SIGRAD 2012
A. Kerren and S. Seipel (Editors)

Visual-Interactive Preprocessing of Time Series Data

J. Bernard[1], T. Ruppert[1], O. Goroll[2], T. May[1], and J. Kohlhammer[1]

[1]Fraunhofer IGD Darmstadt, Germany
[2]Technische Universitität Darmstadt, Germany

Abstract

Time series data is an important data type in many different application scenarios. Consequently, there are a great variety of approaches for analyzing time series data. Within these approaches different strategies for cleaning, segmenting, representing, normalizing, comparing, and aggregating time series data can be found. When combining these operations, the time series analysis preprocessing workflow has many degrees of freedom. To define an appropriate preprocessing pipeline, the knowledge of experts coming from the application domain has to be included into the design process. Unfortunately, these experts often cannot estimate the effects of the chosen preprocessing algorithms and their parameterizations on the time series. We introduce a system for the visual-interactive exploitation of the preprocessing parameter space. In contrast to 'black box'-driven approaches designed by computer scientists based on the requirements of domain experts, our system allows these experts to visual-interactively compose time series preprocessing pipelines by themselves. Visual support is provided to choose the right order and parameterization of the preprocessing steps. We demonstrate the usability of our approach with a case study from the digital library domain, in which time-oriented scientific research data has to be preprocessed to realize a visual search and analysis application.

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

1. Introduction

Large time series repositories are built up in many scientific disciplines like climate research, genomics research or high-energy physics. Many of these repositories are commonly shared by researchers to search for new findings. In virtually all cases raw time series have to be prepared for effective retrieval as well as for effective exploratory analysis. Data preprocessing is necessary whenever the data does not match the requirements of the following analytical tasks and methods. For example, an algorithm for time series clustering may require uniformly sampled data; another analysis algorithm may not be able to deal with missing values etc. Typically, preprocessing is a combination of different operations arranged in a pipeline.

The outset of our work is that a user is given an analytical task and a large repository of time series data. The preprocessing task is shared by two user roles: The data mining expert is responsible to choose and modify the operations for the pipeline. The domain expert is responsible to define the criteria for useful data. Both experts are responsible to identify and communicate on errors in their respective domain. We assume that the visualization of the process and the data can be the medium for this communication. As soon as the pipeline is set up and tested, the preprocessing is applied as an automated process. If it is not known whether the data matches the requirements of the analysis, it certainly is not advisable to apply an automated preprocess as a 'black-box' operation to see what happens. An interactive adjustment of the pipeline can be used to learn about the characteristic properties of the data, to check the requirements and to test the effects of suitable operations and their parameters.

Kandel et al. [KHP*11] coined the term 'data wrangling' to describe scencarios of this type. They state that the process has to be visible and audible to domain experts to identify and correct potential errors or invalid assumptions. (Domain experts needed to search errors). The coupled interactive visualization of data and process - including all data-fixing efforts - is required to enable domain experts to control the process. Following their recommendation, our work

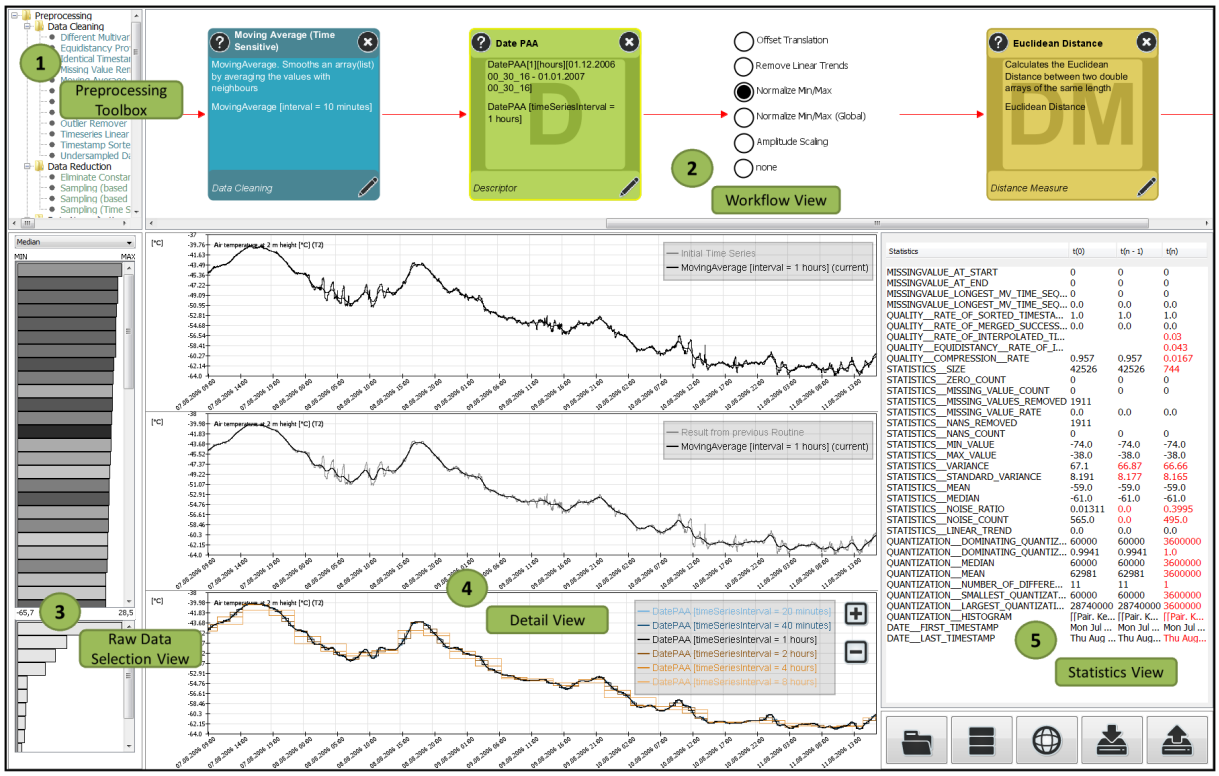

Figure 1: *Preprocessing pipeline creator: a visual-interactive time series preprocessing system. Effects of the Piecewise Approximate Aggregation (PAA) descriptor [KCPM00] are shown in the detail view (4). Statistics are shown in the statistics view (5).*

tightly integrates data diagnostics and data transformation capabilities.

To succeed in her tasks the user must overcome a number of challenges:

The first challenge to the design is to trade-off a compact representation with a faithful representation. A compact representation improves the performance of the analysis on large repositories by eliminating irrelevant parts of the data. It is up to the domain expert to define 'relevance'. Hence, the expert must stay aware if and how a preprocessing step affects the outcome of the analysis.

The second challenge is not to 'overfit' the pipeline. On the one hand, visual inspection and testing is not feasible for a large repository. On the other hand, designing the pipeline using a single series does not always generalize and some important properties of other series might not be processed correctly. Instead the user is required to select a few time series to control and test the design. To increase the robustness of the preprocessing, this selection should at least approximate the diversity of the repository.

The third challenge is the size of the design space. For a typical operation like *sampling* a number of different methods are available. In turn, most of these methods have at least one parameter. Most operations can be combined freely, making the task to search for a suitable, if not optimal, solution as complex as the analysis itself.

The contribution of our work is as follows:

Firstly, we present a system for the interactive design and control of a time series preprocessing pipeline. It provides user support for the composition of preprocessing operations like data cleaning, data reduction an others. These operations can be chosen from a toolkit and arranged freely to define the pipeline. The effects of the preprocessing can be investigated by a visualization of the time series for every step of the pipeline. The approach is designed for expert users and non-expert users alike. In particular, the user is not required to do programming or scripting. The design of the pipeline can be monitored by time series visualizations, showing the effects of every module applied to representative time series. The resulting preprocessing transforms single raw time series into one or multiple descriptors each, depending on the requirements of the following analysis. Since our focus is on the refinement of single time series data, any metadata attached to a raw time series is kept with the corresponding descriptors for further reference. Analytical operations including an aggregation of multiple time series (like clus-

tering, merging of time series etc.) is not considered part of the preprocessing here.

Secondly, the user is supported in the selection of appropriate parameters for every module. An optimal choice of a preprocessing setup actually requires a search in a multi-parameter space. While the user can only change one parameter at a time, we aid her choice by generating an ensemble of alternative parameter values. Alternative results are shown in the time series. An inspection of the alternative effects leaves visual hints for potential optimization.

Thirdly, representative candidates of input time series are suggested to design and test the preprocessing pipeline. At the start of the preprocessing, reliable similarity measures are not available for analysis. Instead we estimate the time series variability by statistical properties. Candidates are suggested for user selection which cover most of this variability to establish the boundaries for the design.

We illustrate the use of our system in collaboration with researchers setting up a digital library for scientific climate data. We selected two tasks with two different preprocessing requirements. For each of these tasks we show how the pipeline is implemented after diagnosing the time series and how adaptions are made to meet the requirements.

The paper is organized as follows: Section 2 gives an overview of existing techniques and toolkits for time series preprocessing. In Section 3 we describe the core of our work, the editor for the preprocessing pipeline including the toolkit and views to try and test methods and parameters. We apply our toolkit to a scientific repository of climate data. In Section 4 we show the design of the preprocessing pipelines for two scenarios posing two different requirements for the analysis and preprocessing of this data. Section 5 summarizes the contribution of our paper.

2. Related Work

In the field of time series analysis, a diversity of analytical tasks exists. Current approaches differ by the targeted user group, the application domain and the analysis goal [AMST11]. For that reason alone the degrees of freedom in *time series preprocessing* are comprehensive [DTS*08, WL05], not only in the choice of methods but also for setting required input parameters [KLR04]. Moreover, many sources of data problems and varying levels of data quality [KHP*11] exacerbate the complexity of time series preprocessing tasks. We review both time series preprocessing techniques and data preprocessing workflows.

2.1. Time Series Preprocessing, Descriptors and Distance Measures

Data cleaning is important to ensure data quality [KCH*03, KHP*11]. Prominent challenges are missing value handling, noise and outlier detection [CBK09] (cf. Figure 3) and avoiding non-equidistant representations [WL05].

Data normalization solves problems with comparing data of different scales or translations. Normalization is essential for natural and subjectively correct similarity calculations [KK03, WL05, GAIM00]. Normalization can have local or global effects, depending on the position within the time series pipeline when applied.

Data sampling is an important method to reduce the amount of data. Data sampling can be defined as a subcategory of data reduction, surveys also describing data reduction in general are given in [Fu11, KK03, WL05]. Data percentage sampling, date-sensitive methods or variants that imply value-specific properties are applied [GAIM00, Fu11].

Data segmentation is applied on time series, since in many indexing, classification and clustering approaches, only subsequences are considered [Fu11, KK03]. Segmentation approaches for patterns with equal length [GAIM00] and unequal length [KCHP01] exist.

Time series descriptors are compact representations of time series raw data, that preserve the relevant information [DTS*08, KK03, LKLC03, Fu11]. Approaches with high compression rates which simultaneously preserve most of the information have been presented. However, distorting the shape of the time series creates problems if compression rates are too high [Fu11], which is also called the trade-off between compression and fidelity [KCHP01].

Similarity measure selection is highly domain- and application-specific process and therefore a challenging task [DTS*08, KK03, WL05, GAIM00]. In many time series applications, the used similarity measures are predefined.

2.2. Visual Analytics Workflows

For the visual analysis of data in general, a great variety of approaches has been presented to date. We refer to the book *Visualization of Time-Oriented Data* [AMST11] for a survey about relevant visualization and visual analysis techniques for time series data. Even though data analysts firstly spent large parts of their time on data cleaning and other preprocessing tasks before actual data analysis tools are executable [KCH*03], comparatively little research advances have been made in how interactive visualization can advance data preprocessing [KHP*11]. We identify works that relate to research on visual-interactive data preprocessing.

Regarding multidimensional data in general, several approaches for visual-interactive analysis exist. For example, an assisted descriptor selection approach based on visual comparative data analysis can be found in [BvLBS11]. In [SS04], the exploratory analysis of multidimensional datasets is supported by a rank-by-feature prism. The user can detect interesting features by choosing statistical ranking criterions (e.g. normality of the distribution) and manually select relevant features in 1D and 2D visualizations. Pretorius et al. [PBCR11] present a technique that supports users

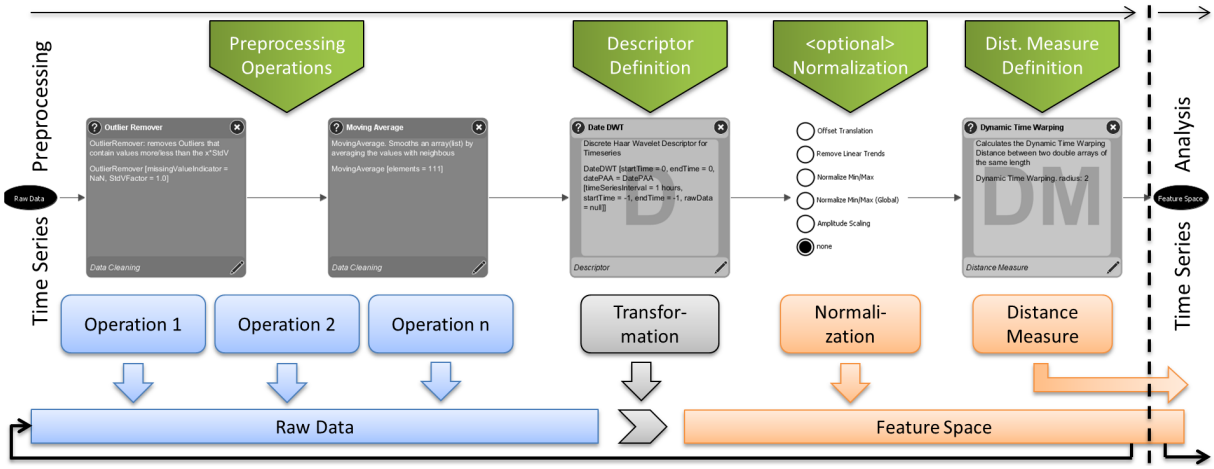

Figure 2: *The time series preprocessing workflow. In the raw data space the user can create a number of operations with the toolkit. A descriptor transforms the raw data into the feature space, the basis for subsequent time series analysis approaches.*

in visually analyzing relationships between the input parameter space and the corresponding output in the field of image analysis. In [IMI*10], a system for visual-interactive dimensional analysis and reduction is introduced. The user can combine a series of analysis steps, change parameters, and get direct visual feedback of the intermediate results. Predefined workflows consisting of several analysis steps can be stored and reused in later analysis processes. These systems guide the users in the analysis of multidimensional data via workflows. However, non of them focus on the specific characteristic of time series data in particular.

An example for the visual-interactive analysis of time series data is realized with the ChronoLenses interface [ZCPB11]. The system focuses on the visual exploration of time series by enabling the user with real-time transformation of selected time series intervals, and pairwise comparison of different time series. However, the authors use cleaned time series data. Therefore, preprocessing in general and the guidance towards appropriate parameter settings is not the focus of their approach. Further approaches for visual-interactive time series analysis are presented in [HDKS05], where importance-driven layouts guide the users in finding interesting time series, and [SSW*12], where the user can compare different parameter setups for the detection of local interest points. In [SBVLK09], clustering of time series trajectory data is performed with enhanced visual-interactive monitoring and controlling functionality.

To the best of our knowledge, there does not exist any technique that focuses on both the visual-interactive definition of an analysis workflow and the user-guidance in setting appropriate parameters for time series preprocessing.

3. Visual Analytics for Time Series Preprocessing

In this section, we introduce our approach for visual-interactive time series preprocessing. Subsection 3.1 presents the idea and the additional value of this work. Each of the following Subsections deals with one contribution, as mentioned in the introduction: user support for the definition of preprocessing scenarios, guidance for the parameter selection for individual preprocessing routines, and supporting the selection of representative testing data.

3.1. A Visual-Interactive Pipeline

We derive three insights from our review of the related work. First of all, we assert that a large number of algorithms for time series preprocessing has been established. However, visual-interactive representations of time series preprocessing operations are scarce, [KCHP01] may serve as an exception. In addition, although visual analysis of time series as such is a popular topic of research [AMST11], hardly any visual-interactive time series preprocessing applications are proposed. In fact, most approaches use preprocessing as a 'black-box'-approach. Finally, we observe that visual preprocessing of other data has become popular, combining algorithmic power with human capability of detecting patterns and steering the preprocessing process [IMI*10, BvLBS11].

We introduce a visual-interactive system for the generation of time series preprocessing pipelines, the conceptual workflow is shown in Figure 2. In the following, we call the preprocessing pipeline a time series *scenario*. We choose a generalizable approach for time series preprocessing. Beginning with the selection of raw data a variety of preprocessing operations can be added to the pipeline and (re-)arranged in arbitrary order. For most scenarios, however, the goal is a compact representation of the time series [KK03], commonly called a *descriptor*. For the definition of a descriptor, the pipeline may include a transformation of the raw data to a feature space, the outcome is a so called feature vector. For example, the Discrete Fourier Transformation may be applied to transform the time series into the signal space.

After an optional normalization step, a distance measure is defined to complete the time series scenario.

We aim to make the different operations as exchangeable and compatible as possible. Hence, the data model of our input time series consists of a list of so-called time-value pairs, each containing a time stamp and a corresponding value. This data model is able to represent virtually all possible characteristics of time series data like non-equidistant time stamps or missing values. Attached attributes like 'location on earth' (meta data) are kept with the produced feature vectors for the subsequent analysis task. Some preprocessing routines split one raw data into many (e.g. segmentation to patterns), which then also holds for the corresponding meta data. Additional qualitative results produced by preprocessing routines (e.g. compression rate) are adhered as additional meta data and displayed in the *statistics view* (5) (cf. Section 3.2). As a consequence of this generalizable approach, our system is not limited to a single time series input format. It only takes the effort of implementing an interpreter interface to make new data sources accessible.

Our intended users are data mining and machine learning experts, but also domain experts with the need to process time series data. Thus, we argue that typically users either have little data domain knowledge or are no experts in scripting interfaces. The visual-interactive approach, combined with user support is aimed to open time series preprocessing workflows for broad user groups. During the selection of preprocessing routines, preview visualizations simplify the selection process. The visualization of alternative parameters helps to discover the impact of distinct degrees of freedom in the time series.

3.2. System and Views

The graphical user interface of our approach consists of five components (see Figure 1). The preprocessing modules needed for the design of the pipeline are provided to the user in a *preprocessing toolbox* (1). The available tools are structured into 6 classes of operations listed below. For each class we implemented a number of alternative approaches.

- data cleaning (e.g. missing values, moving average, etc.)
- data reduction (e.g. sampling methods)
- data normalization (e.g. min-max normalization)
- data segmentation (e.g. subsequence and pattern def.)
- descriptors (e.g. PAA, DTW, DFT, PIP, etc.)
- similarity measure (e.g. Euclidean distance, etc.)

The *workflow view* (2) is the visual representation of the preprocessing pipeline. All modules that possess user adaptive parameters are displayed as rounded rectangles. A module can be added by drag-and-drop from the preprocessing toolbox. Each module glyph contains the name and an external hyperlink for additional information (top), the description and parameter setting (middle), as well as the operation class and an edit button (bottom). Parameter changes can be set when the module is added or afterwards by clicking the edit button. By selecting a module, the preprocessing pipeline is executed on a chosen time series up to this stage. Re-ordering of the pipeline is possible by dragging modules to other positions. Load and Save buttons at the lower right of the system enable to re-edit and branch scenarios at a later time. Since the optional normalization step in the scenario does not require parameters, a radio button-like glyph is used for the representation. Straight forward, the user can select the favored normalization variant with a single click.

The *raw data selection view* (3) is divided in two lists of used and unused raw data represented as bars. The bar size corresponds to raw data values according to a user-definable statistical property (the median in case of Figure 1). The gray value displays the degree of dissimilarity of unused raw data (upper list) in contrast to the ones already visualized. Raw data with black color hold maximum dissimilarity and thus are mostly recommended to visualize. Section 3.4 further describes how this user guidance is algorithmically provided.

In the *detail view* (4) the user can trace the execution by inspecting intermediate preprocessing results of the time series visualized as a line chart. The detail view provides direct feedback on the initial raw data (upper), the last (middle), and the current state of the preprocessed time series (bottom). Section 3.3 describes how the detail view supports the user in setting appropriate preprocessing parameters.

In the *statistics view* (5) statistical information about the selected time series raw data and the preprocessed data is provided to the user. Every change of the statistical information resulting from a preprocessing operation is highlighted in red. By showing this additional meta data, the user can immediately track the effects of the preprocessing.

The proposed workflow is straight forward. The input data is provided in the raw data selection view (3). The user selects the time series to be observed during the analysis. A set of preprocessing modules can be added from the preprocessing toolbox (1) to the workflow view (2), parameter changes, module re-ordering and result-storing is possible at any time. By selecting a module, the user can observe respective modifications on the time series in the detail view (4). Additional statistical information can be monitored at the statistics view (5). After finalizing the scenario by defining a normalization and a suitable similarity measure the preprocessing pipeline completely configured. The scenario is ready for an execution on all time series coming from the current data source, or another data source at a later data. The system supports loading and remodeling of saved scenarios, which enables building branches of preprocessing pipelines.

3.3. User-support for Parameter Setting

Section 3.2 described how preprocessing modules are arranged to an entire time series scenario. In the following, we give details about the parameterization of a single module, which is a problem in itself. Each preprocessing module

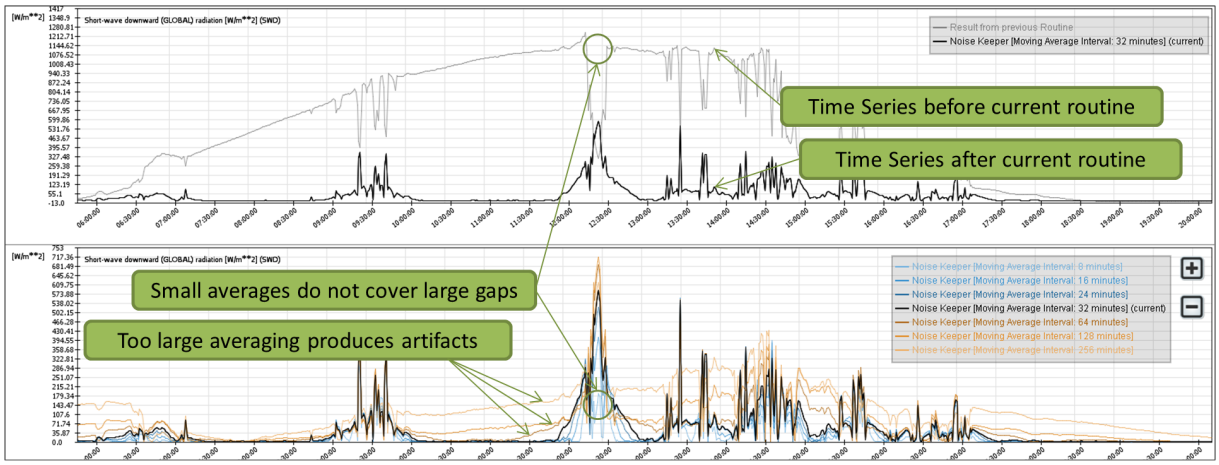

Figure 3: *Emphasizing noisy data by subtracting a moving average. Fine-tuning to a 32 minute moving average interval turns out to be a satisfactory parameterization. The trade off between fidelity and generalizability is accomplished.*

in the system provides ensembles of n alternative parameter values, as appropriate, whereupon n is a user parameter. The time series arising from alternative parameterizations are visualized as line chart bundle in the detail view. Figure 3 demonstrates parameterizations of a *NoiseKeeper* module, alternatively to the examples in the application section. The current parameterization of the particular preprocessing routine is always displayed in black. Curve progressions based on other parameterizations are displayed with bipolar color value from blue to brown. Color brightness is used to illustrate the divergence of each alternative parameterization compared to the current, where darker values are more similar. A colored legend for each parameterization is given at the right of the time series visualization window. In addition, the middle part of the detail view (cf. Figure 1) compares the current time series to the result of the previous preprocessing module (gray) and the upper chart provides a comparison to the raw input data (gray).

We provide two ways for the selection of alternative parameterizations. The user can manually edit the parameterization in the preprocessing pipeline by hand as described in Section 3.2, or choose one of the alternative parameterizations by clicking on the colored legend in the time series visualization window. The number of alternative parameterizations can also be adapted by clicking the 'plus' and 'minus' button at the upper right of the window. Thus the interval of the covered parameter space is also adapted. Incrementing the number n of alternative parameter values (by a larger and a smaller one) increases the parameter interval by the current parameter value times 2 power n.

3.4. Guided Selection of Representative Time Series

We tackle the problem of selecting representatives from a pool of thousands of previously unknown input data. Since scenarios are built to process all raw data, the user needs a method to verify the 'generalizability' of the produced time series preprocessing pipeline based on a small subset of visualized input data. Hence, guidance is needed. To help the user in selecting appropriate time series samples, the system uses a statistics model that estimates the dissimilarity of unused raw data in contrast to the ones already visualized. We define three requirements for the statistical model:

1. Robustness to low quality raw time series data; no preprocessing before the calculation of the statistical model
2. Value and shape-based raw data discrimination
3. Low redundancy between the model features

We use a combined model based on (a) two statistical properties that concern the values of a distinct time series (median, and standard deviation) and (b) three properties that originate from the decomposition of time series (trend, periodicity, and noise). This 5-dimensional feature set is extracted from each input time series, and min-max normalized, afterwards. We recommend to weight each dimension equally in the first iteration of the analysis. The calculation of the Euclidean distance between all used and all unused time series feature sets represented by numerical vectors, produces a single value for each unused time series: the dissimilarity. This property is used for coloring (white is similar and black is unsimilar), to guide the user. The raw data selection view (3) in Figure 1 illustrates our concept.

This similarity concept is independent of the preprocessing operations of the system and of similarity measure chosen at the end of the preprocessing. We are using these parameter-free statistical features to 'boot-strap' the analysis, if no prior knowledge allows for a systematic selection of representatives. After studying hundreds of different input time series, we came to the conclusion that values and shapes are well discriminated with our model. If, however, new insights suggest a refinement of the representative set, the user may change the propery weighting or freely chose an entirely different representative as well.

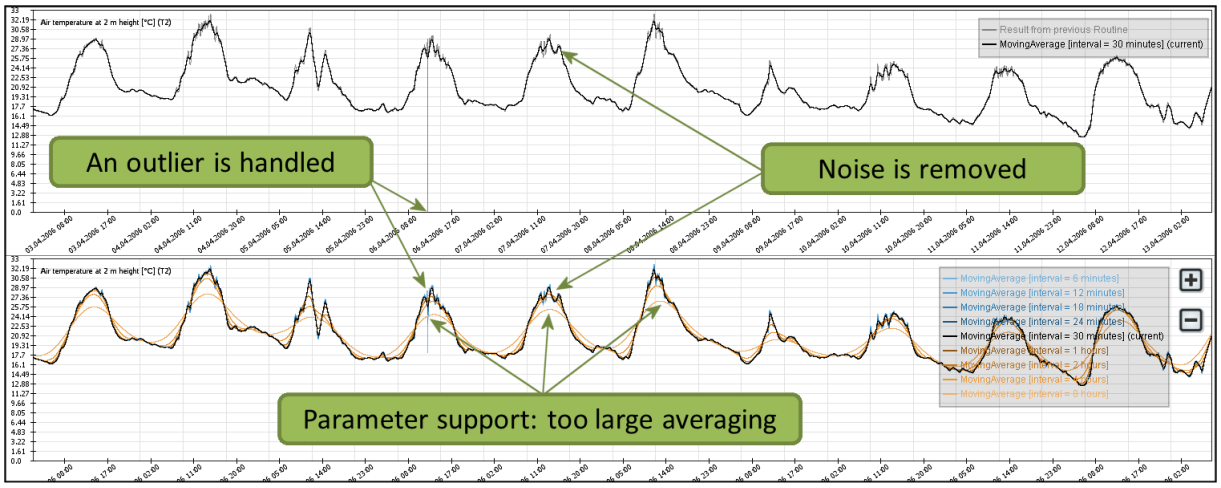

Figure 4: *A 30 minute moving average routine reduces noise too insufficient. For a better result we chose a 1 hour parameterization that smooths the curve yet without too much averaging.*

4. Application

In our case study, we apply our system to tasks coming from the *Digital Library* domain. In the field of digital libraries, scientific research data is considered valuable because research data possibly yields information that has not been discovered yet. It is also challenging because of the size, the heterogeneity and the many degrees of freedom of research data (cf. Section 2.1).

The project *VisInfo* [BBF*11] aims at providing visual search functionality for archived time-oriented research data. In this project it turned out, that domain specialists, librarians, and computer scientists have different time series similarity notions. This complicates the definition of content-based queries for visual search operations in time series data. In a time-intensive case study, we included the researchers and librarians in the definition of preprocessing scenarios for (A) search and clustering (cf. Section 4.2), and (B) compact representations for visualization (cf. Section 4.3). The lack of a system for comprehensive illustration and execution of time series preprocessing impeded the process of defining these final scenarios at the start of the project. In order to provide domain experts with a tool for defining their own time series scenarios with direct visual feedback of its data transformation steps we designed this system.

4.1. Dataset

We focus on *time-oriented scientific research data*, gathered in the scope of the Baseline Surface Radiation Network (BSRN) [ODF*98]. The aim of BSRN is to detect important changes in the earth's radiation field which may be related to climate changes. Thus, up to 100 parameters based on radiation and meteorological characteristics are collected at BSRN stations all over the world, recorded up to a temporal resolution of one measurement per minute. Common physical units include atmospheric pressure, relative air humidity, temperature and a variety of radiation-based measurements like short-wave downward radiation and long-wave upward radiation. The data is archived, published and distributed by the open access library PANGAEA, a data repository operated by the Alfred Wegener Institute (AWI) for Polar and Marine Research in Bremerhaven, Germany. Most of the data is freely available and can be used by domain researchers and interested users to validate hypotheses in earth observations, e.g. regarding climate change.

For our use cases, we choose a data subset from the BSRN data pool [BK12], including 6813 files of monthly measurements originating from 55 BSRN stations on earth. The number of incorporated measurements rises above 500.000.000 data points. Together with the meta data the disc space of the input data is above 20 gigabytes. Thus, data compression will be an important objective in both use cases.

4.2. Use Case A: Extracting Features for Search and Clustering of Time Series Data

Our first use case deals with the extraction of feature vectors from time series data. Assembled in an index structure, the features are used to support fast content-based search and clustering functionality in a web portal for time-oriented research data. Since the user can sketch a time series curve as a query for visual search (or use an example curve), we need to specify a similarity measure for comparing time series data. This similarity has to reflect the similarity notion of climate researchers - the system should find time series that the researcher would expect to. Knowledge of domain expert users has to be included in the time series preprocessing process.

Over a period of two years, we defined a preprocessing pipeline for this purpose within a case study. In the following, we describe a workflow for the creation of time series

scenarios that can be executed within minutes. Here, the pronoun 'we' indicates the cooperation with the domain experts.

We describe the preprocessing workflow for temperature measurements. This data is imported as the raw data source in the pipeline. Next, the statistical information about the raw data is calculated and visualized in the raw data selection view. The measurements differ in a range of values between -70°C and 50°C. The missing value ratio for most raw data is below 10%, the dominating quantization of the measurements is one minute. Most time series have periodic curve progressions with a daily duration, the scientists call this the 'diurnal variation'. We use the raw data selection view and define a test set of raw data with great variabilities regarding median, standard deviation, degree of noise, and periodicity.

We clean the input data to provide a consistent data quality. The *MissingValueRemover* is added to the pipeline and missing values are deleted from the time series. With the *TimeStampSorter* and the *IdenticalTimeStampMerger*, two mandatory routines are established. Since we want to define similarity depending on the overall shape of time series, outliers and local noise can be neglected. We select a *MovingAverageCalculator* to address this task, the results can be seen in Figure 4. By visual comparison we find out that a 30-minute kernel parameterization is too small to reduce noise sufficiently. Hence, we choose a 1-hour interval. The data is now cleaned for subsequent preprocessing steps.

We follow the advice of the domain experts and normalize the time series with a specific *TrueLocalTimeNormalizer*. This method ensures that measurements from all over the globe become comparable, since the time axis is synchronized with the respective time zone. We register that natural periodicities in earth observation mainly have the duration of days and years. We apply the *TimeSeriesSegmentation* routine on our monthly data to receive daily time series patterns, all starting at midnight. Daily patterns with too large gaps compromise the similarity notion. We remove patterns with empty fragments longer than 4 hours with the *UnderSampledDataRemover*. Afterwards, an additional *LinearInterpolation* routine ensures a sampling rate of at least 1 hour.

As a next step we define a descriptor that represents the original time series with respect to the similarity notion of a climate researcher. We choose the PAA descriptor with a quantization of 1 hour as it can be seen in Figure 1. This parameterization is chosen since on the one hand the researcher affirms a sufficient fidelity to the original data and on the other hand the descriptor exhibits a strong compression of 60:1. A higher compression is not feasible, because of too inaccurate results (see the brown line-charts in Figure 1).

Since our domain expert wants to compare (a) absolute values of the measurements (differences between high and low temperatures) and (b) the shape (the slope of the daily temperature curve progressions not depending on absolute values), we decide to branch our pipeline and provide two different preprocessing scenarios. Scenario (b) is additionally normalized with a *MinMaxNormalization* to receive relative temperature curve patterns, while scenario (a) contains absolute values without normalization. Finally we define the *EuclideanDistance* as our similarity measure, since this is a common way for the earth observation scientists to compare their measurements.

We save our preprocessing scenarios. Thus, we are able to modify our preprocessing pipeline if changes in the requirements arise in future.

We assess the outcome of our time series scenario. With this pipeline we achieve a reduction of memory space of more than 98%. The computation of similarities between time series is accelerated considerably. Overall the search results of our prototypical time series search application were judged as very promising by the domain researchers. However, a full evaluation of the preprocessing system and the time series search application is still future work.

4.3. Use Case B: Extracting Features for Visual Representation of Time Series Data

Our second use case deals with the visual representation of large amounts of time series raw data for a web application. A compact representation of each time series has to be stored in a database for quick access. The goal is to optimize the compression of the raw data for minimal memory consumption and quick data transfer. At the same time the important features of the curve progressions have to be preserved for the web visualization. The knowledge of domain experts is crucial for the extraction of representations, since they can define the important visual features to be preserved for a representative line chart visualization.

This use case was developed in the project together with a domain expert from the AWI and the librarians who will host the web application in future. We reconstruct and refine the workflow as follows.

We choose the solar-dependent short-wave downward radiation (SWD) measurements as our data source. Again, the statistical information about the raw data is calculated and visualized in the raw data selection view. In contrast to the first use case the range of values is different. It spans from 0 W/m^2 (at night - no radiation) to approximately 1500 W/m^2 in desert-like environments. The standard deviation and the degree of noise is larger than in temperature curves, which constitutes an additional difficulty for the time series preprocessing pipeline to be designed. SWD measurements contain a variety of spikes, caused by changing terms of cloudiness. This effect (and its geographic dependency) is important, e.g. for photovoltaic power generation. A popular search scenario is the identification of so-called clear-sky conditions, when the sky is free of any clouds. In this case, a curve progression is nearly sine-like as it simply follows the solar altitude. However, our scenario also has to preserve the fidelity

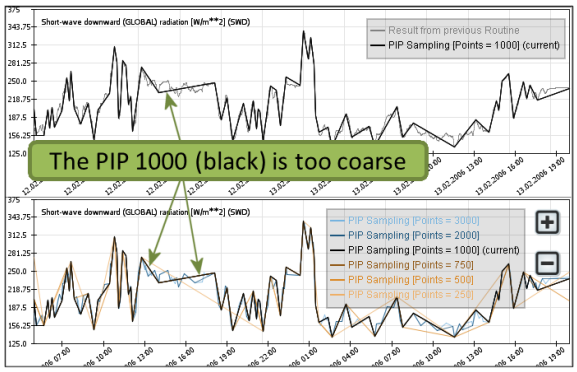

Figure 5: *The PIP 1000 causes losses in fidelity. However, we use the PIP 2000 (dark blue) that has almost the same visual appearance as the raw data (gray).*

of cloudy weather conditions when measurement curves are very noisy (cf. Figure 5).

Firstly we reuse the pipeline from use case A. Subsequent to the *MissingValueRemover*, the *TimeStampSorter*, and the *IdenticalTimeStampHandler*, we create a new branch for use case B. To reduce the fraction of 'micro noise', we apply a 3-minute *MovingAverageCalculator* (the measurements are mostly taken with a one minute quantization).

In this scenario we can neglect the sequence segmentation since we want to visually represent the whole time series in the web application. As a next step we have to define a descriptor that represents the original time series preserving its important visual features, and at the same time optimizes the compression level. We choose the *PerceptionalImportantPoints* descriptor (PIP) as applied in [ZJGK10], since the algorithm is data adaptive and in particular sensitive to the preservation of fidelity. At the start of our project this pipeline has been designed 'by hand'. It compressed the time series to 1000 data points per month. Now, we can see with our system that this is not sufficient in general. The degree of noise in some raw data emerges to be larger than initially expected, an example curve is shown in Figure 5. The refinement of the parameterization to 2000 data points per day reduces losses in fidelity. A benefit of the visual parameter support and the guided selection of input data.

As a result of our preprocessing pipeline we achieve a reduction of memory space of more than 95% with a satisfying visual representation of the time series, again acknowledged by the domain experts. Our prototypical visual time series search system can now be equipped with a linechart visualization that resembles the original raw data.

5. Conclusion and Future Work

The motivation of our work was to make time series preprocessing visible and accessible to domain experts. We presented an approach for the interactive diagnosis, design and control of preprocessing methods used for time series analysis. It allows to compose a pipeline of individual operations drawn from an extensible toolkit. Different steps of the pipeline can be arranged in a very flexible way, because of a common underlying data structure. The composition is guided by a series of line-charts showing intermediate results and statistics. A user learns about the characteristic properties of the data, and how these properties are changed by every step of the pipeline. Many unwanted effects can be pinpointed to a specific operation or parameter. In addition the system provides visual cues for the defensible improvement of parameter settings and the selection of testing data. By exposing the preprocessing and its effects in a visual way, the confidence of the experts to the results have been increased greatly. In the collaboration between experts parameter settings are now open for explanation and discourse. While we illustrated the use of the system with climate data only, the system can be used for time series preprocessing in many other domains. This even applies to fields that use specialized data transformations.

We consider our work a starting point which can be extended with respect to a number of aspects. For example, our system does not include a comprehensive toolkit of transformation methods known from literature. New operations, similarity measures or descriptors can be added to the toolkit, without compromising the system's architecture. A more challenging extension is to expose a relation between the raw data space, the feature space and the results of the analysis. While the first draft of the preprocessing pipeline often draws upon limited knowledge, new analytical findings necessitate further adaptation. Exposing the right leverage points for this adaptation would help to improve the pipeline even faster. Another way to use experts knowledge for data preprocessing is to consider labeled raw data. Labeled data usually serves as a 'ground truth' to measure classification quality. However, since differences and similarities imposed by the labeling should be preserved by the preprocessing, it also could be used as a ground truth to measure its fidelity.

In summary, shifting our attention from visual analysis to visual preprocessing was beneficial. Our collaboration showed that the ability to expose and discuss decisions and their effects is crucial to improve analytical processes and results.

Acknowledgments

We thank the Alfred Wegener Institute (AWI) in Bremerhaven, Germany. The researchers helped us in understanding the data domain, in selecting an appropriate PANGAEA data subset and in creating time series scenarios. We thank Tobias Schreck from the University of Konstanz for a profound and inspiring collaboration. This work was supported by a grant from the Leibniz Association as part of the 'Joint Initiative for Research and Innovation' program.

References

[AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of Time-Oriented Data*. Springer, London, UK, 2011. 41, 42

[BBF*11] BERNARD J., BRASE J., FELLNER D., KOEPLER O., KOHLHAMMER J., RUPPERT T., SCHRECK T., SENS I.: A visual digital library approach for time-oriented scientific primary data. *Springer International Journal of Digital Libraries, ECDL 2010 Special Issue* (2011). 45

[BK12] BERNARD J., KÖNIG-LANGLO G. SIEGER R.: Time-oriented earth observation measurements from the baseline surface radiation network (bsrn) in the years 1992 to 2012 , reference list of 6813 datasets. doi:10.1594/pangaea.787726, 2012. 45

[BvLBS11] BREMM S., VON LANDESBERGER T., BERNARD J., SCHRECK T.: Assisted descriptor selection based on visual comparative data analysis. *Comput. Graph. Forum 30*, 3 (2011), 891–900. 41, 42

[CBK09] CHANDOLA V., BANERJEE A., KUMAR V.: Anomaly detection: A survey. *ACM Comput. Surv. 41*, 3 (July 2009). 41

[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. *Proc. VLDB Endow. 1*, 2 (Aug. 2008), 1542–1552. 41

[Fu11] FU T.-C.: A review on time series data mining. *Engineering Appl. of Artificial Intelligence 24*, 1 (2011), 164–181. 41

[GAIM00] GAVRILOV M., ANGUELOV D., INDYK P., MOTWANI R.: Mining the stock market: Which measure is best. In *In proceedings of the 6 th ACM Int'l Conference on Knowledge Discovery and Data Mining* (2000), pp. 487–496. 41

[HDKS05] HAO M. C., DAYAL U., KEIM D. A., SCHRECK T.: Importance-driven visualization layouts for large time series data. In *INFOVIS* (2005), IEEE Computer Society, p. 27. 42

[IMI*10] INGRAM S., MUNZNER T., IRVINE V., TORY M., BERGNER S., MÖLLER T.: Dimstiller: Workflows for dimensional analysis and reduction. In *Proceedings of the 5th IEEE Conference on Visual Analytics in Science and Technology (VAST)* (Florida, USA, 2010), IEEE Computer Society. 42

[KCH*03] KIM W., CHOI B.-J., HONG E.-K., KIM S.-K., LEE D.: A taxonomy of dirty data. *Data Min. Knowl. Discov. 7*, 1 (Jan. 2003), 81–99. 41

[KCHP01] KEOGH E., CHU S., HART D., PAZZANI M.: An online algorithm for segmenting time series. In *In ICDM* (2001), pp. 289–296. 41, 42

[KCPM00] KEOGH E., CHAKRABARTI K., PAZZANI M., MEHROTRA S.: Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems 3* (2000), 263–286. 40

[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: visuatizations and transformations for usable and credible data. *Information Visualization 10*, 4 (Oct. 2011), 271–288. 39, 41

[KK03] KEOGH E., KASETTY S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Min. Knowl. Discov. 7*, 4 (Oct. 2003), 349–371. 41, 42

[KLR04] KEOGH E., LONARDI S., RATANAMAHATANA C. A.: Towards parameter-free data mining. In *Proceedings of the ACM SIGKDD int. conf. on Knowledge discovery and data mining* (New York, NY, USA, 2004), KDD '04, ACM, pp. 206–215. 41

[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. 41

[ODF*98] OHMURA A., DUTTON E. G., FORGAN B., FRÖHLICH C., GILGEN H., HEGNER H., HEIMO A., KÖNIG-LANGLO G., MCARTHUR B., MÜLLER G., PHILIPONA R., PINKER R., WHITLOCK C. H., DEHNE K., WILD M.: Baseline surface radiation network (BSRN/WCRP): New precision radiometry for climate research. *Bull. Amer. Met. Soc. 79* (1998), 2115–2136. 45

[PBCR11] PRETORIUS A. J., BRAY M.-A., CARPENTER A. E., RUDDLE R. A.: Visualization of parameter space for image analysis. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec. 2011), 2402–2411. 41

[SBVLK09] SCHRECK T., BERNARD J., VON LANDESBERGER T., KOHLHAMMER J.: Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization 8*, 1 (Jan. 2009), 14–29. 42

[SS04] SEO J., SHNEIDERMAN B.: A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *in Proceedings of IEEE Symposium on Information Visualization* (2004), pp. 65–72. 41

[SSW*12] SCHRECK T., SHARALIEVA L., WANNER F., BERNARD J., RUPPERT T., VON LANDESBERGER T., BUSTOS B.: Visual Exploration of Local Interest Points in Sets of Time Series. In *Proc. IEEE Symp. on Visual Analytics Science and Technology (Poster Paper, accepted for publication)* (2012). 42

[WL05] WARREN LIAO T.: Clustering of time series data-a survey. *Pattern Recogn. 38*, 11 (Nov. 2005), 1857–1874. 41

[ZCPB11] ZHAO J., CHEVALIER F., PIETRIGA E., BALAKRISHNAN R.: Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2422–2431. 42

[ZJGK10] ZIEGLER H., JENNY M., GRUSE T., KEIM D. A.: Visual market sector analysis for financial time series data. In *IEEE VAST* (2010), IEEE, pp. 83–90. 47