City Research Online

City, University of London Institutional Repository

Data Abstraction for Visualizing Large Time Series

G. Shurkhovetskyy[1], N. Andrienko[2,3], G. Andrienko[2,3], and G. Fuchs[2]

[1]University of Bonn, Germany
[2]Fraunhofer Institute IAIS, Germany
[3]City University London, UK

Abstract

Numeric time series is a class of data consisting of chronologically ordered observations represented by numeric values. Much of the data in various domains, such as financial, medical, and scientific, are represented in the form of time series. To cope with the increasing sizes of datasets, numerous approaches for abstracting large temporal data are developed in the area of data mining. Many of them proved to be useful for time series visualization. However, despite the existence of numerous surveys on time series mining and visualization, there is no comprehensive classification of the existing methods based on the needs of visualization designers. We propose a classification framework that defines essential criteria for selecting an abstraction method with an eye to subsequent visualization and support of users' analysis tasks. We show that approaches developed in the data mining field are capable of creating representations that are useful for visualizing time series data. We evaluate these methods in terms of the defined criteria and provide a summary table that can be easily used for selecting suitable abstraction methods depending on data properties, desirable form of representation, behavior features to be studied, required accuracy and level of detail, and the necessity of efficient search and querying. We also indicate directions for possible extension of the proposed classification framework.

Categories and Subject Descriptors (according to ACM CCS):   I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Time series are a type of data that can be found frequently in observation of real world phenomena in financial, medical, engineering, scientific, social and military applications. Interest in time series analysis has been present for several decades, if not centuries.

In many settings, the human eye is the most reliable tool for analyzing large data [Kei02]. Therefore, appropriate representation of time series is the foundation for efficient task solving. Many data mining tasks can be facilitated by visualization of the data. There is a bidirectional relationship between visualization of time series and corresponding data mining tasks. The majority of tasks mentioned in a survey on time series mining [Fu11] and associated methods designed to solve them are helpful in creating efficient visualizations. At the same time, efficient visualizations enhance the task solving process for many of the known mining tasks. For instance, visualization and interaction with data facilitates search for optimal parameterization of complex sys-

tems, and increases chances of arriving at results that are close to optimal [CRC03]. This is particularly the case when advanced analytical algorithms are applied, whose tuning is only possible with good understanding of their work principles [SK13].

However, the massive size of many contemporary data sets pose significant challenges to both visualization and analysis [DSP*17, Fek13]. Often, time series data are so large that it becomes either infeasible or useless to display them all. Attempts to do so result either in unresponsive displays or in cluttered visualizations that are impossible to read and do not allow extracting any meaningful information. Analysis algorithms, in turn, might degrade in performance. One way to diminish such effects is to rely on data abstraction. Representing datasets at hand in a form that is significantly smaller in size while preserving features important for the user is the ultimate goal of abstraction of any type of data, including temporal.

As mentioned by Aigner et al. [AMM*07], there are two

principal approaches to data abstraction, aggregation- and feature-based. The former assumes calculating aggregated data values for portions of data and visualizing these aggregates. The latter is based on showing only those parts of the data that satisfy certain preset criteria, which supposedly characterize chunks of data the user deems interesting, or that are important in the context of the analysis task at hand.

The task of time series abstraction for visualization has been addressed by many researchers who introduced diverse methods for solving the problem. In general, the problem can be addressed in two ways: reducing the length (from here on, by length we refer to the number of time steps in a single series), and reducing the number of time series in cases when there are multiple series to be dealt with.

Time series mining is an active area of research in data mining and knowledge discovery that produced, among others, numerous abstraction methods for time series. To the best of our knowledge, there have yet been no attempts to systematically review these methods from the perspective of visualization and visual analytics. Logically, the first step to good visualization design for large time series is the selection of a suitable abstraction algorithm that can facilitate users' perception and analysis. This work aims at laying the foundations for such a selection.

Achieving this aim involves two sub-goals: first, determine the relevant criteria for the selection of an abstraction method, and second, evaluate the existing abstraction methods in terms of these criteria. A starting point towards the first sub-goal is the consideration of two aspects: the possible properties of the *data* (i.e., time series) that need to be abstracted, and the intended *analysis tasks*. The first aspect relates to the applicability of methods while the second aspect refers to the utility of method results, i.e., whether the transformed data can effectively support the intended tasks.

The main contribution of this paper is providing navigation for visual analysts through a systematic inventory of available abstraction methods. Systematization driven by user analysis goals and informed by requirements for enabling visualization tasks is exactly what the visualization community has been advocating [BL10, AMM*08a, BL09, KMS*08].

This paper is organized as follows. In Section 2, we introduce the problem of visualizing large time series, explain the need for using abstraction, and define the scope of our work. Section 3 provides an overview of the related work. Section 4 describes our classification framework. We introduce and substantiate the proposed criteria for classifying and choosing abstraction techniques. In Section 5, representative methods for time series abstraction resulting from research in the field of data mining are evaluated in terms of these criteria. We continue with a discussion of how analysts can benefit from our classification framework and method survey in Section 6. Section 7 suggests possible research directions for extending the proposed framework with additional potentially relevant criteria. It is followed by a conclusion in Section 8.

2. Background

2.1. Time Series

A time series in general form is a data type represented by an ordered sequence of observations, where an observation is represented by a specific value of some attribute or a combination of values of several attributes, or *variables*, the latter term being typically used in statistics and data mining. In this paper, we focus on numeric-valued variables. A time series comprising values of a single variable is called *univariate*. A univariate time series of length T can be represented as $x(t) = x(1)...x(t)...x(T)$. A time series comprising values of several variables is called *multivariate*. A multivariate time series with N variables can be represented by N per-variable univariate time series with a common temporal domain. In literature, data with multiple variables are also referred to as *multidimensional*, the variables are called *dimensions*, and the number of variables in data is called the *dimensionality* of the data.

Multivariate (multidimensional) time series need to be distinguished from multiple univariate time series sharing the same variable. For example, a time series of weather parameters, including the temperature, wind speed, wind direction, and precipitation, is a multivariate time series in which each weather parameter is represented by one variable. Multiple time series of measurements of the same parameter, such as the temperature, taken at different locations is an example of multiple univariate time series with a common variable. These two cases require different approaches to the analysis and to the abstraction. Thus, multiple univariate time series can be clustered by similarity, which makes no sense for time series with the variables differing in their meanings, measurement units, and value ranges.

Numeric time series make a subclass of a larger class of data called *time-oriented* [AMST11] or *time-referenced* [AA05]. The latter term indicates that data consist of items referring to moments or intervals in time. The data items may not only be scalar values but also events [DSP*17], graphs [vdEHBvW16], images [BSH*16a], spatial distributions [AAB*10], or positions of moving objects [AA17]. Time-referenced data characterize processes and phenomena that vary in time and may be thus called *time-variant*.

Shneiderman [Shn96] underlines the importance of temporal information by considering it as a separate data type. Aigner et al. [AMM*08b] emphasize the uniqueness of the time parameter in temporal data and claim that its presence requires principally different approaches for data representation than those used for data without a temporal component.

Tufte [Tuf86] claims the time series plot is the most frequently used graphical representation of data, and that its

oldest example dates back to the tenth century. The set of techniques used to visualize time series has been notably expanded in recent decades by sophisticated approaches like ThemeRiver [HHWN02], calendar-based visualizations [VWVS99], spiral visualizations [WAM01], stacked graphs [BW08], horizon graphs [HKA09], time curves [BSH*16b], to name a few. A visual survey of time series visualizations compiled by Aigner et al. [AMM*08a] can be found at http://survey.timeviz.net.

2.2. Need and Requirements for Time Series Abstraction

When it comes to visualization, there are several conditions that must be taken into consideration when dealing with large time series. First, displaying the entire raw data on a commodity screen (rarely larger than 32 inches diagonally or having resolution greater than 4K) almost certainly results in a cluttered view, which will be of very little utility to a user whose understanding of the data will be hindered. For instance, using a FullHD display that has a horizontal resolution of 1920 pixels to visualize time series of more than 1920 time steps as a line chart creates the problem of accommodating more than one time step in every pixel column.

Second, massive amounts of data, if presented in their raw form, may overtax computing resources resulting in unacceptably slow responsiveness of the system. Abstraction is used to avoid or at least mitigate these issues [EF10].

The main goal of creating an abstraction is simplification, which means removing unnecessary details and/or extracting only important information (i.e., relevant to the analysis goals). Simplification involves generating a new, usually smaller in size, representation of given time series. Esling and Agon [EA12] define such a representation as "a model of original time series of reduced dimensionality such that the model closely approximates original time series", where the term 'dimensionality' refers to the length of the time series, i.e., the number of time steps in it. Simplification is done with the aim to ease processing, querying, and ultimately, visualization.

Shahar [Sha97a] defines temporal abstraction as "a process which, given a set of time-stamped parameters, external events and abstraction goals, produces abstractions of the data and interpret past and present states and trends that are relevant for the given set of goals". This definition emphasizes two things: (1) an abstraction must preserve important features, such as states and trends; (2) what features are important depends on the goals (in other words, analysis tasks).

The general requirements for data algorithms aimed at dealing with large data sets and use approximate representations of original data have been outlined by Faloutsos et al. [FRM94] as follows:

- they must be accurate despite the constraint of working with approximate representation of data;

- they must be carried out in main memory thus avoiding disk I/O overhead which is the principal bottleneck of any data-intense operation;
- they must be fast (i.e., computationally efficient).

The same requirements must hold in the context of visualization. That is, an abstracted data representation must not limit users in their actions required for analysis, nor negatively affect their interpretation of data. As stated by Faloutsos et al. [FRM94], it is acceptable for an algorithm (and thus potentially for a visualization, too) to result in false initial findings since they can (hopefully, rapidly) be discarded in later steps. But it is unacceptable if an algorithm or abstracted data visualization results in missing potentially useful findings.

The latter problem is of particular importance when producing aggregated or abstracted visualizations of data. It also strongly affects the choice of algorithms one would use for achieving this very abstraction. For instance, in some applications it might be necessary to preserve peaks of time series in their original form, or at least so that maximal values are never altered even in an aggregated view.

2.3. Data Space vs Visual Space Abstractions

Cui et al. [CWRY06] classify abstraction approaches into two groups, namely, abstraction in *data space* and abstraction in *visual space*. Their classification corresponds to the notion of an Information Visualization Pipeline [CMS99, Chi00]. Abstraction methods that operate in visual space are zooming [BSH94] and distortion [KLS00].

Keim [Kei02] provides a overview of various abstraction techniques including those operating in visual space.

Our focus is on abstraction in data space. Generally, data abstraction approaches include sampling [DE02], clustering [DGM97], segmentation [cFlCCm06], projection [Tor52], dimensionality reduction (i.e., reducing the number of variables) [YYS05], and others. However, these generic classes of methods may not be straightforwardly applicable to time series. Abstraction of time series requires specific approaches that respect the nature of this data type, particularly, the presence of temporal ordering relationships between the data items. Such approaches are developed in the field of data mining with the aim on generating representations of time series that are to be processed and analyzed by algorithms. In visualization, they can be used to perform abstraction of data prior to its visual mapping and display [LKL05].

Keim et al. [KMS*08] introduce a Scalable Visual Analytics mantra "analyse first - show the important - zoom, filter and analyse further - details on demand" which assumes application of computational processing methods to data prior to visualization. In particular , these may be methods for data abstraction.

2.4. Scope

In this paper, we review and systematize data abstraction methods that generate simplified representations of time series (the terms *time series simplification* and *time series abstraction* are used interchangeably), which can be useful for visualization purposes. We do not attempt to survey visualization methods or visual mappings of time series data. Thus, our focus is on transformations that occur in the early stages of the InfoVis Pipeline [CMS99, Chi00]. We pay particular attention to the temporal nature of time series critical for visualizing and analyzing such temporal data [AMM*08b]. Figure 1 summarizes the scope of our paper. In the next section we discuss other research in the area covered, and explain differences to, and contribution of, our paper.

3. Related Work

We structure the overview of the related works according to the topics and different foci of the publications:

- papers discussing the use of abstraction for visualization in general;
- papers proposing conceptual frameworks for time and time-referenced data;
- papers focusing primarily on visualization of time series but mentioning the use of abstraction;
- papers describing methods for time series transformation.

3.1. Data Abstraction in Visualization

A taxonomy for abstraction in information visualization is proposed by Ellis and Dix [ED07]. The authors do not distinguish between approaches that operate in data space versus visual space and devote more attention to visual methods. They give an overview of benefits and drawbacks of various clutter reduction techniques, but consider only general methods and not any data type-specific transformations.

Elmqvist and Fekete [EF10] propose a formal model for data abstraction in visualization, survey existing techniques, and derive guidelines for designing new ones. However, their focus is on hierarchical data aggregation rather than all possible abstractions in data space. Since only one abstraction technique is considered (aggregation), no mapping between techniques and data properties or user tasks is provided.

Du et al. [DSP*17] introduce an empirically derived taxonomy of analytical focusing strategies, that is, the ways in which users of visualization tools can interactively simplify the contents of a visual display and extract relevant information from it. The strategies include extraction of data subsets, pattern simplification (e.g., by grouping and merging), and partitioning. This work differs from ours mainly in two respects. First, it deals with a different type of data, namely, sequences of events, rather than numeric time series. The simplification methods described are not directly applicable to time series. Second, it describes interactive methods of

simplification, which are applied by users manipulating a visual display. Our paper discusses automatic methods of time series abstraction. The abstraction is supposed to be made prior to visualization by a visualization designer rather than end users.

3.2. Conceptualization of Time

Frank [Fra98] proposes a conceptual model that distinguishes between different types of time:

- **ordinal time:** time points happening one after another.
- **interval time:** every time point (event) is measured on an interval scale and has a length (duration).
- **cyclic time:** description of cyclic processes for whom application of an ordered relation is meaningless.
- **branching time:** events (even same ones) can occur in different branches (alternatives) that describe several scenarios or processes.

Although Frank suggests different approaches to visualize and analyze data based on their types, and his taxonomy provides a solid foundation of categorizing methods for analyzing temporal data, it is too general and falls short in the attempt to further classify types of tasks and possible methods in each of the derived categories. Despite the fact that the majority of approaches for visualizing time-referenced data consider ordered time (according to Frank's taxonomy), his work does not investigate this type any deeper than less common types of time that he describes.

3.3. Visualization of Time-referenced Data

Frank's taxonomy is taken into account by Aigner et al. [AMM*07] who propose a classification of visualization methods for time-oriented data based on three aspects: time, data and representation. This work provides a good example of systematization, similar to what we wish to achieve for temporal data abstraction methods. However, it should be taken into account that the purpose of the classification scheme proposed by Aigner et al. [AMM*07] was to provide a basis for a systematic overview of the state of the art in visualization rather than to support the choice of suitable methods. Our main goal is a classification supporting method selection (regarding abstraction methods). A more recent paper by Aigner et al. [AMM*08b] can serve as a guide for visual analysts to whom certain number of choices are presented. The authors discuss three aspects that they deem necessary to consider when generating visual representation of time-oriented data, namely: visualization, analysis methods, and user needs. They emphasize the importance of considering all three aspects in order to design an efficient visualization of time-dependent data and provide examples of how certain choices in each of the three aspects affect efficacy of visualization. However, although several possible solutions to challenges that arise in the visualization generating process are presented, their number is limited. For instance,
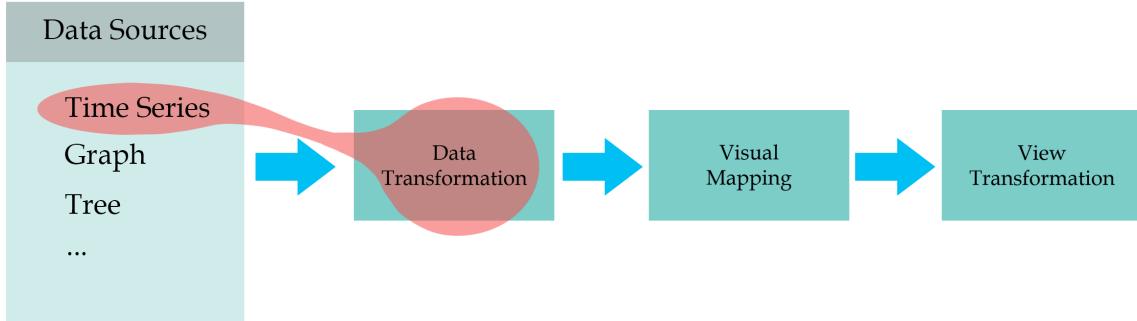
Figure 1: *Scope of this survey with regard to the main stages of the information visualization pipeline [CMS99, Chi00]. In contrast to previous state of the art reports (STAR) [LMW*15], we provide a refined view of a specific stage of the visualization generation process, specifically, the data transformation stage as shown on the diagram. We further narrow our focus by selecting time series data as the source, and develop our systematization according to properties of temporal data.*

when elaborating on analytical methods the authors distinguish between temporal data abstraction, principal component analysis and clustering, leaving out a variety of other approaches to obtaining abstracted representations of time series data, such as polynomials, spectral methods, segmentation, etc. When describing temporal data abstraction approaches, the authors admit that each of these have different levels of complexity and implications for data properties preservation. However, only a few examples of what they call complex temporal abstraction approaches are given.

The book by Aigner et al. [AMST11], which combines the classifications and descriptions from the previous works [AMM*07, AMM*08b], presents a survey of over one hundred time series visualizations. Some of these visualizations involve abstraction, but the book does not systematize the abstraction methods used. There is a general discussion regarding temporal data abstraction, by which the authors mean transformation of raw data to qualitative values, classes, or concepts. Vertical abstraction considers multiple variables over a particular time point and combines them into a qualitative value or pattern. Horizontal abstraction infers a qualitative value or pattern based on values from several consecutive time steps. Aigner et al. refer to some other works discussing qualitative abstraction, e.g., [CKPS10]. Our survey is not limited to qualitative abstraction but includes methods producing different types of representations.

Miksch and Aigner [MA14] propose a "design triangle" framework to be used in the creation of visual analytics methods for time-referenced data. They emphasize the importance of considering three main aspects: (1) the characteristics of the data, (2) the users, and (3) the users' tasks.

Examples of visual analytics systems designed for different data, users, and tasks are given, but no general guidelines regarding how to address these aspects. McLachlan et al. [MMKN08] describe a design process of a system for time series visualization where abstraction in the visual space is applied to show data at multiple levels of detail. Bernard et al. [BDF*15] apply user-centered design to create a system for visual search and exploration in a large set of time series in which abstraction in the data space (by means of time series clustering) is applied to facilitate the exploration.

Tominski [Tom11] proposes a general framework for user-centered visualization, called "event-based visualization", where the term 'event' refers to anything that can be of interest to a user. The main idea is that users specify their interests as event types, which are formally represented using predicate logic, and a computer system searches for instances of these event types in data and represents the results in visualizations tailored to the users' needs. The framework is not specific to time-referenced data but encompasses any data type; however, examples of detection and visualization of events in temporal data are provided. Extraction of events from data can be viewed as a kind of data abstraction; however, the author focuses on the visualization and does not provide references to specific event extraction methods.

There is a principal possibility of considering time series as a vectors in high-dimensional space, where each time step is treated as one dimension; hence, methods for visualization of high-dimensional data are potentially of interest for time series visual analysts. Liu et al. [LMW*15] provide an extensive overview of such methods. However, since the authors attempted to cover an extremely broad topic, their pro-

posed categorization appears to be on a very general level. They classify methods in accordance to the InfoVis Pipeline in its entirety. For instance, they put color blending methods [KGZ*12, HSKIH07] into the group of methods applicable on the *View Transformation* stage of InfoVis Pipeline, while dimensionality reduction [Jol02] is categorized into the *Data Transformation* stage. They further group methods based on their algorithmic nature, e.g. Subspace Clustering, Regression Analysis, etc. Although explanations for several algorithms in each group are given, no unified picture of the effects of selecting a particular approach is shown.

3.4. Transformation of Time-referenced Data

In time series mining literature [WL05, EA12] analytical methods are surveyed and classified, but with little to no mention of implications for visualization. Fu [Fu11] dedicates a section in the paper to visualization of time series data but does not draw any connections between mining methods that are mentioned and their effect on visualization tasks the end-user might need to perform.

Dozens of methods for simplified time series representation have been proposed in the field of database management and knowledge discovery [DTS*08, WMD*13]. Their design historically was mainly driven by the need to perform two essential operations in data mining, namely, querying and measuring similarity between objects [LKLC03].

Stacey and McGregor [SM07] consider complexity of patterns that temporal abstraction techniques are capable of conveying. Höppner [Höp02] classifies time series abstraction approaches into inductive (grouping similar parts), deductive (fixing shapes of interest in advance), and multiscale, which generate multiple abstractions at multiple levels. In this work, important implications on how abstraction techniques affect time series data are mentioned, but only few examples are provided, which do not allow for a unified picture of the rich variety of methods that exist. Roddick and Spiliopoulou [RS02] provide a survey of knowledge discovery approaches for temporal data and propose a classification framework based on identification of similarities. Classification dimensions include data type (scalar, unordered signals, etc), mining paradigm (algorithmic nature of the method), and temporal ordering. Existing methods are surveyed from the perspective of knowledge discovery rather than data abstraction.

Keogh et al. [LKLC03] proposed a classification of time series representation approaches which is shown in Figure 2. This classification organizes methods according to their algorithmic nature and type of output, but not by the analysis tasks for which these methods are suitable. It is also not clear what features of the original data are preserved by the methods and what could be lost.

Temporal abstraction has been an active area of research in medicine and clinical systems [VSP*07, SG-

BBT06, MS09]. Stacey and McGregor [SM07] provide a framework for classifying temporal abstraction methods based on the following criteria: data (medical sources like diabetes records, or heart rate), complexity of abstraction (whether abstraction is capable of preserving trends or more complex patterns like spikes in heart rate), number of variables (whether a temporal abstraction algorithm is capable of abstracting multivariate patterns), and reasoning (how knowledge is represented and conveyed to the clinician). Although the authors characterize temporal abstraction methods by their ability to preserve complex patters of time series and other features that we consider important for visualization, their survey only covers a handful of abstraction techniques. In fact, the authors admit that their work is not meant to be a complete coverage of temporal abstractions, but rather a guide for research directions in clinical intelligent data analysis systems.

Possible transformations of time series data are not limited to abstraction. It may be necessary to apply some transformations prior to abstraction to improve the quality of data and make them suitable for analysis [KHP*11, BRG*12]. Bernard et al. [BRG*12, Ber15] describe an interactive visual system for composing time series data preprocessing pipelines from a set of operations for data cleaning, reduction, normalization, segmentation, etc. The user can immediately observe the result of applying each operation to the time series. However, the paper does not propose a formal classification or taxonomy of methods, nor does it provide guidelines as to when the available methods would be suitable to use on the data and analysis tasks at hand.

3.5. Summary

To summarize, plenty of research has been done on categorizing time series visualizations and time series mining/abstraction techniques. Many works have proposed criteria for systematization of abstraction methods which could be useful to a visual analyst. However, there is no unified framework that would guide a visualization designer through the rich set of abstraction techniques. Our paper is aimed at filling this gap and providing guidelines for visual analysts who tackle the problem of visualizing large time series data. We characterize the existing temporal abstraction methods based on criteria driven by visualization tasks and end-user needs with regard to properties of data.

4. Framework for Characterizing Abstraction Methods

To create a systematic overview of time series abstraction methods, we first need to define the relevant aspects and criteria for characterizing methods. According to the "design triangle" framework [MA14], the important aspects to be accounted for in designing visual analytics methods are data, users, and users' tasks. These aspects may be relevant, in particular, to the choice of an abstraction method, which is a
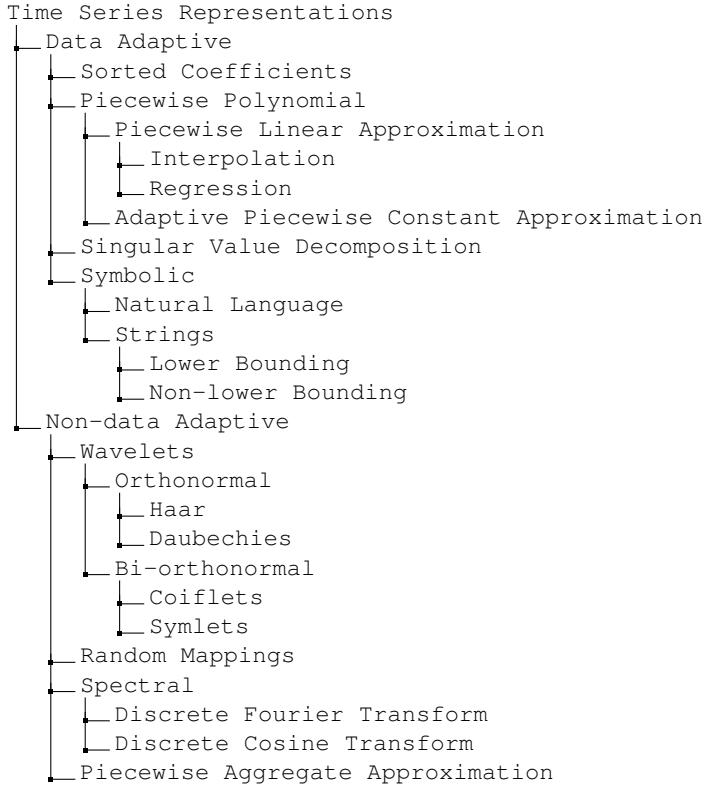
```
Time Series Representations
├─Data Adaptive
│  ├─Sorted Coefficients
│  ├─Piecewise Polynomial
│  │  ├─Piecewise Linear Approximation
│  │  │  ├─Interpolation
│  │  │  └─Regression
│  │  └─Adaptive Piecewise Constant Approximation
│  ├─Singular Value Decomposition
│  ├─Symbolic
│  │  ├─Natural Language
│  │  └─Strings
│  │     ├─Lower Bounding
│  │     └─Non-lower Bounding
├─Non-data Adaptive
│  ├─Wavelets
│  │  ├─Orthonormal
│  │  │  ├─Haar
│  │  │  └─Daubechies
│  │  └─Bi-orthonormal
│  │     ├─Coiflets
│  │     └─Symlets
│  ├─Random Mappings
│  ├─Spectral
│  │  ├─Discrete Fourier Transform
│  │  └─Discrete Cosine Transform
│  └─Piecewise Aggregate Approximation
```

Figure 2: *Classification of time series representations based on Lin et al. [LKLC03]. The leaf nodes are representations and the internal nodes are classes.*

critical step in the design of visualization and analytic methods. Let us consider to what extent each of these aspects may be relevant to choosing an abstraction method.

4.1. Data

According to the existing frameworks [AMM*07, AMM*08b, AMST11, MA14], the essential characteristics of temporal data include *scale* (quantitative vs. qualitative), *frame of reference* (abstract vs. spatial), *kind of data* (events vs. states), *number of variables* (univariate vs. multivariate), *time arrangement* (linear, cyclic, or branching), and *time primitives* (instant, interval, or span). Not all of these characteristics are relevant to the scope of our survey, which considers abstraction methods applicable to numeric time series, that is, where the scale is quantitative, the frame of reference is abstract, and the kind of data is states. The distinction according to the number of variables is important. As we are not aware of time series abstraction methods that specifically address cyclic or branching time, the distinction according to time arrangement does not apply. The same refers to the nature of the temporal primitives: the existing abstraction methods are agnostic of this characteristic. Most of the methods even do not take

into account the temporal references as such but treat time series as mere linearly ordered sequences of values. It is very hard to evaluate the existing abstraction methods with regard to the distinction between time instants and intervals because the papers describing the methods do not discuss this issue. That is why we do not include this characteristics in our classification framework.

There are other characteristics of data that appear essential for choosing appropriate abstraction methods. One of them is whether all data are available at once (stationary data) or new data arrive over time (streaming data). The latter case requires special algorithms that process data incrementally, while most of the existing methods can only be applied to a whole dataset.

Time series may be unevenly spaced, i.e., the time intervals between consecutive observations may vary, which needs to be accounted for in data analysis. To deal with such time series in visualization, Aris et al. [ASP*05] propose to apply sampling of events at regular intervals, aggregation, or special display techniques that represent time in a non-linear way. Event sampling (a.k.a. re-sampling in the data mining literature) requires an algorithm that determines what value to use when there is no value for some time moment in the

original time series. Aris et al. [ASP*05] take the most recent value. In data mining and time series analysis, very few methods exist that can specially deal with uneven time series [Eck14, Eck17]. This also refers to the time series abstraction methods, which mostly implicitly assume equal time spacing between observations and, as we mentioned earlier, typically take into account only the value ordering but not the temporal references. Before applying these methods, uneven time series need to be transformed to equally spaced. A common approach is to apply some method of interpolation. The interpolation methods for time series are surveyed by Adorf [Ado95]. These methods, however, can introduce biases in the data. Beygelzimer et al. [BEMR05] propose instead approaches to representing uneven time series by statistical models, which can then be used for reconstructing values at equal time intervals.

Since only very few time series abstraction methods explicitly deal with uneven spacing of time series, we do not include this aspect in the overall framework for method classification but instead refer to these few methods directly here. The main recommended method is exponential moving average (EMA) [Mül91, DGM*01], as cited by Eckner [Eck17]. The latter proposes several other techniques belonging to the class of so-called rolling time series operators, which allow to extract a certain piece of local information about a time series within a rolling time window of a fixed length. Essentially, the author proposes a generalization of the simple and exponential moving average operators.

The other time series abstraction methods surveyed in our paper assume even spacing of time series. They can be applied to uneven time series after re-sampling by means of the existing interpolation or modeling methods [Ado95, BEMR05].

An aspect also requiring consideration is data quality: poor quality can affect not only the applicability of abstraction and analysis methods but also the possibility of accomplishing user tasks [SLW97]. For instance, if recordings are corrupt, correlations between different observed variables might be distorted and lead to wrong conclusions [HSW07]. Gschwandtner et al. [GGAM12] give a comprehensive overview of types of data problems that may occur, including missing data, duplicates, implausible or outdated values, etc. Most of these possible problems are not directly relevant to selecting methods of abstraction, visualization, and analysis. They rather need to be fixed at a prior stage by changing formats, aligning fields, removing duplicates, correcting implausible values, finding up-to-date data, etc. Methods used to deal with data issues are mostly mechanical, or low-level. It is admitted that the connection between simplification and the ability to notice or detect dirty data is rather doubtful [KHP*11, HSW07]. Instead, it is visualization of raw data that can help identify problems with it [KHP*11] rather than simplification or abstraction methods applied prior to visualization.

Data quality issues are, of course, important in any real-world analysis setting. We see dealing with such quality issues as a distinct and prior phase of analysis, i.e., exploratory data analysis where the analyst wants to confirm expected patterns, but may also wish to discover unexpected patterns. We uphold that exploration of data quality issues entails searching for particular kinds of patterns [KHP*11, AAF16], such as temporal gaps with missing data. Obviously, simplification must not eliminate such patterns, e.g., by using interpolating methods that fill holes in data coverage – or, if they do, visual mapping should at least ensure that artificial values are recognizable as such.

A special note needs to be made concerning the noise in data. In time series analysis, it is acknowledged that data may have irregular fluctuations (e.g., [DB16]), that is, noise is treated as an indispensable component of time series. Methods for time series analysis, including the abstraction methods, are developed under the assumption that irregular fluctuations may be present. Hence, it can be expected that the existing abstraction methods will generally cope with noise in data. Moreover, abstraction reduces the amount of noise and thus helps to reveal trends and regularities. Still, if the noise results from frequently reoccurring measurement errors, the revealed trends and regularities can hardly be trusted. Therefore, significant errors in data need to be detected and corrected before applying abstraction.

Methods and workflows for data preprocessing, in particular, resolving data quality issues are proposed in several works [KHP*11, BRG*12, Ber15]. Assuming that data quality issues have been previously resolved, the distinguishing characteristics of data that affect the choice of an abstraction method are ***dimensionality (univariate*** or ***multivariate)*** and whether the data are ***stationary or streaming***.

4.2. Users

The aspect "users" refers to the necessity of accounting for the users' capabilities, mental models, as well as established practices and conventions in the application domain [MA14]. By its essence, data simplification addresses users' capabilities as its main goal is to facilitate users' perception and understanding of large data. However, simplified data do not go to the end user directly but they need to be represented visually. Therefore, there is no direct link between the user characteristics and the choice of an abstraction method. The primary choice to be made by a visualization designer is a suitable method for the visual representation of time series, which should correspond to the needs, characteristics, and expectations of the users. Only then the designer selects a time series abstraction method that supports the chosen visual representation.

In this respect, the most important characteristic of an abstraction method is the type of output, i.e., the computer-oriented ***representation*** of abstracted time series, which may

be *numeric*, *symbolic*, or have the form of a *functional model* or *rules*; the latter may describe relationships between several variables (e.g., what values of different variables tend to occur together) or temporal relationships between patterns occurring in time series (e.g., what patterns frequently occur one after another). The computer-oriented representation must be compatible with the chosen visual representation. Thus, transformation to the symbolic form or rules does not allow subsequent visualization of abstracted time series by line plot, bar graph, and similar methods [AMST11]. Generally, the visualization methods that involve mapping of numeric values to display dimensions or retinal visual variables [Ber83] require a representation that either consists of numeric values or, as a functional model, allows obtaining such values for given time references. Symbolic representation is compatible with display types similar to EventFlow [DSP*17], where categories are encoded by colors, or SparkClouds [LRKC10], where text labels are accompanied by sparklines (tiny line plots) showing the times of occurrence of the texts. Representation by rules is compatible with visualizations that focus on showing relationships, such as node-link diagrams, arc diagrams, and matrix views [HBO10]. Such displays do not explicitly involve time. Rules describing re-occurring pattern sequences in time series are suitable for visualization in the form of state transition graphs [BBG*09, AA17].

One more type of output of an abstraction method is *clusters* of similar time series. This kind of abstraction is applied when the user is supposed to explore a large number of univariate time series with a common variable. It has also been applied to bivariate time series where two variables have comparable value ranges [SBvLK09] and to multivariate time series after transforming the original values of the variables, which were incomparable, to z-scores [AAB*10]. Multiple time series may result from dividing one very long time series into segments of equal length, e.g., by daily or weekly time intervals [VWVS99, SBvLK09]. Clustering reduces a large number of time series to a much smaller number of representative time series of the clusters, which can be easier explored by users. The visualization needs to be designed so that the users are able to see the representative time series and the distribution of the clusters over the whole dataset. For example, a calendar display [VWVS99] shows the temporal distribution of clusters over a year, a matrix display [SBvLK09] shows the distribution over a long time period and a set of objects, and geographic maps [AAB*10, vLBR*16] can show the distribution of the clusters in the geographic space.

4.3. Users' Tasks

In order to perform simplification in an appropriate way, a visualization designer needs to know what is important to the user so that valuable information that is essential for analysis is not lost [EA12]. The answer to this question lies in tasks

that the user plans to perform on simplified data [AMST11]. User's ability to accomplish them should not be impeded by the simplification.

Numerous task taxonomies exist, e.g., [Shn96, AES05, BM13]. We are particularly interested in those where tasks are defined in terms of data (since our focus is abstraction in the data space) rather than in terms of user's activities performed through a visual display. First of all, we have looked for task taxonomies dedicated specifically to time-referenced data. In the book by Aigner et al. [AMST11], the following groups of tasks are proposed:

- **Classification**: Given a predefined set of classes, determine which class a data item belongs to.
- **Clustering:** Grouping data into clusters based on some measure of similarity.
- **Search and retrieval:** Locate exact or approximate matches to a given example in a large collection of data.
- **Pattern discovery:** Find interesting patterns, such as sequential, periodic, or associative, without having any a priori assumptions.

This list can be supplemented with additional groups of tasks that can be found in the literature on time series analysis [LKLC03, Moe06, Fu11, EA12]:

- **Segmentation:** Divide time series into segments (internally homogeneous continuous sets of observations).
- **Subsequence searching:** Find continuous sets of observations that correspond to some constraints.
- **Motif discovery:** Find repeated occurrences of similar individual time series or subsequences.
- **Anomaly detection**: Find observations that are rare and stand out excessively among the neighboring observations.

All these tasks have been originally defined for automated analysis using methods of data mining and machine learning. Not all of them can be treated also as users' tasks. Thus, clustering is an important instrument of analysis and a tool for data abstraction, but this can hardly be a task that an end user may primarily wish to perform. In other words, clustering may be the means but not the goal of analysis. Similar considerations refer to segmentation. Classification is meant for automatic assignment of data items to classes rather than for helping users to understand data; hence, this is also not a typical task of a user.

The remaining classes of tasks refer, in this or that way, to *patterns* that may exist in time series. Users may wish to find particular patterns of interest specified by examples or by constraints, which corresponds to tasks 'search and retrieval' and 'subsequence searching'. Users may also wish to discover unexpected patterns. Pattern discovery is a kind of task that can be effectively fulfilled by humans supported by appropriate visualizations. Humans can utilize their unique capabilities for pattern recognition, which are not yet equaled by computers. The task of motif discovery may also call for

human capabilities when there is no predefined similarity measure that can be computed. In fact, motif discovery is a subtype of the pattern discovery task type, where it is necessary to discover re-occurring patterns. Anomaly detection can also be considered as discovery of a particular kind of pattern, namely, dissimilarity of some observations to others. Hence, it can be concluded that notion of *pattern* is key for the tasks of pattern discovery, motif discovery, and anomaly detection, and it is also relevant to search tasks.

The event-based visualization framework by Tominski [Tom11] assumes that users specify their interests as 'event types', and computer searches for instances of these event types in data; hence, the framework focuses basically on search tasks. In a more general taxonomy of tasks in exploratory data analysis [AA05], users' tasks are defined based on data structure. Data components are categorized into independent and dependent variables, called references and attributes, respectively. The authors propose to view data as a representation of a function that matches references to attributes. The general aim of data analysis is studying the behavior of this function. There are four classes of synoptic analysis tasks (i.e., addressing the function behavior rather than individual data items):

- **Behavior characterization**: describe the behavior of one or more attributes.
- **Pattern search:** locate a particular behavior, i.e., find subsets of references where attributes have this behavior.
- **Behavior comparison:** identify similarities and differences between two or more behaviors.
- **Relation seeking:** find subsets of references for which a particular relation ('same', 'different', 'opposite', etc.) exists between the behaviors of two or more attributes.

The authors of the taxonomy also use the notion of *pattern*, which is defined as a construct reflecting essential features of a behavior in a parsimonious manner, i.e., substantially shorter and simpler than describing each individual data item. Thus, "to characterize a behavior" means to represent it by one or several patterns (which corresponds to the pattern discovery task in the previously discussed taxonomies); the other classes of tasks can also be related to the notion of pattern. The definition of a pattern is similar to that adopted in data mining, where a pattern is defined as an expression in some language describing a subset of facts without enumerating all these facts [FPSS96]. The definition proposed for exploratory data analysis [AA05] has a broader scope, also including representations in human's mind.

In both definitions, 'pattern' is a representation (constructed by a human or a computer) of something that objectively exists in the studied behavior, i.e., "essential features of a behavior" [AA05]. Accordingly, if a task needs to be fulfilled by a human with the help of visualization, the visualization must convey these essential features for enabling the human to construct appropriate patterns. Consequently, if data transformation, such as abstraction, is per-

formed prior to visualization, the essential features must be preserved. Some transformation methods may ruin this or that kind of features. A visualization designer must be aware of this when selecting an abstraction method. The designer needs either to anticipate the kinds of features that may exist in the studied behavior and choose a method that preserves them, or needs to provide several methods that preserve different features.

Please note that we use the term 'feature' to refer to prominent parts or essential characteristics of real-world phenomena and processes. This is different from the more technical usages of the term in machine learning, such as 'feature vector', 'feature space', and 'feature engineering'.

Hence, by considering the types of users' tasks that require visualization support, we came to the conclusion that all these tasks involve generation of patterns, which need to faithfully represent essential features of the studied behavior. Therefore, preservation of features that can exist in a behavior is a paramount criterion for selecting appropriate data abstraction methods, whereas the types of tasks that need to be fulfilled are not directly relevant to the method choice.

The essential features of time-variant behaviors that are most frequently referred to in literature on time series analysis [EA12, Kle15, DB16] are *trends* and *seasonality* (a.k.a. *periodicity* or *cyclical variation*). A trend is a long-term increase or decrease of values. Cyclical variation means regular re-occurrence of some behavior along with repetition of some time cycle, such as daily, weekly, annual, or a domain-specific cycle (e.g., in astronomy or in economy). The term 'seasonality' usually refers to the annual cycle.

In the context of visualization, important features of time-variant behaviors are also *events*, that is, significant changes [AMM*08b, AAM*10], including peaks, drops, and trend changes. *Outliers*, or *anomalies* are a specific kind of events when some observations greatly differ from the preceding and following observations. Outliers need to be considered as a separate type of feature since many abstraction methods involve data smoothing, which destroys outliers. Hence, if users are interested in detecting outliers, such methods should not be used.

To summarize, in selecting abstraction methods, it is necessary to take into account what features may exist in the studied behavior and which of these features the users are interested to detect and analyze. The types of features are: **trend**, **cyclical variation**, **event**, and **outlier**. It is important to choose methods that do not destroy essential features expected to be present in the studied behavior and/or are relevant to the goal of analysis. Further, it may be beneficial to use methods that detect and *extract* the features of interest.

4.4. Properties of Algorithms

The "design triangle" [MA14] sets external requirements for choosing an abstraction method, which must be suitable for

the data, produce a representation matching the users' mental models and established practices, and preserve essential features of the behavior that will be studied. However, for an informed selection of a method, it is also necessary to understand some internal characteristics of the algorithms. We do not attempt to evaluate the complexity or execution speeds of algorithms, as these depend on implementation details often omitted in original papers [KK03]. Instead, we consider two properties: involvement or possibility of indexing, which helps speed up data access, and involvement of partitioning of the time series. In data mining and in the context of our paper, the term *indexing* denotes creation of special data structures that facilitate and speed up searching and retrieval of information in response to queries. This is different from some usages of this term in visualization literature [Ber83, HBO10, AKMM11].

Indexing is very important in data mining because it allows efficient similarity search, which, in turn, is a subroutine to other data mining tasks, such as clustering and classification. From the perspective of visualization, indexing may be beneficial for efficient implementation of interactive operations, such as dynamic querying and highlighting. Hence, when these operations need to be enabled, it may be reasonable to choose an abstraction algorithm that builds an index structure. Another possible approach is application of a common indexing method, such as R-tree, SB-tree, etc., to the output of an abstraction method. In this case, the form of the output must be suitable for applying the indexing method.

A simplified representation of a time series produced by an abstraction algorithm is often called a *model* in data mining literature. A model of a time series can represent the time series in its entirety, or a model may consist of parts representing segments of the time series. The latter type of model is called *piecewise*. For obtaining a piecewise model, a time series may be divided into segments of equal length, which is simpler, more efficient, and better suited for indexing, or into segments of variable length, which enables more accurate representation of behavior features but is more complex in terms of indexing and querying. The involvement of *partitioning* and the way of partitioning (equal or variable segment length) affects, on the one hand, model accuracy, and on the other hand, algorithm complexity and efficiency. A visualization designer should choose a method depending on the required level of detail in representing time series as well as on characteristics of the display device, particularly, pixels resolution. It may be required to vary the level of detail in response to interactive zooming. In this case, the designer may consider using different abstraction methods for different zoom levels, or choose such a method where the level of detail of the output can be regulated through parameter settings. For this purpose, piecewise equi-length modeling methods can be more convenient since the number of segments in which they will divide time series is specified as a parameter.

Hence, we include in our classification framework the following two properties of the abstraction methods: (1) involvement or possibility of **indexing** and (2) involvement and the way of **partitioning**.

5. Methods Classification

Based on the previous argumentation, we classify time series abstraction methods according to the following facets:

- Data properties:
  dimensionality: univariate vs. multivariate
  form of availability: stationary vs. streaming
- Users' mental models and practices:
  representation form
- Users' tasks:
  preservation and extraction of **behavior features**
- Algorithm properties:
  indexing
  partitioning

For the ease of navigation among the papers included in the survey, we also pay attention to the *type of paper* in which a method is described. **System** papers present abstraction techniques as parts of analysis or visualization systems. These papers are not entirely focused on the abstraction methods they use but dedicate significant part of the description to other topics. **Method** papers are dedicated entirely to the problem of proposing novel approaches to modeling, representing or abstracting time series.

5.1. Data Properties

5.1.1. Dimensionality

Univariate. A majority of the abstraction methods focus on singular univariate time series and attempt to derive their models by considering one series at a time. In principle, this does not disqualify these methods from applying to multivariate time series. As we noted earlier, multivariate time series can be represented as combinations of univariate time series, one per variable. An abstraction method can be applied to each of these time series. The performance can be improved by parallelizing this process.

When there is a large number of time series sharing the same variable (univariate time series) or the same combination of variables (multivariate time series), clustering may be applied to reduce the number of time series by grouping similar series and taking a representative time series from each group. K-means [GSBO13] and spectral clustering [NJW02] methods have been utilized for this purpose and resulted in much simpler representations although the choice of appropriate parameter settings may be difficult.

Multivariate. Yang and Shahabi [YS04] propose a PCA-based similarity measure for multivariate time series, as well as an indexing structure [YS05]. For dimensionality reduction of multivariate time series, Yoon et al. [YYS05] propose

a PCA-based approach that aims at preserving correlation between the variables of the time series. On top of these methods, an efficient k-nearest neighbor search approach over multivariate time series [YS07] is built. Smyth [Smy97] proposes clustering sequences of time series with Hidden Markov Models.

5.1.2. Form of Data Availability

Stationary. The majority of algorithms can only be applied to the whole dataset. It is not appropriate to apply them straightforwardly to portions of streaming data because the algorithms would treat them independently and ignore continuity of the data.

Streaming. Some algorithms are capable of abstracting data as it arrives. In most cases, data items from a certain interval are buffered and continuously incorporated into a currently existing approximation obtained from previously processed items.

5.2. Representation Form

Numeric values. A simplified model of a time series is a sequence of numeric values of a shorter length than the original sequence. Methods producing such abstractions include, for instance, Piecewise Aggregate Approximation [KCPM01].

Symbolic. A numeric time series is represented by a sequence of symbolic strings. This representation enables the use of some analysis methods that cannot be applied to other representations [LKLC03], for example, derivation of decision trees.

Functional Model. A time series is approximated by a linear combination of several representative functions. The more functions are used, the more accurate the model is; however, the complexity increases and the degree of abstraction decreases. An example of this approach is singular value decomposition [KJF97].

Rules. Rules derived from temporal data can represent relationships between multiple variables [Sha97a, VSP*07, ACD*06, Sta09]. For example, in clinical data several variables can be combined to create an abstraction that describes the state of a patient [SM96]. Such representations can be automatically processable and also directly readable by end-users (e.g., physicians).

Clusters. Clusters of time series are generated by grouping together sequences that are similar in respect to a predefined similarity measure.

Besides these primary forms of the output, many methods compute various statistical characteristics of time series, which may be useful in analysis. Therefore, our summary table (Table 2) includes a column labelled **'Statistics'**, in which the methods producing statistical descriptors of time series are marked.

5.3. Feature Preservation and Extraction

Trends. This label refers to algorithms that can detect and extract trends and generate models that consist entirely or mainly of the trends derived from the original time series.

Events. This label refers to approaches that produce models based on events of interest, usually user-predefined.

Outliers. The group of algorithms suitable for outlier detection somewhat intersects with the previous group but is conceptually different, because its parameter setting is of different nature. In case of event extraction, the user is expected to specify criteria for defining events, while for outliers detection the user specifies the normal behavior, and the algorithm extracts data that do not qualify as such.

Cyclic variation. Detection and analysis of periodicity in time series is important in many fields [EAE05, RAA11, HDY99]. Wang et al. [WMD*13] claim that spectral methods (DFT [AFS93], DCT [KJF97], etc) are slightly better at grasping periodicity of data and allow more compact and more accurate representation for abstraction of time series in comparison to polynomial methods such as SAX [LKLC03] or APCA [CKMP02].

5.4. Method Properties

5.4.1. Indexing

Full-fledged indexing. Many algorithms produce representations that are easily indexed by common indexing structures like R-trees, SB-trees, Binary trees, etc. Some methods build index structures during the approximation phase.

Limited indexing. It is possible that an abstraction method is only capable of approximating series of certain lengths, or produces outputs that cannot be indexed in an efficient way. For example, discrete wavelet transform [CF99] can only be applied to time series with a lengths of integral powers of two.

No indexing. Abstraction methods may produce models that cannot be mapped to any index structure.

5.4.2. Time Series Partitioning

Entire-length. This label refers to methods that apply simplification or feature extraction on the entire length of time series and derive a model accordingly. For instance, Discrete Wavelet Transform decomposes the entire signal into a combination of Wavelet bases (Fig. 3).

Piecewise Equi-length. The most intuitive and basic approach is to divide the time series into segments of equal length and then finding appropriate representations for these regions. This approach facilitates indexing since an underlying indexing structure does not have to be complex [CKMP02]. Piecewise Aggregate Approximation
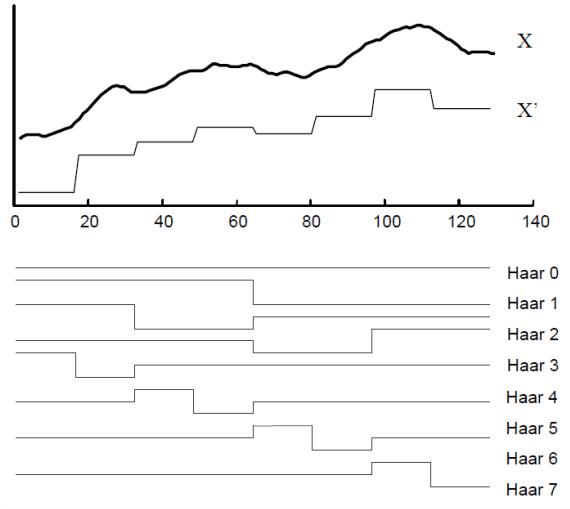
Figure 3: *The first eight wavelet bases and their linear combination X′ to represent the original data X. [CF99]*
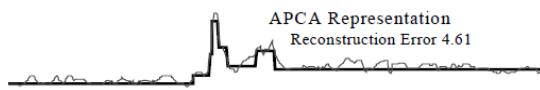

Figure 4: *Visual output of the Piecewise Aggregate Approximation algorithm [CKMP02]*

[KCPM01] is one of the most intuitive and simplistic approaches that produce equi-length segments. The idea behind the algorithm is to divide data of length n into m equi-sized frames, where $1 < m < n$. The mean value of the values that fall within boundaries of each frame form the vector of length m which becomes the reduced representation of the original data. Piecewise Constant Approximation [KP00] algorithm is ultimately the same approach towards reduction that has different implications in indexing.

Piecewise Adaptive-length. A more precise representation of the original data is possible if segments of variable length are allowed. However, these approaches may result in approximations that are more difficult to index and query. Adaptive Piecewise Constant Approximation presented in [CKMP02] approximates the time series by segments of constant values with varying lengths (Fig. 4). The authors demonstrate that the algorithm provides accurate representation and is indexable, that is, capable of finding neighbors.

6. Detailed Summary

Table 1 gives a brief overview of the surveyed abstraction methods. Table 2 summarizes the abstraction methods in respect to the properties, which are organized into groups, or facets, according to the classification scheme presented in Section 5. This view fulfills several purposes.

1. Allows fast and easy selection of abstraction algorithms.

The visualization designer needs to consider the aspects discussed in Section 4 and then choose an algorithm that satisfies most or all of the requirements in terms of the data properties, output representation, feature preservation and extraction, and method properties. For example, if the goal is to design an event-driven visualization system with interactive querying capabilities and line chart representation, the visualization designer should look for algorithms that satisfy the *Events* property, with *Symbolic* or *Real-valued* representation, preferably of *Piecewise-Adaptive Length* type and with *Indexing* support. It may happen that some facets are not important for a particular design and can be ignored. For example, in case of designing a system for offline analysis, both batch and online algorithms are suitable.

2. Provides examples for classification of new algorithms.

The proposed table can be easily extended to include approaches that are not covered in this survey as well as new approaches that continuously appear. A method that is added needs to be evaluated with regard to each group of properties specified in the table header. Note that several properties at once could be satisfied within each facet.

3. Shows principles behind the organization of the properties into groups.

Understanding these principles is a prerequisite for introduction of new meaningful dimensions in the future. We identify two key requirements for including a new group of properties into the framework. First, the properties in the group must be inclusive, that is, every algorithm must satisfy at least one of the properties in each group. Second, they must be informative. A visualization designer should be able to clearly understand what they lose or gain when they select an abstraction algorithm that satisfies particular properties and fails for others.

7. Future Work

Additional aspects of time series modeling methods can also be considered for making an informative choice of abstraction algorithms. Some groups of categories could be good candidates for extending the existing framework. However, due to the limited amount of details in published method descriptions, it is hard to assess the methods in respect to the following categories [WMD*13].

Reference	Description
Keogh & Pazzani [KP00]	A time series is divided into equally sized segments, from which mean values are taken.
Agrawal et al. [AFS93]	In applying Discrete Fourier Transform, only the first few frequencies are taken, so that the data size is reduced.
Fu et al. [cFlCCm06]	Produces a binary tree representation of a time sequence with each tree node being one of the perceptually important points (PIP). A PIP is the most distant point from a line connecting two arbitrary points in the sequence.
Chakrabarti et al. [CKMP02]	APCA: Divides a time sequence into segments of varying lengths so that the value range in each segment is minimal and takes the mean of each segment (Fig. 4).
Lavrenko et al. [LSL*00]	Identifies trends by top-down fitting of regression lines in a greedy manner using the t-test as a stopping criterion.
Lin et al. [LKLC03], Lkhagva et al. [LSK06], Fuad & Marteau [FM13]	Normalize, then divide a sequence into segments of equal length and assign labels based on the discretization. These representations are specifically designed to allow definition of distance measures.
Keogh et al. [KCHP01]	An online algorithm that accumulates a point sequence in a sliding window and then performs bottom-up segmentation.
Park et al. [PKC01]	Within an expandable time window, connects the first and the last points and checks the deviation from the original data against a threshold. Tries to expand the window until the threshold is exceeded.
Hunter & McIntosh [HM99]	Within an expandable time window, builds a regression line and checks the deviation against a threshold. Tries to expand the window until the threshold is exceeded.
Zhu et al. [ZWL07]	To produce a piecewise linear representation, finds appropriate sampling intervals taking into account the curvature of data segments.
Jiang et al. [JZW07]	Segmentation based on feature points similar to PIP in [cFlCCm06].
Himberg et al. [HKM*01]	Produces segmentation in a top-down manner but allows flexible breakpoints as opposed to classical approaches. Results in a better accuracy and remains efficient.
Fitzgibbon et al. [FDA02]	Coding a signal into messages of as small lengths as possible by carefully selecting parameters for data distribution agreed between the sender and the receiver.
Xu et al. [XZKP12], Fuchs et al. [FGNS10]	Polynomial approximation of a temporal sequence.
Hatwar & Badhiye [HB15], Dan et al. [DSDH13]	Based on the trend types, time series are segmented into pieces of different lengths, which are labeled with symbols.
Wang et al. [WSH05]	Time series are characterized and clustered based on the characteristics obtained.
Keogh & Pazzani [KP98]	Produces a piecewise linear representation by dividing the sequence into vectors and assigning a weight to each to indicate its importance.
Bingham et al. [BGH*06]	Application of modified PCA methods to reduce the dimensionality of time series.
Eads et al. [EHD*02]	Feature extraction for further classification of time series using support vector machines.
Kalpakis [KGP01]	A Linear Predictive Coding approach is introduced for clustering ARIMA time series.
Yoon et al. [YYS05]	A set of unsupervised methods for selection of key features that can compactly describe a process.
Smyth [Smy97]	Clustering of time series using hidden Markov models.
Shahar et al. [Sha97a], Verduijn et al. [VSP*07], Shahar et al. [Sha97b]	Frameworks for deriving rules that describe relationships between different segments within a single single series and between different time series.
Ghoniem et al. [GSBO13]	Spectral clustering based on cosine similarity and other metrics of multivariate time series is used to reduce the dimensionality and identify features of interest.
Chan et al. [CF99]	Represents a time series as a combination of wavelets (Fig. 3).
Korn et al. [KJF97]	Finds a representation in a feature space of a lower dimensionality.
Megalooikonomou et al. [MWLF05]	Vector quantization is used to generate a codebook of subsequences. Codebook keys are then matched to data and used for representation of original data at different resolutions.
Batal et al. [AFS93]	Time series are segmented and qualitative descriptions of each segment are obtained. Descriptions are predefined abstract states. Frequency of states, their length and relationship are then used as features based on which new vector representation of data is generated.

Table 1: *Brief descriptions of the surveyed data abstraction methods.*

Reference	Univariate	Multivariate	Streaming	Stationary	Symbolic	Model	Rules	Clusters	Numeric values	Statistics	Trends	Events	Outliers	Cyclic variation	Indexing	Limited indexing	No indexing	Entire-length	Pcws. Equi-Len	Pcws. Adapt. Len	System	Method
	Dim.		Data		Representation						Features				Indexing			Partitioning			Paper	
Keogh & Pazzani [KP00]	●			●					●		●			●	●				●			●
Agrawal et al. [AFS93]	●			●	●						●			●	●			●				●
Fu et al. [cFlCCm06]	●			●					●		●		●		●					●		●
Chakrabarti et al. [CKMP02]	●			●					●			●	●		●					●		●
Lavrenko et al. [LSL*00]	●			●	●						●	●					●	●			●	
Lin et al. [LKLC03]	●		●								●	●	●		●				●			●
Lkhagva et al. [LSK06]	●			●							●	●	●		●				●			●
Fuad & Marteau [FM13]	●			●	●						●	●			●			●				●
Keogh et al. [KCHP01]	●		●	●					●		●					●			●			●
Park et al. [PKC01]	●		●	●				●			●					●			●			●
Hunter & McIntosh [HM99]	●			●				●			●				●			●			●	
Zhu et al. [ZWL07]	●			●		●		●			●			●			●			●		●
Jiang et al. [JZW07]	●			●				●			●					●		●				●
Himberg et al. [HKM*01]	●			●				●			●						●	●				●
Fitzgibbon et al. [FDA02]	●			●				●		●	●						●	●				●
Xu et al. [XZKP12]	●		●	●		●					●						●		●			●
Fuchs et al. [FGNS10]	●		●	●		●					●						●		●			●
Hatwar & Badhiye [HB15]	●			●	●						●			●	●					●		●
Dan et al. [DSDH13]	●			●	●						●			●	●					●		●
Wang et al. [WSH05]		●		●				●			●			●			●	●				●
Keogh & Pazzani [KP98]	●			●				●				●				●		●				●
Bingham et al. [BGH*06]	●	●		●		●					●		●				●	●				●
Eads et al. [EHD*02]	●	●		●					●	●			●				●	●				●
Kalpakis [KGP01]	●		●	●				●				●	●					●	●			●
Yoon et al. [YYS05]		●		●				●		●			●	●				●				●
Smyth [Smy97]		●		●				●		●	●		●					●			●	
Shahar et al. [Sha97a]		●		●			●			●						●		●			●	
Verduijn et al. [VSP*07]		●		●			●			●						●		●			●	
Shahar et al. [Sha97b]		●		●			●			●						●		●			●	
Ghoniem et al. [GSBO13]		●		●				●					●				●	●			●	
Chan et al. [CF99]	●			●		●					●						●	●				●
Korn et al. [KJF97]	●			●				●			●		●		●			●				●
Megalooikonomou et al. [MWLF05]		●		●	●						●		●		●				●			●
Batal et al. [BSBH09]		●		●	●						●		●				●			●		●

Table 2: *Data abstraction methods are assessed according to the dimensions defined in Section 5.*

7.1. Error Boundaries

Our study of the literature has revealed that a lot effort is being put in deriving quality metrics for data abstractions [BTK11,CWRY06]. It is important to be informed about the degree of abstraction to avoid oversimplification and loss of important features. An error bound guarantee is achieved by assuring that an algorithm will not produce approximations differing from the original data by more than a specified value.

Global error boundary. This is a guarantee that the sum of all errors across the entire length of a time series will not exceed a given value. Piecewise Polynomial Representation [FGNS10] approximates time series segments with polynomials of arbitrary degree. By resorting to the least squares approximation over a sliding/growing time window, the authors achieve an online algorithm with a global error boundary.

Individual error boundary. This is a guarantee that any point in the approximated data will not differ from the corresponding point in the original data by more than a specified value. Discrete Fourier Transform [AFS93] was proved to hold the lower bounding condition for its approximation.

No error boundary. In some cases, it turns out to be difficult to provide a proven error boundary although an algorithm may perform remarkably well compared to other approaches for which error boundaries have been stated.

One aspect that is often considered as a quality measure is how accurately the model preserves the distances between objects. However, some researchers claim that this measure is not sufficient for temporal data [LP11]. A deeper investigation and evaluation of quality measures for temporal data should be carried out.

7.2. Parameters

Algorithms might require some parameterization that strongly affects the output.

Number of segments. Determines the level of accuracy of the output for algorithms involving time series partitioning. Piecewise Constant Approximation [KP00] proposed by Keogh et al. takes k as the only input to determine the number of equi-sized segments the mean values from which will be taken for the representation.

Resolution. In a case when an algorithm produces a global approximation of time series, the user has to provide, instead of the number of segments, the number of the first coefficients, or the degree of polynomials, or the number of eigenwaves to be used in modeling the original data. This parameter can be referred to as the resolution of the abstraction.

Error bound. This is often an additional parameter, but it may also be the only parameter required by an algorithm.

Approaches that consider the total error bound calculate the sum of residues at all data points. If it exceeds the specified value, the previous approximation is refined so that the the bound is met. Unlike a total error bound, that considers the sum of all errors, an individual error bound is aimed at preserving the distance between each original data point and its approximation at a certain value which must not be exceeded.

Non-parametrized. Some algorithms do not take any parameters and can produce approximation with default settings or derive such in an automated manner by heuristics.

As noted by Aigner et al. [AMM*08a], the user's ability to tune parameters is an important aspect of the usability of a system or method. At the same time, it is difficult to evaluate how easy or difficult is each approach for different groups of users in different contexts.

7.3. Temporal Characteristics

Aigner et al. [AMM*08a] emphasize that time should be respected as an independent dimension when visualizing temporal data; however, the authors admit that even those methods that do not respect time as an independent dimension can be applicable to time series data. An example is Principal Component Analysis [Jol02], which has been successfully applied to time series without extracting time as a separate dimension [BDA11]. Hence, even though time should be dealt with in a special way, analytical methods that do not do this may still be capable of producing meaningful results. Consequently, this work could be extended by considering abstraction techniques from the domains of data mining other than time series mining.

8. Conclusion

We have described the challenges of large time series visualization and substantiated the need for abstraction. We have discussed the requirements that need to be considered to find a suitable abstraction method for time series visualization.

Our contribution is two-fold. First, we propose a framework for the classification of time series abstraction algorithms that specifies and justifies the essential criteria for informed method selection. Second, we provide a classification of a large number of existing abstraction methods, which can be practically used for method selection. It is challenging to make an exhaustive survey of all time series abstraction methods. However, we believe that the framework we presented, along with the proposed directions for its extension (Section 7), provides a good opportunity for obtaining a unified picture of data abstraction for large time series and will prove useful for visualization designers, who would be able to either use our model as is or customize/extend it based on their vision.

We believe that this work can facilitate the pursuit of a

closer integration of computational and visual methods in visual analytics systems.

References

[AA05] ANDRIENKO N., ANDRIENKO G.: *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. 2, 10

[AA17] ANDRIENKO N., ANDRIENKO G.: State transition graphs for semantic analysis of movement behaviours. *Information Visualization ?*, ? (2017), 1473871617692841. doi: 10.1177/1473871617692841. 2, 9

[AAB*10] ANDRIENKO G., ANDRIENKO N., BREMM S., SCHRECK T., LANDESBERGER T. v., BAK P., KEIM D.: Space-in-Time and Time-in-Space Self-Organizing Maps for Exploring Spatiotemporal Patterns. *Computer Graphics Forum* (2010). 2, 9

[AAF16] ANDRIENKO G., ANDRIENKO N., FUCHS G.: Understanding movement data quality. *Journal of Location Based Services 10*, 1 (2016), 31–46. URL: http://dx.doi.org/10.1080/17489725.2016.1169322, doi:10.1080/17489725.2016.1169322. 8

[AAM*10] ANDRIENKO G., ANDRIENKO N., MLADENOV M., MOCK M., POELITZ C.: Extracting events from spatial time series. In *2010 14th International Conference Information Visualisation* (2010), IEEE, pp. 48–53. 10

[ACD*06] ADLASSNIG K.-P., COMBI C., DAS A. K., KERAVNOU E. T., POZZI G.: Temporal representation and reasoning in medicine: Research directions and challenges. *Artificial Intelligence in Medicine 38*, 2 (2006), 101 – 113. Temporal Representation and Reasoning in Medicine. URL: http://www.sciencedirect.com/science/article/pii/S0933365706001412. 12

[Ado95] ADORF H.-M.: Interpolation of Irregularly Sampled Data Series—A Survey. In *Astronomical Data Analysis Software and Systems IV* (1995), Shaw R. A., Payne H. E., Hayes J. J. E., (Eds.), vol. 77 of *Astronomical Society of the Pacific Conference Series*, p. 460. 8

[AES05] AMAR R., EAGAN J., STASKO J.: Low-level components of analytic activity in information visualization. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization* (Washington, DC, USA, 2005), INFOVIS '05, IEEE Computer Society, pp. 15– . URL: http://dx.doi.org/10.1109/INFOVIS.2005.24, doi:10.1109/INFOVIS.2005.24. 9

[AFS93] AGRAWAL R., FALOUTSOS C., SWAMI A. N.: Efficient similarity search in sequence databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms* (London, UK, UK, 1993), FODO '93, Springer-Verlag, pp. 69–84. URL: http://dl.acm.org/citation.cfm?id=645415.652239. 12, 14, 15, 16

[AKMM11] AIGNER W., KAINZ C., MA R. K. K., MIKSCH S.: Bertin was right: An empirical evaluation of indexing to compare multivariate time-series data using line plots. *Comput. Graph. Forum 30*, 1 (2011), 215–228. 11

[AMM*07] AIGNER W., MIKSCH S., MÜLLER W., SCHUMANN H., TOMINSKI C.: Visualizing time-oriented data-a systematic view. *Comput. Graph. 31*, 3 (June 2007), 401–409. URL: http://dx.doi.org/10.1016/j.cag.2007.01.030, doi:10.1016/j.cag.2007.01.030. 1, 4, 5, 7

[AMM*08a] AIGNER W., MIKSCH S., MULLER W., SCHUMANN H., TOMINSKI C.: Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 47–60. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2007.70415. 2, 3, 16

[AMM*08b] AIGNER W., MIKSCH S., MULLER W., SCHUMANN H., TOMINSKI C.: Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 47–60. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2007.70415. 2, 4, 5, 7, 10

[AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of Time-Oriented Data*, 1st ed. Springer Publishing Company, Incorporated, 2011. 2, 5, 7, 9

[ASP*05] ARIS A., SHNEIDERMAN B., PLAISANT C., SHMUELI G., JANK W.: Representing unevenly-spaced time series data for visualization and interactive exploration. In *IFIP Conference on Human-Computer Interaction* (2005), Springer, pp. 835–846. 7, 8

[BBG*09] BLAAS J., BOTHA C., GRUNDY E., JONES M., LARAMEE R., POST F.: Smooth graphs for visual exploration of higher-order state transitions. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 969–976. 9

[BDA11] BANKÓ Z., DOBOS L., ABONYI J.: Dynamic principal component analysis in multivariate time-series segmentation. *Conservation, Information, Evolution-towards a sustainable engineering and economy 1*, 1 (2011), 11–24. 16

[BDF*15] BERNARD J., DABERKOW D., FELLNER D., FISCHER K., KOEPLER O., KOHLHAMMER J., RUNNWERTH M., RUPPERT T., SCHRECK T., SENS I.: Visinfo: a digital library system for time series research data based on exploratory search—a user-centered design approach. *International Journal on Digital Libraries 16*, 1 (2015), 37–59. URL: http://dx.doi.org/10.1007/s00799-014-0134-y, doi:10.1007/s00799-014-0134-y. 5

[BEMR05] BEYGELZIMER A., ERDOGAN E., MA S., RISH I.: Statictical models for unequally spaced time series. In *Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005, Newport Beach, CA, USA, April 21-23, 2005* (2005), pp. 626–630. URL: http://dx.doi.org/10.1137/1.9781611972757.74, doi:10.1137/1.9781611972757.74. 8

[Ber83] BERTIN J.: *Semiology of Graphics*. University of Wisconsin Press, 1983. 9, 11

[Ber15] BERNARD J.: *Exploratory search in time-oriented primary data*. PhD thesis, Technische Universität, Darmstadt, December 2015. URL: http://tuprints.ulb.tu-darmstadt.de/5173/. 6, 8

[BGH*06] BINGHAM E., GIONIS A., HAIMINEN N., HIISILÄ H., MANNILA H., TERZI E.: Segmentation and dimensionality reduction. In *Proceedings of the Sixth SIAM International Conference on Data Mining, April 20-22, 2006, Bethesda, MD, USA* (2006), pp. 372–383. URL: http://dx.doi.org/10.1137/1.9781611972764.33, doi:10.1137/1.9781611972764.33. 14, 15

[BL09] BERTINI E., LALANNE D.: Surveying the complementary role of automatic data analysis and visualization in knowledge discovery. In *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration* (2009), ACM, pp. 12–20. 2

[BL10] BERTINI E., LALANNE D.: Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *ACM SIGKDD Explorations Newsletter 11*, 2 (2010), 9–18. 2

[BM13] BREHMER M., MUNZNER T.: A multi-level typology of abstract visualization tasks. *IEEE Trans. Visualization and Computer Graphics (TVCG) (Proc. InfoVis) 19*, 12 (2013), 2376–2385. 9

[BRG*12] BERNARD J., RUPPERT T., GOROLL O., MAY T., KOHLHAMMER J.: Visual-interactive preprocessing of time series data. In *Proceedings of SIGRAD 2012; Interactive Visual Analysis of Data; November 29-30; 2012; Växjö; Sweden* (2012), no. 081, Linköping University Electronic Press, pp. 39–48. 6, 8

[BSBH09] BATAL I., SACCHI L., BELLAZZI R., HAUSKRECHT M.: Multivariate time series classification with temporal abstractions. 15

[BSH94] BEDERSON B. B., STEAD L., HOLLAN J. D.: Pad++: Advances in multiscale interfaces. In *Conference Companion on Human Factors in Computing Systems* (New York, NY, USA, 1994), CHI '94, ACM, pp. 315–316. URL: http://doi.acm.org/10.1145/259963.260379, doi:10.1145/259963.260379. 3

[BSH*16a] BACH B., SHI C., HEULOT N., MADHYASTHA T., GRABOWSKI T., DRAGICEVIC P.: Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan 2016), 559–568. 2

[BSH*16b] BACH B., SHI C., HEULOT N., MADHYASTHA T., GRABOWSKI T., DRAGICEVIC P.: Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE transactions on visualization and computer graphics 22*, 1 (2016), 559–568. 3

[BTK11] BERTINI E., TATU A., KEIM D.: Quality metrics in high-dimensional data visualization: An overview and systematization. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2203–2212. 16

[BW08] BYRON L., WATTENBERG M.: Stacked graphs – geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov. 2008), 1245–1252. URL: http://dx.doi.org/10.1109/TVCG.2008.166, doi:10.1109/TVCG.2008.166. 3

[CF99] CHAN K.-P., FU A.-C.: Efficient time series matching by wavelets. In *Data Engineering, 1999. Proceedings., 15th International Conference on* (Mar 1999), pp. 126–133. doi:10.1109/ICDE.1999.754915. 12, 13, 14, 15

[cFlCCm06] CHUNG FU T., LAI CHUNG F., CHAK-MAN N.: Financial Time Series Segmentation based on Specialized Binary Tree Representation. In *Int. Conf. on Data Mining* (2006), pp. 3–9. 3, 14, 15

[Chi00] CHI E. H.: A taxonomy of visualization techniques using the data state reference model. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on* (2000), pp. 69–75. doi:10.1109/INFVIS.2000.885092. 3, 4, 5

[CKMP02] CHAKRABARTI K., KEOGH E., MEHROTRA S., PAZZANI M.: Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Trans. Database Syst. 27*, 2 (June 2002), 188–228. URL: http://doi.acm.org/10.1145/568518.568520, doi:10.1145/568518.568520. 12, 13, 14, 15

[CKPS10] COMBI C., KERAVNOU-PAPAILIOU E., SHAHAR Y.: Temporal information systems in medicine, 2010. 5

[CMS99] CARD S. K., MACKINLAY J. D., SHNEIDERMAN B. (Eds.): *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. 3, 4, 5

[CRC03] CEGLAR A., RODDICK J. F., CALDER P.: Managing data mining technologies in organizations. IGI Global, Hershey, PA, USA, 2003, ch. Guiding Knowledge Discovery Through Interactive Data Mining, pp. 45–87. URL: http://dl.acm.org/citation.cfm?id=778595.778606. 1

[CWRY06] CUI Q., WARD M., RUNDENSTEINER E., YANG J.: Measuring data abstraction quality in multiresolution visualizations. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5 (Sept 2006), 709–716. doi:10.1109/TVCG.2006.161. 3, 16

[DB16] DAGUM E., BIANCONCINI S.: *Seasonal Adjustment Methods and Real Time Trend-Cycle Estimation*. Statistics for Social and Behavioral Sciences. Springer International Publishing, 2016. URL: https://books.google.de/books?id=K-t6DAAAQBAJ. 8, 10

[DE02] DIX A., ELLIS G.: By chance enhancing interaction with large data sets through statistical sampling. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2002), AVI '02, ACM, pp. 167–176. URL: http://doi.acm.org/10.1145/1556262.1556289, doi:10.1145/1556262.1556289. 3

[DGM97] DAS G., GUNOPULOS D., MANNILA H.: Finding similar time series. In *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery* (London, UK, UK, 1997), PKDD '97, Springer-Verlag, pp. 88–100. URL: http://dl.acm.org/citation.cfm?id=645801.669017. 3

[DGM*01] DACOROGNA M., GENÇAY R., MULLER U., PICTET O., OLSEN R.: *An Introduction to High-Frequency Finance*. Elsevier Science, 2001. URL: https://books.google.de/books?id=dobO95EBcqsC. 8

[DSDH13] DAN J., SHI W., DONG F., HIROTA K.: Piecewise trend approximation: A ratio-based time series representation. *Abstr. Appl. Anal. 2013, Special Issue* (2013), 1–7. URL: http://dx.doi.org/10.1155/2013/603629, doi:10.1155/2013/603629. 14, 15

[DSP*17] DU F., SHNEIDERMAN B., PLAISANT C., MALIK S., PERER A.: Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus. *IEEE Transactions on Visualization and Computer Graphics 23*, 6 (June 2017), 1636–1649. doi:10.1109/TVCG.2016.2539960. 1, 2, 4, 9

[DTS*08] DING H., TRAJCEVSKI G., SCHEUERMANN P., WANG X., KEOGH E.: Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment 1*, 2 (2008), 1542–1552. 6

[EA12] ESLING P., AGON C.: Time-series data mining. *ACM Comput. Surv. 45*, 1 (Dec. 2012), 12:1–12:34. URL: http://doi.acm.org/10.1145/2379776.2379788, doi:10.1145/2379776.2379788. 3, 6, 9, 10

[EAE05] ELFEKY M. G., AREF W. G., ELMAGARMID A. K.: Periodicity detection in time series databases. *IEEE Transactions on Knowledge and Data Engineering 17*, 7 (2005), 875–887. 12

[Eck14] ECKNER A.: A framework for the analysis of unevenly spaced time series data. *Preprint. Available at: http://eckner.com/papers/unevenly_spaced_time_series_analysis.pdf* (July 2014). 8

[Eck17] ECKNER A.: Algorithms for unevenly spaced time series: Moving averages and other rolling operators. *Preprint. Available at: http://eckner. com/papers/Algorithms for Unevenly Spaced Time Series.pdf* (July 2017). 8

[ED07] ELLIS G., DIX A.: A taxonomy of clutter reduction for

information visualisation. *Visualization and Computer Graphics, IEEE Transactions on 13*, 6 (Nov 2007), 1216–1223. doi:10.1109/TVCG.2007.70535. 4

[EF10] ELMQVIST N., FEKETE J.-D.: Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics 16*, 3 (2010), 439–454. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2009.84. 3, 4

[EHD*02] EADS D. R., HILL D., DAVIS S., PERKINS S. J., MA J., PORTER R. B., THEILER J. P.: Genetic algorithms and support vector machines for time series classification, 2002. URL: http://dx.doi.org/10.1117/12.453526, doi:10.1117/12.453526. 14, 15

[FDA02] FITZGIBBON L., DOWE D., ALLISON L.: Change-point estimation using new minimum message length approximations. In *PRICAI 2002: Trends in Artificial Intelligence*, Ishizuka M., Sattar A., (Eds.), vol. 2417 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2002, pp. 244–254. URL: http://dx.doi.org/10.1007/3-540-45683-X_28, doi:10.1007/3-540-45683-X_28. 14, 15

[Fek13] FEKETE J.: Visual analytics infrastructures: From data management to exploration. *IEEE Computer 46*, 7 (2013), 22–29. URL: http://dx.doi.org/10.1109/MC.2013.120, doi:10.1109/MC.2013.120. 1

[FGNS10] FUCHS E., GRUBER T., NITSCHKE J., SICK B.: Online segmentation of time series based on polynomial least-squares approximations. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 12 (Dec. 2010), 2232–2245. URL: http://dx.doi.org/10.1109/TPAMI.2010.44, doi:10.1109/TPAMI.2010.44. 14, 15, 16

[FM13] FUAD M. M. M., MARTEAU P.: Towards a faster symbolic aggregate approximation method. *CoRR abs/1301.5871* (2013). URL: http://arxiv.org/abs/1301.5871. 14, 15

[FPSS96] FAYYAD U., PIATETSKY-SHAPIRO G., SMYTH P.: From data mining to knowledge discovery in databases. *AI magazine 17*, 3 (1996), 37. 10

[Fra98] FRANK A. U.: Different types of" times" in gls. *Spatial and temporal reasoning in geographic information systems* (1998), 40. 4

[FRM94] FALOUTSOS C., RANGANATHAN M., MANOLOPOULOS Y.: Fast subsequence matching in time-series databases. *SIGMOD Rec. 23*, 2 (May 1994), 419–429. URL: http://doi.acm.org/10.1145/191843.191925, doi:10.1145/191843.191925. 3

[Fu11] FU T.-C.: A review on time series data mining. *Eng. Appl. Artif. Intell. 24*, 1 (Feb. 2011), 164–181. URL: http://dx.doi.org/10.1016/j.engappai.2010.09.007, doi:10.1016/j.engappai.2010.09.007. 1, 6, 9

[GGAM12] GSCHWANDTNER T., GÄRTNER J., AIGNER W., MIKSCH S.: A taxonomy of dirty time-oriented data. In *International Conference on Availability, Reliability, and Security* (2012), Springer, pp. 58–72. 8

[GSBO13] GHONIEM M., SHURKHOVETSKYY G., BAHEY A., OTJACQUES B.: Vafle: visual analytics of firewall log events, 2013. URL: http://dx.doi.org/10.1117/12.2037790, doi:10.1117/12.2037790. 11, 14, 15

[HB15] HATWAR K., BADHIYE S.: Alphabetic time series representation using trend based approach. In *Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on* (March 2015), pp. 1–4. doi:10.1109/ICIIECS.2015.7193171. 14, 15

[HBO10] HEER J., BOSTOCK M., OGIEVETSKY V.: A tour through the visualization zoo. *Commun. ACM 53*, 6 (June 2010), 59–67. URL: http://doi.acm.org/10.1145/1743546.1743567, doi:10.1145/1743546.1743567. 9, 11

[HDY99] HAN J., DONG G., YIN Y.: Efficient mining of partial periodic patterns in time series database. In *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)* (Mar 1999), pp. 106–115. doi:10.1109/ICDE.1999.754913. 12

[HHWN02] HAVRE S., HETZLER E., WHITNEY P., NOWELL L.: Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (Jan. 2002), 9–20. URL: http://dx.doi.org/10.1109/2945.981848, doi:10.1109/2945.981848. 3

[HKA09] HEER J., KONG N., AGRAWALA M.: Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), ACM, pp. 1303–1312. 3

[HKM*01] HIMBERG J., KORPIAHO K., MANNILA H., TIKANMAKI J., TOIVONEN H.: Time series segmentation for context recognition in mobile devices. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on* (2001), pp. 203–210. doi:10.1109/ICDM.2001.989520. 14, 15

[HM99] HUNTER J., MCINTOSH N.: Knowledge-based event detection in complex time series data. In *Artificial Intelligence in Medicine*, Horn W., Shahar Y., Lindberg G., Andreassen S., Wyatt J., (Eds.), vol. 1620 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1999, pp. 271–280. URL: http://dx.doi.org/10.1007/3-540-48720-4_30, doi:10.1007/3-540-48720-4_30. 14, 15

[Höp02] HÖPPNER F.: Time series abstraction methods-a survey. In *GI Jahrestagung* (2002), pp. 777–786. 6

[HSKIH07] HAGH-SHENAS H., KIM S., INTERRANTE V., HEALEY C.: Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. *Visualization and Computer Graphics, IEEE Transactions on 13*, 6 (Nov 2007), 1270–1277. doi:10.1109/TVCG.2007.70623. 6

[HSW07] HERZOG T. N., SCHEUREN F. J., WINKLER W. E.: *What is Data Quality and Why Should We Care?* Springer New York, New York, NY, 2007, pp. 7–15. URL: http://dx.doi.org/10.1007/0-387-69505-2_2, doi:10.1007/0-387-69505-2_2. 8

[Jol02] JOLLIFFE I.: *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002. 6, 16

[JZW07] JIANG J., ZHANG Z., WANG H.: A new segmentation algorithm to stock time series based on pip approach. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on* (Sept 2007), pp. 5609–5612. doi:10.1109/WICOM.2007.1374. 14, 15

[KCHP01] KEOGH E., CHU S., HART D., PAZZANI M.: An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on* (2001), pp. 289–296. doi:10.1109/ICDM.2001.989531. 14, 15

[KCPM01] KEOGH E., CHAKRABARTI K., PAZZANI M., MEHROTRA S.: Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems 3*, 3 (2001), 263–286. URL: http://dx.doi.org/10.1007/PL00011669, doi:10.1007/PL00011669. 12, 13

[Kei02] KEIM D. A.: Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (Jan. 2002), 1–8. URL: http://dx.doi.org/10.1109/2945.981847, doi:10.1109/2945.981847. 1, 3

[KGP01] KALPAKIS K., GADA D., PUTTAGUNTA V.: Distance measures for effective clustering of arima time-series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on* (2001), pp. 273–280. doi:10.1109/ICDM.2001.989529. 14, 15

[KGZ*12] KUHNE L., GIESEN J., ZHANG Z., HA S., MUELLER K.: A data-driven approach to hue-preserving color-blending. *Visualization and Computer Graphics, IEEE Transactions on 18*, 12 (Dec 2012), 2122–2129. doi:10.1109/TVCG.2012.186. 6

[KHP*11] KANDEL S., HEER J., PLAISANT C., KENNEDY J., VAN HAM F., RICHE N. H., WEAVER C., LEE B., BRODBECK D., BUONO P.: Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization 10*, 4 (Oct. 2011), 271–288. URL: http://dx.doi.org/10.1177/1473871611415994, doi:10.1177/1473871611415994. 6, 8

[KJF97] KORN F., JAGADISH H. V., FALOUTSOS C.: Efficiently supporting ad hoc queries in large datasets of time sequences. *SIGMOD Rec. 26*, 2 (June 1997), 289–300. URL: http://doi.acm.org/10.1145/253262.253332, doi:10.1145/253262.253332. 12, 14, 15

[KK03] KEOGH E., KASETTY S.: On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery 7*, 4 (2003), 349–371. 11

[Kle15] KLEIST C.: Time series data mining methods - a review, 2015. URL: http://edoc.hu-berlin.de/docviews/abstract.php?id=41733. 10

[KLS00] KREUSELER M., LOPEZ N., SCHUMANN H.: A scalable framework for information visualization. In *Proceedings of the IEEE Symposium on Information Vizualization 2000* (Washington, DC, USA, 2000), INFOVIS '00, IEEE Computer Society, pp. 27–. URL: http://dl.acm.org/citation.cfm?id=857190.857688. 3

[KMS*08] KEIM D. A., MANSMANN F., SCHNEIDEWIND J., THOMAS J., ZIEGLER H.: Visual data mining. Springer-Verlag, Berlin, Heidelberg, 2008, ch. Visual Analytics: Scope and Challenges, pp. 76–90. URL: http://dx.doi.org/10.1007/978-3-540-71080-6_6, doi:10.1007/978-3-540-71080-6_6. 2, 3

[KP98] KEOGH E. J., PAZZANI M. J.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, August 27-31, 1998* (1998), pp. 239–243. URL: http://www.aaai.org/Library/KDD/1998/kdd98-041.php. 14, 15

[KP00] KEOGH E., PAZZANI M.: A simple dimensionality reduction technique for fast similarity search in large time series databases. In *Knowledge Discovery and Data Mining. Current Issues and New Applications*, Terano T., Liu H., Chen A., (Eds.), vol. 1805 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2000, pp. 122–133. URL: http://dx.doi.org/10.1007/3-540-45571-X_14, doi:10.1007/3-540-45571-X_14. 13, 14, 15, 16

[LKL05] LIN J., KEOGH E., LONARDI S.: Visualizing and discovering non-trivial patterns in large time series databases. *Information Visualization 4*, 2 (July 2005), 61–82. URL: http:

//dx.doi.org/10.1057/palgrave.ivs.9500089, doi:10.1057/palgrave.ivs.9500089. 3

[LKLC03] LIN J., KEOGH E., LONARDI S., CHIU B.: A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (New York, NY, USA, 2003), DMKD '03, ACM, pp. 2–11. URL: http://doi.acm.org/10.1145/882082.882086, doi:10.1145/882082.882086. 6, 7, 9, 12, 14, 15

[LMW*15] LIU S., MALJOVEC D., WANG B., BREMER P.-T., PASCUCCI V.: Visualizing high-dimensional data: Advances in the past decade. In *Proc. Eurographics Conf. Visualization* (2015), pp. 20151115–127. 5

[LP11] LI L., PRAKASH B. A.: Time series clustering: Complex is simpler! In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (2011), pp. 185–192. 16

[LRKC10] LEE B., RICHE N. H., KARLSON A. K., CARPENDALE S.: Sparkclouds: Visualizing trends in tag clouds. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (Nov. 2010), 1182–1189. 9

[LSK06] LKHAGVA B., SUZUKI Y., KAWAGOE K.: New time series data representation ESAX for financial applications. In *Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDE 2006, 3-7 April 2006, Atlanta, GA, USA* (2006), p. 115. URL: http://dx.doi.org/10.1109/ICDEW.2006.99, doi:10.1109/ICDEW.2006.99. 14, 15

[LSL*00] LAVRENKO V., SCHMILL M., LAWRIE D., OGILVIE P., JENSEN D., ALLAN J.: Mining of concurrent text and time series. In *IN PROCEEDINGS OF THE 6 TH ACM SIGKDD INT'L CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING WORKSHOP ON TEXT MINING* (2000), pp. 37–44. 14, 15

[MA14] MIKSCH S., AIGNER W.: A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data. *Computers & Graphics 38* (2014), 286–290. 5, 6, 7, 8, 10

[MMKN08] MCLACHLAN P., MUNZNER T., KOUTSOFIOS E., NORTH S.: Liverac: Interactive visual exploration of system management time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), CHI '08, ACM, pp. 1483–1492. URL: http://doi.acm.org/10.1145/1357054.1357286, doi:10.1145/1357054.1357286. 5

[Moe06] MOERCHEN F.: *Time series knowledge mining*. Görich & Weiershäuser, 2006. 9

[MS09] MOSKOVITCH R., SHAHAR Y.: Medical temporal-knowledge discovery via temporal abstraction. *AMIA annual symposium proceedings 2009* (2009), 452. 6

[Mül91] MÜLLER U. A.: Specially weighted moving averages with repeated application of the EMA operator. *Technical report UAM* (1991), 10–14. 8

[MWLF05] MEGALOOIKONOMOU V., WANG Q., LI G., FALOUTSOS C.: A multiresolution symbolic representation of time series. In *Proceedings of the 21st International Conference on Data Engineering* (Washington, DC, USA, 2005), ICDE '05, IEEE Computer Society, pp. 668–679. URL: http://dx.doi.org/10.1109/ICDE.2005.10, doi:10.1109/ICDE.2005.10. 14, 15

[NJW02] NG A. Y., JORDAN M. I., WEISS Y.: On spectral clustering: Analysis and an algorithm. In *Advances in Neural Infor-*

mation Processing Systems 14, Dieterich T., Becker S., Ghahramani Z., (Eds.). MIT Press, 2002, pp. 849–856. 11

[PKC01] PARK S., KIM S.-W., CHU W. W.: Segment-based approach for subsequence searches in sequence databases. In *Proceedings of the 2001 ACM Symposium on Applied Computing* (New York, NY, USA, 2001), SAC '01, ACM, pp. 248–252. URL: http://doi.acm.org/10.1145/372202.372334, doi:10.1145/372202.372334. 14, 15

[RAA11] RASHEED F., ALSHALALFA M., ALHAJJ R.: Efficient periodicity mining in time series databases using suffix trees. *IEEE Transactions on Knowledge and Data Engineering 23*, 1 (Jan 2011), 79–94. doi:10.1109/TKDE.2010.76. 12

[RS02] RODDICK J. F., SPILIOPOULOU M.: A survey of temporal knowledge discovery paradigms and methods. *IEEE Trans. on Knowl. and Data Eng. 14*, 4 (July 2002), 750–767. URL: http://dx.doi.org/10.1109/TKDE.2002.1019212, doi:10.1109/TKDE.2002.1019212. 6

[SBvLK09] SCHRECK T., BERNARD J., VON LANDESBERGER T., KOHLHAMMER J.: Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization 8*, 1 (2009), 14–29. URL: http://dx.doi.org/10.1057/ivs.2008.29, doi:10.1057/ivs.2008.29. 9

[SGBBT06] SHAHAR Y., GOREN-BAR D., BOAZ D., TAHAN G.: Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions. *Artificial Intelligence in Medicine 38*, 2 (2006), 115 – 135. Temporal Representation and Reasoning in Medicine. URL: http://www.sciencedirect.com/science/article/pii/S0933365705000771, doi:http://dx.doi.org/10.1016/j.artmed.2005.03.001. 6

[Sha97a] SHAHAR Y.: A framework for knowledge-based temporal abstraction. *Artificial Intelligence 90*, 1–2 (1997), 79 – 133. URL: http://www.sciencedirect.com/science/article/pii/S0004370296000252, doi:http://dx.doi.org/10.1016/S0004-3702(96)00025-2. 3, 12, 14, 15

[Sha97b] SHAHAR Y.: A framework for knowledge-based temporal abstraction. *Artificial Intelligence 90*, 1–2 (1997), 79 – 133. URL: http://www.sciencedirect.com/science/article/pii/S0004370296000252, doi:http://dx.doi.org/10.1016/S0004-3702(96)00025-2. 14, 15

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages* (Washington, DC, USA, 1996), VL '96, IEEE Computer Society, pp. 336–. URL: http://dl.acm.org/citation.cfm?id=832277.834354. 2, 9

[SK13] SCHULTZ T., KINDLMANN G.: Open-box spectral clustering: Applications to medical image analysis. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (Dec. 2013). Accepted for publication. 1

[SLW97] STRONG D. M., LEE Y. W., WANG R. Y.: Data quality in context. *Communications of the ACM 40*, 5 (1997), 103–110. 8

[SM96] SHAHAR Y., MUSEN M. A.: Knowledge-based temporal abstraction in clinical domains. *Artificial intelligence in medicine 8*, 3 (1996), 267–298. 12

[SM07] STACEY M., MCGREGOR C.: Temporal abstraction in intelligent clinical data analysis: A survey. *Artificial Intelligence in Medicine 39*, 1 (2007), 1 – 24. URL: http://www.sciencedirect.com/science/article/pii/S0933365706001345, doi:http://dx.doi.org/10.1016/j.artmed.2006.08.002. 6

[Smy97] SMYTH P.: Clustering sequences with hidden markov models. In *Advances in Neural Information Processing Systems* (1997), MIT Press, pp. 648–654. 12, 14, 15

[Sta09] STACEY M. R.: *A Framework for Multi-dimensional Online Temporal Abstraction*. University of Western Sydney, 2009. 12

[Tom11] TOMINSKI C.: Event-based concepts for user-driven visualization. *Information Visualization 10*, 1 (2011), 65–81. URL: http://dx.doi.org/10.1057/ivs.2009.32, doi:10.1057/ivs.2009.32. 5, 10

[Tor52] TORGERSON W.: Multidimensional scaling: I. theory and method. *Psychometrika 17*, 4 (1952), 401–419. URL: http://dx.doi.org/10.1007/BF02288916, doi:10.1007/BF02288916. 3

[Tuf86] TUFTE E. R.: *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986. 2

[vdEHBvW16] VAN DEN ELZEN S., HOLTEN D., BLAAS J., VAN WIJK J. J.: Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE transactions on visualization and computer graphics 22*, 1 (2016), 1–10. 2

[vLBR*16] VON LANDESBERGER T., BRODKORB F., ROSKOSCH P., ANDRIENKO N., ANDRIENKO G., KERREN A.: Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan 2016), 11–20. 9

[VSP*07] VERDUIJN M., SACCHI L., PEEK N., BELLAZZI R., DE JONGE E., DE MOL B. A.: Temporal abstraction for feature extraction: A comparative case study in prediction from intensive care monitoring data. *Artificial Intelligence in Medicine 41*, 1 (2007), 1 – 12. URL: http://www.sciencedirect.com/science/article/pii/S093336570700070X, doi:http://dx.doi.org/10.1016/j.artmed.2007.06.003. 6, 12, 14, 15

[VWVS99] VAN WIJK J. J., VAN SELOW E. R.: Cluster and calendar based visualization of time series data. In *Proceedings of the 1999 IEEE Symposium on Information Visualization* (Washington, DC, USA, 1999), INFOVIS '99, IEEE Computer Society, pp. 4–. URL: http://dl.acm.org/citation.cfm?id=857189.857665. 3, 9

[WAM01] WEBER M., ALEXA M., MÜLLER W.: Visualizing time-series on spirals. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)* (Washington, DC, USA, 2001), INFOVIS '01, IEEE Computer Society, pp. 7–. URL: http://dl.acm.org/citation.cfm?id=580582.857719. 3

[WL05] WARREN LIAO T.: Clustering of time series data-a survey. *Pattern Recogn. 38*, 11 (Nov. 2005), 1857–1874. URL: http://dx.doi.org/10.1016/j.patcog.2005.01.025, doi:10.1016/j.patcog.2005.01.025. 6

[WMD*13] WANG X., MUEEN A., DING H., TRAJCEVSKI G., SCHEUERMANN P., KEOGH E.: Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery 26*, 2 (2013), 275–309. 6, 12, 13

[WSH05] WANG X., SMITH K., HYNDMAN R.: Dimension reduction for clustering time series using global characteristics. In *Computational Science – ICCS 2005*, Sunderam V., van Albada G., Sloot P., Dongarra J., (Eds.), vol. 3516 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 792–795. URL: http://dx.doi.org/10.1007/11428862_108, doi:10.1007/11428862_108. 14, 15

[XZKP12] XU Z., ZHANG R., KOTAGIRI R., PARAMPALLI U.: An adaptive algorithm for online time series segmentation with error bound guarantee. In *Proceedings of the 15th International Conference on Extending Database Technology* (New York, NY, USA, 2012), EDBT '12, ACM, pp. 192–203. URL: http://doi.acm.org/10.1145/2247596.2247620, doi:10.1145/2247596.2247620. 14, 15

[YS04] YANG K., SHAHABI C.: A pca-based similarity measure for multivariate time series. In *Proceedings of the Second ACM International Workshop on Multimedia Databases, ACM-MMDB 2004, Washington, DC, USA, November 13, 2004* (2004), pp. 65–74. URL: http://doi.acm.org/10.1145/1032604.1032616, doi:10.1145/1032604.1032616. 11

[YS05] YANG K., SHAHABI C.: A multilevel distance-based index structure for multivariate time series. In *12th International Symposium on Temporal Representation and Reasoning (TIME 2005), 23-25 June 2005, Burlington, Vermont, USA* (2005), pp. 65–73. URL: http://dx.doi.org/10.1109/TIME.2005.8, doi:10.1109/TIME.2005.8. 11

[YS07] YANG K., SHAHABI C.: An efficient k nearest neighbor search for multivariate time series. *Information and Computation 205*, 1 (2007), 65 – 98. Special Issue: {TIME} 2005. URL: http://www.sciencedirect.com/science/article/pii/S0890540106001039, doi:http://dx.doi.org/10.1016/j.ic.2006.08.004. 12

[YYS05] YOON H., YANG K., SHAHABI C.: Feature subset selection and feature ranking for multivariate time series. *IEEE Trans. Knowl. Data Eng. 17*, 9 (2005), 1186–1198. URL: http://dx.doi.org/10.1109/TKDE.2005.144, doi:10.1109/TKDE.2005.144. 3, 11, 14, 15

[ZWL07] ZHU Y., WU D., LI S.: A piecewise linear representation method of time series based on feature points. In *Knowledge-Based Intelligent Information and Engineering Systems*, Apolloni B., Howlett R., Jain L., (Eds.), vol. 4693 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, pp. 1066–1072. URL: http://dx.doi.org/10.1007/978-3-540-74827-4_133, doi:10.1007/978-3-540-74827-4_133. 14, 15