# TECHNICAL RESEARCH REPORT

## Representing Unevenly-Spaced Time Series Data for Visualization and Interactive Exploration (2005)

*by Aleks Aris, Ben Shneiderman, Catherine Plaisant, Galit Shmueli and Wolfgang Jank*

**TR 2005-73**

INSTITUTE FOR SYSTEMS RESEARCH

# Representing Unevenly-Spaced Time Series Data for Visualization and Interactive Exploration

Aleks Aris[1], Ben Shneiderman[1], Catherine Plaisant[1],
Galit Shmueli[2] and Wolfgang Jank[2]

[1] Human-Computer Interaction Laboratory
University of Maryland Institute for Advanced Computer Studies
{aris, ben, plaisant} @cs.umd.edu

[2] Department of Decision and Information Technologies,
Robert H. Smith School of Business
{gshmueli, wjank} @rhsmith.umd.edu

University of Maryland, College Park, MD 20742 USA

**Abstract**. Visualizing time series data is useful to support discovery of relations and patterns in financial, genomic, medical and other applications. In most time series, measurements are equally spaced over time. This paper discusses the challenges for unevenly-spaced time series data and presents four methods to represent them: sampled events, aggregated sampled events, event index and interleaved event index. We developed these methods while studying eBay auction data with TimeSearcher. We describe the advantages, disadvantages, choices for algorithms and parameters, and compare the different methods. Since each method has its advantages, this paper provides guidance for choosing the right combination of methods, algorithms, and parameters to solve a given problem for unevenly-spaced time series. Interaction issues such as screen resolution, response time for dynamic queries, and meaning of the visual display are governed by these decisions.

## 1 Introduction

Time series data consist of measurements of a variable over time. In many cases, measurements are evenly-spaced over time. Muller & Schumann provide an extensive survey on visualizing real-valued multivariate time-dependent data, with the emphasis on evenly-spaced data [14]. Silva & Catarci's review extends to categorical data and shows examples of unevenly-spaced data [16]. Like many researchers ([1], [3], [17]) the designers of TimeSearcher 1 assumed equally spaced time series data [9]. Common examples are daily stock prices and electric potential measurements taken from electrocardiograms at regular short intervals. Van Wijk & Van Selow show how to visualize evenly-spaced data on multiple scales. In their paper [18], they visualize the number of employees working in a company at various time points on daily, weekly, monthly or yearly scale by using time series plots and a calendar. Several researchers show how to visualize evenly-spaced periodic data using spirals to discover patterns and relations ([8], [19]). While evenly-spaced data occurs frequently, there are many examples where data are not spaced equally over time.

Our current research looks at online auction data, which consist of series of bids with timestamps, dollar amounts, bidder ID etc. Other examples are traffic incident data on highways, blood test results in patient records, and postings on Internet discussion boards. In these examples, the measurements are not scheduled beforehand and can occur at any time. We call such measurements "events". Their occurrence over time is unpredictable and in general, no simple formula can map natural numbers to the timing of these events. We call such time series "unevenly-spaced", as opposed to the more common "evenly-spaced" time series.

While it is straightforward to plot an evenly-spaced time series, at least the ones with small cardinality, it is challenging to plot unevenly-spaced ones. The arbitrary spacing in such time series poses trade-offs and problems such as precision of data, encoding of timing, and representation of data, which may or may not result in data loss. For instance, consider visualizing eBay auction data that displays all the bids that were placed during an auction. Consecutive bids can be separated by as much as entire days or by as little as a single second as is often the case toward the end of the auction. Some auctions only have a few bids; for other auctions the number of bids ranges in the hundreds. In order to see the real data with full precision, the time points on the x-axis would need to be 1 second apart from each other. Since the longest eBay auction is 10 days, this would result in 864,000 time points. The average screen has only between 1,024 and 1,600 horizontal pixels, and therefore, it becomes a challenge to fit the data in this resolution for getting an overview, and to provide fast processing to support interactive exploration and access to all details. As a result, we investigate several methods to overcome this problem and discuss the issues surrounding them. Each of the methods we discuss transforms unevenly-spaced time series into evenly-spaced points on the x-axis of the visualization. Each representation is effective for addressing a subset of the problems and users' tasks, but we focus on representations that provide rich overviews of the data while minimizing data loss and required memory resources.

In Section 2, we refine the scope of this paper and describe the sample dataset we use to illustrate the methods. In Section 3, we present four methods to represent unevenly-spaced time series in detail and discuss their advantages and disadvantages. Section 4 compares the methods in more depth. Section 5 reviews some additional methods.

## 2 Characteristics of Time Series under Consideration

Time series data is a general term that includes many variations, such as nominal, ordinal, and continuous values at evenly- or unevenly-spaced time points [14]. The data may consist of a single time series or multiple ones; it may consist of a single variable or multiple variables. For example, meteorological data may have multiple variables, such as temperature, barometric pressure, and rainfall for each day at multiple locations. Although TimeSearcher now supports multiple variables, for simplicity, this paper focuses on multiple time series for a single variable.

Our sample dataset consists of 3 auctions from www.eBay.com, which is the largest consumer-to-consumer electronic auction house. Each auction is a time series of bids placed throughout that auction. eBay uses a "second price" auction format with a "proxy bidding" mechanism. This means that the winner pays the second highest bid, and that at every point in the auction only the second highest bid is disclosed. Bidders, therefore, sometimes submit bids that are lower than the highest bid, and therefore the resulting time series is not necessarily monotonic.
Table 1 describes the main characteristics of the 3 sample auctions:

**Table 1 Auction characteristics**

| Auction Nam | Length | Number of bids | Opening bid | Closing bi |
|---|---|---|---|---|
| A | 3 days | 13 | 99.00 | 224.72 |
| B | 5 days | 29 | 9.99 | 232.50 |
| C | 10 days | 18 | 5.00 | 172.50 |

The auctions start and end at different times, however, they occur close to each other such that all auctions fall within a 13-day period. This sample dataset is used in all the figures of the paper.

In all our sample visualizations, the events (=placed bids) are connected via line segments. Although this has the disadvantage of making the detection of change in the y-value more difficult, the perception of which events belong to the same time series is much easier. An alternative is to use profile plots [15], which provide the reverse effect. With profile plots measurements are visualized as points. As a result, it is easy to detect them. However, without any other cue, there is no way to tell which measurement belongs to which time series.

## 3 Description of Four Representation Methods

Our discussions led to several proposals for representation methods, and we selected four for implementation and detailed investigation: sampled events, aggregated sampled events, event index, and interleaved event index.

### 3.1 Sampled Events

In this representation, we sample the value at regular intervals. Fig. 1 illustrates the visualization of our sampled auction dataset with a sampling interval of 5 minutes. The y-axis reflects the bid amount ($) and the x-axis is an ordinary time scale.

We define the *cardinality of a time series* to be the number of time points a single time series contains, and define the *cardinality of the x-axis* to be the number of time points on the x-axis of the visualization of the multiple time series. Those numbers will be used to compare the four methods. Along with the number of time series, the cardinality of the x-axis is useful for estimating the memory and processing requirements of the representation. For our sample dataset (Fig. 1) the sampled events method at five minute intervals makes the cardinality of A = 866, B = 1444, C = 2880. The cardinality of the x-axis is 3851.

Sampling requires an algorithm to determine what y-value to use when no event occurred at the time of the sampling. In Fig. 1, the most recent event value is used. (**Fig. 1** shows the data as it is visualized in TimeSearcher 2.1.) In auction data this makes sense as the most recent bid corresponds to the current auction price. As the sampling interval increases, the chance that events will not appear on the display increases, resulting in greater data loss.

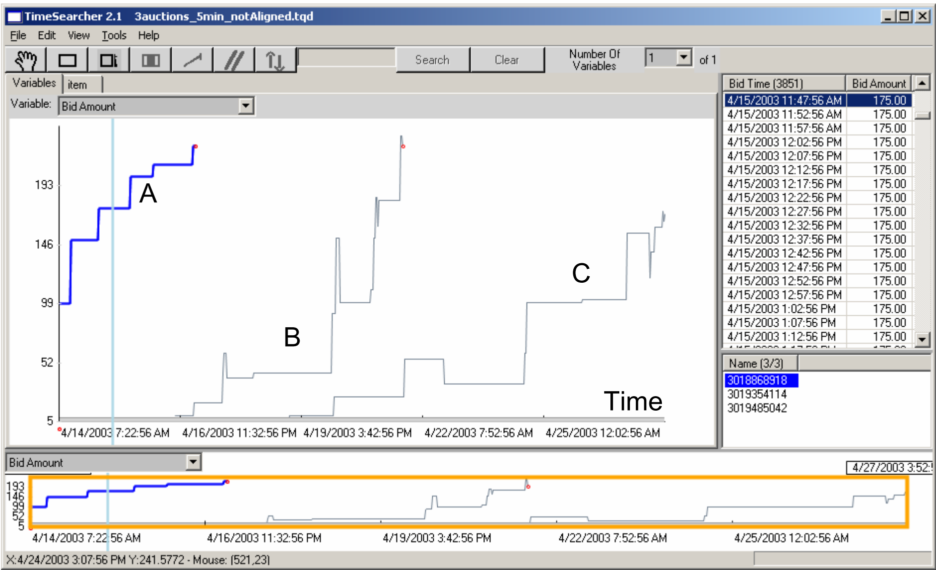



**Fig. 1.** Time series sampled at 5-minute intervals (not aligned), visualized in TimeSearcher 2.1. Bid amounts sometimes decline because eBay uses second-price auctions and does not disclose the highest bid

**Cardinality of time series:** A = 866, B = 1444, C = 2880.
**Cardinality of the x-axis:** 3851 time points.

**Advantages:**

- Horizontal distance between time points accurately encodes time. When and how long a bid is placed before/after another can be perceived by the horizontal distance (up to the accuracy of the interval).
- Both the time-order relation and the length of time are conveyed, even though zooming might be required to see the order of close events.
- The visual length of a line indicates the actual duration of the corresponding time series.

**Disadvantages:**

- Many time points are generated.

- There may be data loss. Data loss will increase with the sampling interval. For example, if more than one bid occurred within the sampling interval, the presence and value of all the bids except for the last one will be lost.
- The y-values are exact but the time points (on the x-axis) are approximate. The error is bounded by the length of the sampling interval.

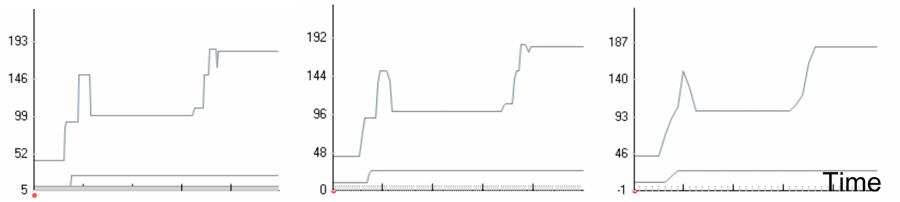
### 3.2 Aggregated Sampled Events

The sampled events representation generates many time points. When viewing time series over long periods with small sampling intervals, the number of pixels on the screen will not be sufficient for an overview to show all the data points. In such instances, aggregation can be used to allow the overview to fit into the screen, mitigate the amount of processing required and maintain interactivity. The main difference between aggregation and sampling is that a sampled value represents only one event, while an aggregated value represents a collection of events.

In order to implement aggregation for time series, the following parameters should be specified: 1) the *level of aggregation* is the number of consecutive values that an aggregated value represents. 2) an *algorithm (a summary statistic) to determine the aggregated y-value*: A common approach is to take the average. However, there are other choices such as the median, min or max. 3) an a*lgorithm to determine the aggregated x-value*: Many possibilities exist. Besides taking the first, last, or middle value of the time points, a totally new value could be generated as well. One motivation to use aggregation is to reduce the overhead for operations available in interactive visualization of time series. This is especially useful when the difference between the aggregated view and the sampled view is indiscernible. For example, when the data of **Fig. 1** is aggregated with 12x aggregation level, the overview of the aggregated data has the same appearance as in **Fig. 1**. While the sampling interval is 5 minutes in **Fig. 1**, the granularity of every time point is 1 hour in the aggregated version with 12x level. We used the average function to determine the aggregated y-value. To determine the x-value, we used the last time point from the set of points that are aggregated. The cardinalities are greatly reduced: While the cardinality of x-axis reduced from 3851 to 321 time points, the cardinality of time series reduced from 866, 1444, and 2880 time points to 73, 121, and 240 time points, respectively, in the aggregated version.

In an aggregated view, the details can become apparent when users zoom in. **Fig. 2** shows three zoomed versions of a portion of the data in **Fig. 1** with different aggregation levels, showing the impact of different aggregation levels: 5-minute, 2-hour, and 6-hour intervals. While **Fig. 2a** has non-aggregated data (relative to **Fig. 1**), **Fig. 2b** and **Fig. 2c** show increasing aggregation levels. As the aggregation level increases, the precision degrades. One possibility is to aggregate dynamically considering the amount of information to be displayed and available space on the screen to keep the differences unnoticeable. The algorithm for such a dynamic aggregation can become complex but several researchers implemented examples of dynamic aggregation [5], [7]. In Brodbeck & Girardin's TrendDisplay, the raw data is never shown, rather they are represented with one of the four different levels of detail, which are density distributions, thin box plots, box plots plus outliers, and bar histograms. Hence, they can be considered as some form of aggregation, and the aggregation level dynamically changes within these four levels according to the density of the data to be displayed [7]. This strategy is an implementation of semantic zooming [4]. In Berry & Munzner's BinX, the aggregation technique of binning is used, and users can choose the aggregation level interactively. There are visual cues to indicate the aggregation level, and evenly-spaced time series are used [5].

Sampling and aggregation both require a method to determine what value to use when no event occurred at all in the time interval. For applications with highly variable values, interpolated or moving average estimates are risky, making it necessary to have a special indicator for missing values [17].




a) 3000 time points b) 125 time points c) 40 time points

**Fig. 2.** Three zoomed versions of a portion of Fig. 1 with various aggregation levels. a) shows the data at 5-min. intervals (1x) . b) at 2-hour intervals (24x) c) at 6-hour intervals (72x)

**Advantages:**

- The cardinality of the time series and the x-axis are reduced, which increases performance.
- Both the time-order relation of events and the length of time are conveyed, although approximately.
- The visual length of a line indicates the actual duration of the corresponding time series.

**Disadvantages:**

- Besides the data loss due to sampling, the data is only approximate. Neither the y-values, nor the time points precisely reflect the reality. As the aggregation level increases, greater degradation occurs.
- Parameters of aggregation need to be changed according to the density of the data to minimize error. The algorithm for finding optimal parameters tends to be complex.

### 3.3 Event Index

Unlike the previous two methods, which represent time linearly on the horizontal x-axis, the event index method does not. This method distorts the x-axis and stretches it when there are more events by separating each event by an equal amount of space, regardless of the elapsed time between events. In our example of auctions, the event index method provides a simple and powerful representation when users are analyzing and comparing the number of bids and their amounts, independently from the time they took place.

In this method, the cardinality of an auction is equal to the number of bids in that auction (e.g. 13 for Auction A). The cardinality of the x-axis equals the largest time-series cardinality among all time series (here 29 because of auction B) (**Fig. 3**).

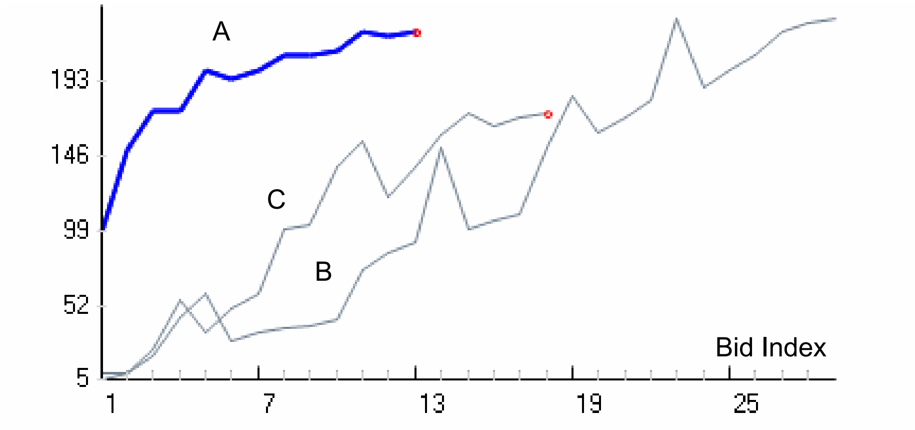



**Fig. 3.** Event index visualization of the same dataset

**Cardinality of each time series:** A = 13, B = 29, C = 18
**Cardinality of the x-axis:** 29

In Fig. 3, the horizontal length in each auction corresponds to the number of bids in that auction. Note that clock time is not encoded anywhere although the order of events is. Even though the $n^{th}$ bid of each auction is lined up vertically, this alignment does not imply that the bids were placed at the same time.

**Advantages:**

- The cardinality of the time series and the x-axis are kept as small as possible without loss of data in terms of bid amounts. To contrast with the sampled events method, which may have data loss, notice that while A looks monotone in **Fig. 1**, in reality, it is not monotone as we understand from the dip at the $6^{th}$ index in **Fig. 3**. (Bid data from eBay are non-monotone as eBay uses second-price auctions, where the highest bid is hidden. This results in bids that might be lower than the previously placed bid.)
- Having the smallest number of data points maximizes performance in an interactive visualization setting.
- Comparing the $n^{th}$ bids across two or more auctions is easy.
- In the context of auctions, long lines with many bids represent more competitive auctions.

**Disadvantages:**

- Time is not conveyed, neither absolute nor relative. The inherent alignment of auctions results in no information on which auction started earlier.

- The order of bids, although preserved within an auction, is not preserved across auctions.
- In the context of auctions: the visual length of a time series doesn't indicate the duration of an auction. For example, C appears shorter than B because it has fewer bids, while in fact its duration is twice as long as B (10 days vs. 5 days).

### 3.4 Interleaved Event Index

Similar to the previous method, the Interleaved event index does not represent time linearly. On the contrary, it represents the sequence of events across multiple time series. In this method, all the time points of all the time series are collected, sorted by time, and indexed. This new index is used for the x-axis. It treats the time points as ordinal, but ignores the time interval information.

All events are shown in the order they occurred, therefore whenever an event appears to the right of another one - even on a different auction, it happened later in time. However, since the time duration information is not encoded, users cannot determine the duration between events. The resulting effect is the stretching of time series during the periods where they have many events, but also during the periods where they have no events while other series do (Fig. 4).

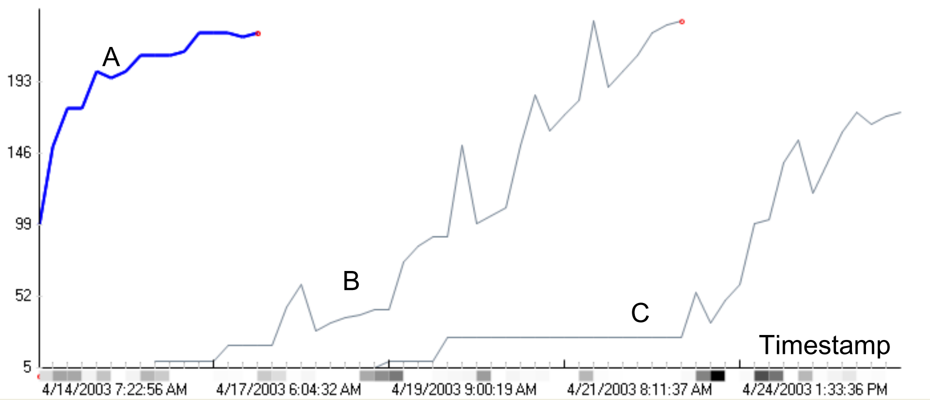



**Fig. 4.** Interleaved event index visualization of the same sample auction dataset. Dates have been used to label the time axis but note that these dates are only index labels

**Cardinality of each time series:** A = 16, B = 37, C = 44 (excluding the missing values at the beginning and end)
**Cardinality of the x-axis:** 60

In our example, there are 60 time points on the x-axis (13+29+18 = 60). The cardinality of the x-axis is bounded by the sum of the cardinality of the time series, but it is smaller when there are simultaneous events across auctions. Our count excludes the data points needed to mark the missing values, i.e. the normal lack of data before and after the auction period.

**Advantages:**

- Uses a small number of time points without any loss of data
- Preserves the temporal order of events across time series.
- The visual length of a time series is an indicator of its length in relation to other time series in terms of time. It conveys whether it is longer or not, but not how long.

**Disadvantages:**

- The time between two consecutive events is not conveyed.
- The granularity in time is arbitrary and changes from one time point to another. In **Fig. 4**, it is conveyed by labeling with date-time information; however, this is difficult for users to interpret. Nevertheless, it is possible to convey with cues such as the color intensity or thickness of the time point segments on the x-axis. **Fig. 4** illustrates how shading the segments on the x-axis can be used to indicate density of intervals in terms of time. The darker the shade, the longer the time span the interval represents.
- It is difficult to tell which points are actual events, versus additional points introduced by the representation. In the auction example changes in angle in the line obviously correspond to bids, but bids of equal amount will not be visible. One solution is to mark events with a dot or a small shape.

## 4. Discussion

Each of the methods in Section 3 is specialized to deal with a subset of users' goals. There are trade-offs in choosing one method over another. The following table compares several features of the four visualizations:

**Table 2 Comparison of the features of the four visualizations**

| | Sampled | Aggregated | Event index | Interleaved |
|---|---|---|---|---|
| # of points for our example | 3851 time points | 320 time points | 29 indexed time points | 60 indexed time points |
| Bid order preserved across auctions? | Yes | Yes | No | Yes |
| Time encoded? | Yes | Yes | No | No |
| Visual length of a line shows: | Time series length in time | Time series length in time | # of events in that time series | # of distinct-time events in all time series |
| Event loss? | Yes | Yes (but used in aggregating) | No | No |
| Left alignment of time series | Optional | Optional | Inherent | Optional |
| Parameters required for the technique | Interval | level, method,.. | None | None |

The event index representation is ideal when users want to compare the measurements across time series. For example, users might want to compare the 2nd or 3rd bid values across auctions, and they don't need to know the specific times that the bids are placed. In this case, event index visualization provides just the right amount of data and enables very fast processing.

If users do need to know the order of event arrivals over time, the interleaved event index representation is more appropriate. For a small total number of events dispersed over a long period of time, this representation will be ideal and will facilitate optimal processing times for visual exploration. However, as the number of events increases, or when the screen space for the x-axis is too small, it may be wiser to switch to the sampled events representation. The interleaved event index representation is also good when users do not care to know how far apart in time the events took place. One instance where this representation is ideal is when one is interested in only the consecutive values of bids in an auction, and needs to compare them and investigate how the bid amounts in one or more than one auctions parallel in time affect bidders' behavior.

If users do care about the length of time between events, it is best to use the sampled events visualization. They will need to carefully choose the sampling interval because as the sampling interval gets longer, the processing becomes faster due to decreased number of time points, however, loss of data and approximation errors increase. If sampling leads to slow performance, users should consider aggregation.

## 5. Other Representations

Viewing multiple unevenly-spaced time series has the inherent characteristic that an individual time series can have a varying start time, end time, and length. For example the auction start time is chosen by the seller and could be anytime of the day or night. eBay auctions vary in length from 1 to 10 days. An important design choice is to decide if absolute time (actual clock time) or relative time (time relative to the beginning of the auction) should be used in the visual representation. Absolute and relative time representations allow different hypotheses to be made about phenomena taking place between or across auctions. For instance, "last minute bidding" is a known phenomenon in eBay auctions. To study this we would use absolute time for auctions of all durations. In comparison, if we are interested in the percentage of bids achieved by mid-auction, then relative time should be used. Using relative time is likely to reduce the cardinality of the x-axis. For example, aligning the start times in the sampled events method results in a cardinality of 2880 (opposed to 3851 when not aligned). It is important to make clear to the user if alignment has been used or not.

Another alignment method is to stretch the time series so that both the start and end times of the time series are aligned [3]. Bapna, Jank and Shmueli [1] use a similar approach for auction data. They represent each time series (i.e., auction) by a smooth curve, which is derived by penalized smoothing splines. Then, they use a linear transformation to stretch and align the curves to start and end at the same time.

Hybrid techniques could be employed where different sections of the x-axis use different techniques. This is not appropriate for the two index techniques, but designers may consider using the sampled events method for sections that require detailed information and the aggregated sampled events method for other sections.

Designers can also choose to use different parameters for different sections. For example, auctions typically have more events toward the end; therefore a smaller sampling interval will help convey the excitement of the final moments. Clearly indicating to users the location of those sections and their characteristics is crucial. Another completely different hybrid method would be aggregating indexed data (i.e. aggregating over a fixed number of events, not over a fixed time period).

There are certainly other methods for representing time series data, such as the Symmetric Aggregate Approximation (SAX), which could be adapted for unevenly-spaced time series [13]. Even more compact representations using iconic or glyph strategies could be helpful in getting a quick glance to see similar or different time series [9]. Other tasks such as motif finding or anomaly detection could inspire further novel methods [13] as could coping with uncertainty and frequent missing values. Novel methods may also emerge in dealing with large numbers of measurements (more than $10^4$).

## 6. Conclusions

This paper identifies the issues and problems with representing and visualizing time series with unevenly-spaced measurements. We considered four methods and illustrated each on a sample dataset to understand their implications. Finally, we compared them and discussed which situations are best to use in different cases. Each method has its strengths and weaknesses for certain tasks, so users must understand the tradeoffs. Our contribution is to present the features of various representations in order to help users decide which one(s) to use. We have implemented all of these four methods to be visualized in TimeSearcher. There is a preprocessing step for the data for each method to put it into TimeSearcher format. In other words, computations such as sampling, aggregation, and interleaving event indices are performed outside of TimeSearcher to produce an evenly-spaced time series data, which can then be loaded into TimeSearcher 2. We plan to implement other methods as well and refine our understanding of benefits and disadvantages by exploring datasets of different origins.

**Acknowledgements**

We would like to thank the Center for Electronic Markets and Enterprises and the R.H. Smith School of Business at the University of Maryland for providing support for this project. We also thank Eamonn Keogh, Bill Kules, and Harry Hochheiser for providing useful feedback on early drafts of the paper.

**References**

1. Bapna, R., Jank, W., Shmueli, G. Price Formation and its Dynamics in Online Auctions, *Working paper, Smith School of Business, Univ. of Maryland*, 2004.

2. Bar-Joseph, Z., Analyzing time series gene expression data, *Bioinformatics*, 20(16), 2004.
3. Bar-Joseph, Z., Gerber, G., Gifford, D.K., Continuous representations of time series gene expression data, *Journal of Comp. Biology*, 10(3-4): 241-256, 2003.
4. Bederson, B.B., Hollan, J.D., Pad++: a zooming graphical interface for exploring alternate interface physics, *Proceedings of the 7th annual ACM symposium on User interface software and technology*, 17-26, 1994.
5. Berry, L., Munzner, T., BinX: Dynamic exploration of time series datasets across aggregation levels, *IEEE Info. Visualization 2004, Posters Compendium*, 5-6.
6. Bettini, C., A glossary of time granularity concepts, *Temporal Databases: Research and Practice*, Etzion et al. (Eds), Springer, 406-413, 1998.
7. Brodbeck, D., Girardin, L., Trend analysis in large timeseries of high-throughput screening data using a distortion-oriented lens with semantic zooming, *IEEE Symposium on Information Visualization,* Seattle, October 19-21, 2003.
8. Carlis, J.V., Konstan, J.A., Interactive visualization of serial periodic data, *Proc. of ACM UIST'98, San Francisco, CA,* 29-38, 1998.
9. Hinneburg, A., Keim, D.A., Wawryniuk, M.,HD-Eye: Visual mining of high-dimensional data, *IEEE Computer Graphics and Applications,* v.19 n.5, 22-31, September 1999.
10. Hochheiser, H., Shneiderman, B., Dynamic query tools for time series data sets: timebox widgets for interactive exploration, *Information Visualization*, Vol.3, Issue 1, Spring 2004, 1-18.
11. Keim, D.A, Information visualization and visual data mining, *IEEE Transactions on Visualization and Computer Graphics,*8(1), 1-8, 2002.
12. Keogh, E., Chakrabarti, K., Pazzani, M., Locally adaptive dimensionality reduction for indexing large time series databases, *ACM SIGMOD Record,* 30(2), 151-162.
13. Lin, J., Lankford, J., Keogh, E., Lonardi, S., Visually mining and monitoring massive time series*, Proc. ACM Conference on Knowledge Discovery and Data Mining*, ACM Press, New York, 460-469, 2004.
14. Muller, W., Schumann, H., Visualization methods for time-dependent data, *Proceedings of the 2003 Winter Simulation Conference*, 737-746, 2003.
15. Shmueli, G., Jank, Wolfgang, Visualizing online auctions, *Journal of Computational and Graphical Statistics*, Forthcoming.
16. Silva, S.F., Catarci, T., Visualization of linear time-oriented data: a survey, *Proceedings of the First International Conference on Web Information Systems Engineering*, Vol.1, 2000, 310-319.
17. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB, Missing value estimation methods for DNA microarrays, *Bioinformatics*, 2001 Jun;17(6):520-5.
18. Van Wijk, J. J., Van Selow, E. R., Cluster and calendar based visualization of time series data, *Proc. 1999 IEEE Symposium on Information Visualization*, IEEE Press, Piscataway, NJ, 4-9, 1999.
19. Weber, M., Alexa, M., Muller, W., Visualizing time series on spirals, *Proc. 2001 IEEE Symposium on Information Visualization,* IEEE Press, Piscataway, NJ, 7-14.