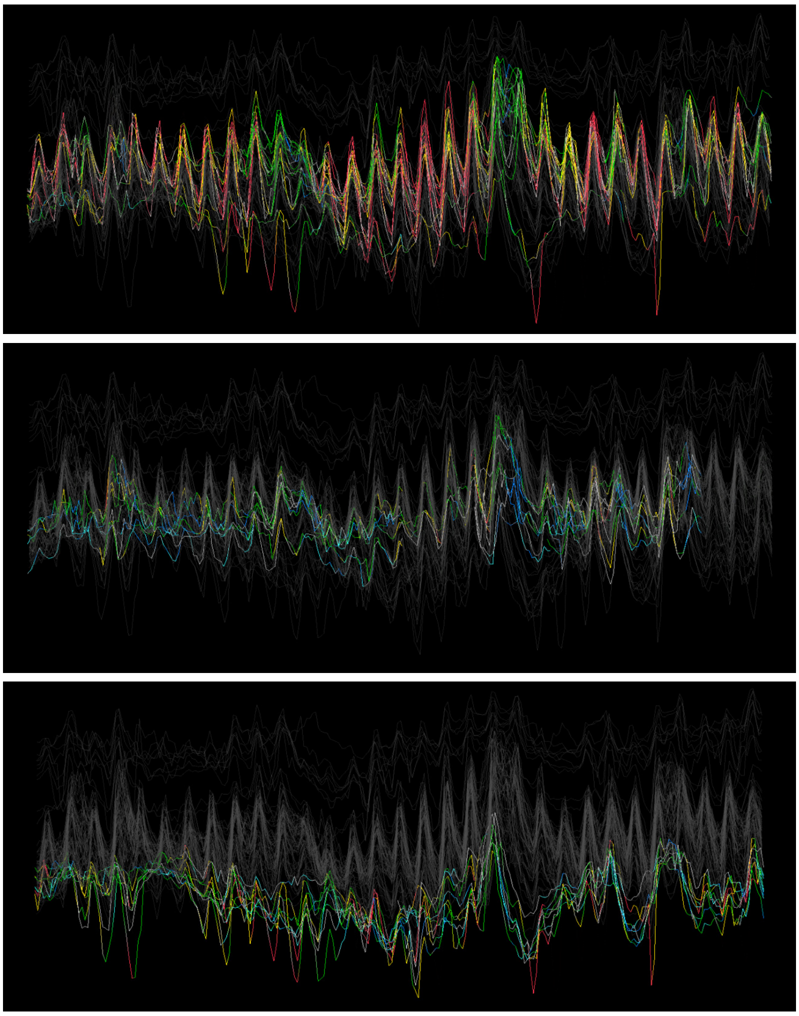

Figure 5: Interactive selection. (Upper) Polylines tagged as ”Clear” or ”Sunny” at the clicked points are highlighted. (Center) Polylines tagged as ”Rainy” or ”Snowy” at the clicked points are highlighted. (Lower) Polylines which have extremely lower values at clicked points are highlighted.

for interactive exploration, *Information Visualization*, 3(1):1-18, 2004.

[4] M. Imoto, T. Itoh, A 3d visualization technique for large scale time-varying data, *14th International Conference on Information Visualisation*, 17-22, 2010.

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

[6] Y. Uchida, T. Itoh, A visualization and level-of-detail control techniquefor large scale time series data, *13th International Conference on Information Visualisation*, 80-85, 2009.

[7] C. Wang, H. Yu, K.-L. Ma, Importance-driven time-varying data visuaization, *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547-1554, 2008.

[8] 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.