Erschienen in: Computer Graphics Forum ; 33 (2014), 3. - S. 401-410

Visual Analysis of Time-Series Similarities for Anomaly Detection in Sensor Networks

Martin Steiger[1], Jürgen Bernard[1], Sebastian Mittelstädt[2], Hendrik Lücke-Tieke[1], Daniel Keim[2], Thorsten May[1], Jörn Kohlhammer[1]

[1]Fraunhofer IGD, Germany
[2]University of Konstanz, Germany

Abstract

We present a system to analyze time-series data in sensor networks. Our approach supports exploratory tasks for the comparison of univariate, geo-referenced sensor data, in particular for anomaly detection. We split the recordings into fixed-length patterns and show them in order to compare them over time and space using two linked views. Apart from geo-based comparison across sensors we also support different temporal patterns to discover seasonal effects, anomalies and periodicities.

The methods we use are best practices in the information visualization domain. They cover the daily, the weekly and seasonal and patterns of the data. Daily patterns can be analyzed in a clustering-based view, weekly patterns in a calendar-based view and seasonal patters in a projection-based view. The connectivity of the sensors can be analyzed through a dedicated topological network view. We assist the domain expert with interaction techniques to make the results understandable. As a result, the user can identify and analyze erroneous and suspicious measurements in the network. A case study with a domain expert verified the usefulness of our approach.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques H.5.2 [Information Interfaces and Presentation]: User Interfaces—User-centered design C.2.3 [Computer-Communication Networks]: Network Operations—Network monitoring

1. Introduction

The amount of sensor data has seen a rapid growth over the past years in many different applications and scenarios. In this paper, we focus on univariate sensors that measure a single variable over time. In many applications it can be safely assumed that the sensors measure variables that are explicitly or implicitly linked. We motivate our approach with a practical example: the exploratory analysis of the power consumption in a small to medium sized electrical grid.

In this example, the operators in the control rooms are interested in the recordings of the power consumption. They are interested if sensors at two stations that are connected by electric cables measure similar values or not. Depending on the task, the network analyst wants to know about daily, weekly and seasonal patterns and trends. In how far do the consumption patterns change over the year? What are the differences between workdays and weekends? What are the regional differences in the grid? These characteristics apply to other application scenarios as well (like, e.g. traffic analysis, water-level-predictions on rivers or logistics). While we aim for a generic application, we will motivate the definition of tasks and goals by transfer from the specific scenario.

Typical approaches include expert systems based on rule inference to analyze the system in a fully automated manner. These software systems usually operate in a black box manner that do not allow for user interaction. The user has to rely on the fact that the a-priori knowledge encoded in the rules is sufficient. On the other hand, performing a manual analysis of all sensor readings is often hardly feasible, because it is difficult to have an eye on all sensors at the same time. Also, it is quite difficult to derive trends and frequently occurring patterns from simple line chart plots. Therefore, a set of visual tools can successfully assist the human in the analysis process to become more effective and efficient.

The users that work with this kind of data are, for example, operators in the control room of an electrical grid. Their task is to monitor the state of the network using the measurements provided by the installed power meters across the country. The operators must be able to identify repeating patterns as well as anomalies with respect to changes over time

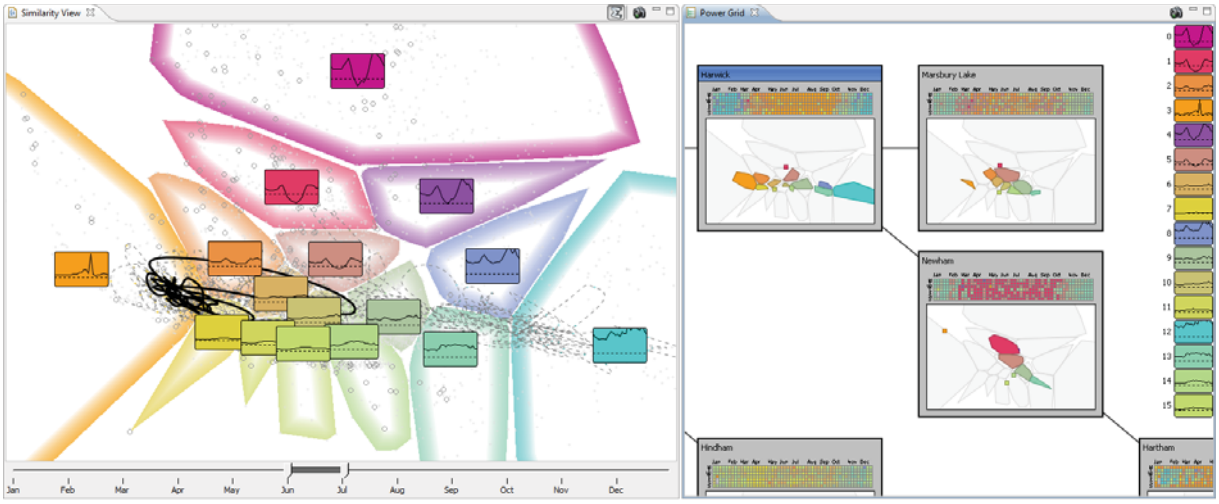

Figure 1: *A screenshot of the entire system. The similarity view on the left side shows all daily patterns of all sensors. Similar patterns are assigned to the same group and color. The change of patterns over time for a selected sensor is indicated by the black spline. The network view on the right side gives an overview of the network topology. The small calendar in the node glyph shows changes over time and a fingerprint view underneath shows the sensor patterns in the global context.*

and across different sensors. For that, it is important to recognize diverging states, anomalies and suspicious patterns as quickly as possible. More specifically, we have identified the following problems:

- Getting an overview of a sensor network is required in order to get an impression of the "big picture" and to identify potential problems.
- The user needs to analyze the network in space and time to find atypical patterns in the network.
- Comparison of different sensors at the same time but also the development of a single sensor over time is relevant.
- Based on the pattern similarity, the user must be enabled to quickly identify non-standard patterns and trends.

Based on these tasks, we derive a set of design criteria. Atomic entities (i.e. daily patterns) are analyzed with respect to three different criteria: based on content and relations to the geographic and temporal context. These different aspects must be linked to enable the analyst to provide additional insight and to solve multi-criteria problems. We identify the most appropriate visualizations with respect to the properties of the data and the user task:

- The data is recorded at several linked univariate sensors that measure the same physical quantity.
- The time-series data can be segmented into meaningful equally-sized day-long patterns.
- Interesting patterns are expected to be daily, weekly or yearly.
- The system must be able to robustly detect and deal with outliers and missing values.

We contribute a visualization system that is able to assist the analyst in dealing with these problems. It consists of two tightly coupled views that complement each other: a *Similarity View* and a *Network View* (see Figure 1). A topological map of the network gives a geo-based topological overview on the network in space and development of patterns over time for every sensor. Using a calendar-based visualization, the analyst is able to identify trends on different scales, based on individual sensors. As a result, the user can identify erroneous and suspicious measurements in the network. A similarity-based view gives important details on the global relations of different temporal patterns (in our example the power consumption over the day). The user can thus analyze daily patterns of the sensors, grouped by their pair-wise similarity. On demand, points that belong to the same sensor can be connected. This gives the analyst a quick overview on the variability of daily patterns over a period of time. If the patterns are very similar this spline would look like a tiny hairball and anomalies can be easily spotted. Tight linking between the two ensures that the user recognizes the same element and sets of similar elements in both views.

The paper is organized as follows: In Chapter 2, we discuss related analysis systems for time series data and sensor networks. Chapter 3 covers the algorithms and data structures before Chapter 4 describes visualizations and interaction. Results from the case study and the design study are presented in Chapter 5 and 6.

2. Related Work

We briefly discuss scientific work in terms of related concepts in this chapter, grouped by topic. In the presentation of our approach a short explanation of why we chose a certain technique is given.

2.1. Sensor network analysis

The analysis of sensor networks is often associated with a very specific application domain, be it wireless networks, surveillance systems or electrical grids. The GreenGrid visualization system of Wong et al. focuses on the analysis of very large electrical grids [WSM*09]. Similar to our approach, it illustrates the potential advantages of force-directed layouts based on physical properties over geographic coordinates. The visualization system of Hadlak et al. analyzes the temporally changing link quality of a wireless network [HSCW13]. The authors cluster time-series data and display the data in a node-link diagram, but we take this one step further and enable the analysis of repeating patterns which was not part of their work. Saraiya et al. conduct a user study to evaluate different node glyphs for graphs with multi-variate node attributes [SLN05], which we use as a guideline for our glyph design. Using a combination of spiral visualizations and treemaps, Janetzko et al. detect anomalies in power consumption data of commercial buildings [JSMK14]. Shi et al. demonstrate anomaly detection for multi-variate sensor data in hierarchical networks. In contrast to our work, the authors do not focus on pattern analysis [SLH*11].

2.2. Dimensionality Reduction

In order to make large data sets accessible to the user, a variety of data reduction techniques exist. One of the most used projections is Principal Component Analysis (PCA). Being a linear method, it is very sensitive to outliers and does not use the available display space too well. Multi-Dimensional Scaling (MDS), a group of methods for dimensionality reduction, is also very popular. Since its original presentation [Tor52], many variations have been developed [Kru64] and has gained popularity also in the graph drawing community [BP09]. Also, non-deterministic projection methods such as Stochastic Neighbor Embedding have been employed [HR02]. Using locally restricted projections, Joia et al. present not only a new projection approach, but also a comparison of different approaches [JPC*11]. Another survey is presented by Lee et al. who discuss dimensionality reduction schemes without user supervision [LV10].

2.3. Dimensionality Reduction Quality

With the reduction of data comes a loss of data quality. Many different measures are available that assess the quality of a given projection. A natural choice, in particular for MDS methods, is to use the weighted stress function as an indicator for the projection quality. Kruskal proposed a small variation of this stress function as well as some reference values for quality [Kru64]. However, measuring the quality with the same means as the actual algorithm seems to be an unreliable choice for MDS. Sips presents *"...two quantitative measures of class consistency, one based on the distance to the class's center of gravity, and another based on the entropies of the spatial distributions of classes..."* [SNLH09], which are robust against outliers. In the work of Bertini et al., an overview on many different quality measures, pros and cons as well as application domains can be found [BTK11].

2.4. Time Series Analysis

The analysis goals of this approach are influenced by social-temporal (daily, weekly), natural-temporal (daily, yearly), and geo-spatial variances in the data. We present related techniques for time series preprocessing and for visual-interactive time series analysis. The data mining community has spent great effort in the development of time series preprocessing techniques [DTS*08, Fu11]. Usually, pipelines are applied that may execute several cleaning, reduction, normalization, segmentation, or feature extraction steps on the underlying time series data. Recently, visual-interactive applications to support time series preprocessing and model creation have been presented [BRG*12, BAF*13]. Relevant overviews of time series visualization [AMST11] and the visual analysis of time and geo-spatial data [AAD*10] exist. We shed light on single techniques that are most closely related. Stoffel et al. present a client-server visual analytics systems for anomaly detection in computer networks [SFK13]. Its main views show a collection of vertically oriented line charts that are compared with a reference model of the data. An inspiring technique is the calendar view [VWVS99] by van Wijk et al. Similar to our approach clustering of daily patterns is applied to visually encode a calendar visualization. However, the calendar view differs in the chosen clustering technique, the color coding (which is not similarity-preserving) and a lack of spatial comparison capability. A technique that combines the comparison of daily temperature patterns and geo-spatial meta data information was presented in the digital library context [BRS*12]. However, a chronological representation of the daily patterns is not provided. Techniques that focus on the visualization of periodic time series data may rely on radial [ZFH08], cyclic [TS08], or on projection-based layouts [BWS*12, WG11].

3. Data and Algorithms of the System

In this chapter we present the data and algorithms for the visualization system. We do not use pattern shapes directly but provide similarity-based measures to support the identification of similar and different patterns, as well as projection and clustering techniques. In this way we support the user in the identification of both frequent patterns as well as outlier patterns. We first cover the data preprocessing routines before we explain the rationale behind the design decisions. Before we start, the data is condensed from high-dimensional data vectors to lower-dimensional feature vectors that are presumed to contain the majority of information and are faster and more robust to work with. We compute the similarity based on this data, before we aggregate similar

patterns into groups. This allows the user to get an overview of recorded measurements and identify trends and repeating patterns in the visualization.

3.1. Input Data

The input data is a collection of time series measurements of a single variable spanning over one year. The sensors that record the measurements are organized in a network structure. Nodes indicate sensors, the edges between two nodes indicate some kind of connectivity.

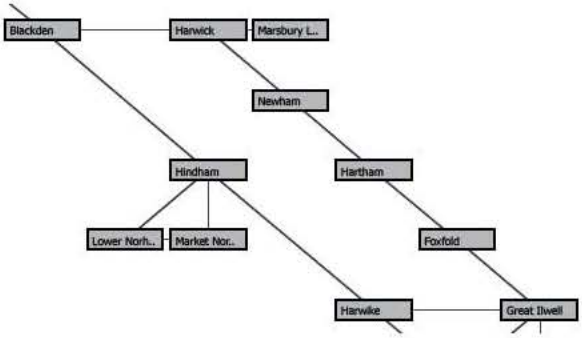

Figure 2: *A part of the sensor network, displayed as a octilinear topological map. At the cost of uniform edge lengths, geographical directions are preserved, if possible [NW06].*

Not all sensors have recordings for every time stamp, and day, respectively. Since missing values may be an important aspect for analytical tasks, our visual representations need to be sensitive to an explicitly defined missing value indicator. Another aspect of the data is that some patterns are partly filled with zeros. In some scenarios, zero can be interpreted as missing value, whereas in others, it cannot. We therefore do not assign a special meaning to this value. We prefer a shape-preserving in favor of a domain-preserving pattern comparison strategy. Thus, we apply a *standard score* normalization for the input data per sensor. To reduce the impact of outliers we previously apply a moving average procedure with a kernel range of one hour. The next step in the analysis process is the segmentation of the time series data into individual patterns. In our scenario, diurnal variations are the smallest repeating patterns and therefore the segmentation into days appears to be the right level of granularity.

3.2. Similarity Measures

Many different algorithms for measuring the similarity of time series data exist. Our system supports different analysis tasks and therefore supplies different similarity measures. For the analysis of values or changes in the values, the Euclidean distance is a useful measure to compare patterns. We argue in accordance to Hadlak et al. [HSCW13] that trend-based similarity measures support the user in finding simultaneous changes over time well. If the shape of the consumption pattern is of interest, the correlation coefficient and Dynamic Time Warping (DTW) are reasonable choices. The DTW algorithm compares two time series by aligning sequences of the data so that the distance between the two is minimal [BC94]. This makes DTW robust to shifts and length of the temporal sequences. While the original version is rather expensive to compute – it is in complexity class $O(n^2)$ – several speed improvements have been implemented since then. We use the optimized FastDTW algorithm as described by Salvador and Chan [SC07].

3.3. Projecting Similarity

The generated distance information is rather extensive and not directly interpretable by the analyst. At this point, dimensionality reduction becomes necessary to be able to convey the information to the user. The user needs to be enabled to detect changes, especially outlier patterns and to find clusters of similar patterns.

Inspired by projection-based approaches such as the MotionExplorer system [BWK*13], we derive a 2D projection of the time series patterns based on the pair-wise distances. The goal here is to preserve the distances from the original data set as good as possible. Patterns that are similar should have 2D positions that are close and patterns that are very different should have a large distance between them. Here, the first part of the statement is more important than the second one. If two very similar patterns are plotted apart, the user gets a wrong impression of the data. On the other side, if two different patterns are far apart, it is not that important how different they are. This allows us to use non-linear projection methods that preserve local structure in favor of global projection quality.

The resulting scatter-plot represents the similarity of the daily patterns. Any kind of projection introduces errors, due to the expected loss of information. After a series of tests, the class consistency measure of Sips et al. turned out to be the most robust quality measure [SNLH09]. For each point, the set of n nearest neighbors in high-dimensional space is compared to the n nearest neighbors in 2D space. The quality is defined by the set of elements that appear in both sets. We use this approach to assure that the projection quality is high enough to allow for drawing reliable conclusions from the data. In practice, stress-based non-linear projection methods such as those from the MDS family perform quite well for many data sets [JPC*11].

4. Visualization & Interaction

We present two tightly coupled views of the linked time-series data to the domain expert. Based on an atomic entity – i.e. daily patterns – all data records are arranged based

on their pair-wise similarity and displayed in the Similarity View (Figure 1, left). This provides an overview of the recorded patterns in the network and allows for different interactions. In a second, geo-based view (Figure 1, right), the same patterns can be analyzed on different levels of granularity in time, but also in space using a visual calendar. The visual link in between is based on the color that is assigned to the different groups of daily patterns. In combination, the two views show the data from two complementary perspectives, both supporting each other.

4.1. Similarity View

The Similarity View gives an overview on all recorded patterns of the entire network. It displays the patterns with respect to their pairwise similarities that were computed in Section 3.3. Every pattern is represented by a single point in the screen space.

By selecting a particular node in the network, the analyst can investigate the change of patterns over time. The view connects all daily patterns of that stations and orders them by time. The result can be seen in Figure 3. All days of the station *Newham* are plotted for the month May. The patterns oscillate at a high frequency up and down with one outlier on the left. In contrast to straight line segments, bézier splines are used to interpolate between the patterns, because the changes are expected to happen gradually.

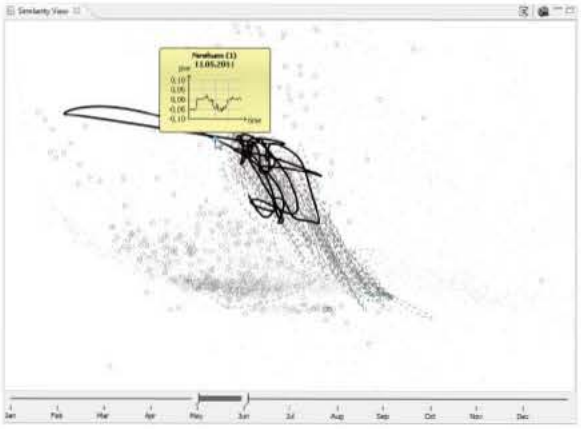

Figure 3: *The sensor at Newham, plotted based on the similarity of daily patterns for the month May.*

Using a range-based slider that spans over the entire year, the user can filter the data set with respect to recording time. Filtered elements are not being hidden to preserve the context, but they are rendered small and their color becomes faint. The filtered part of the spline turns into a thin, gray, dashed line. See Figure 3 for an example.

The user can access the actual shape of the pattern on demand by hovering over any element. A tooltip with additional information about the data point appears, providing details about the sensor, the recording date and a line-chart of the time series of the particular day.

As already mentioned, the axes in this display do not have an intrinsic meaning. The information in the scatter-plot is very fine-grain and does not permit an intuitive understanding of its organization. We create an abstraction layer on top of the scatter-plot display. In other words, we discretize the continuous space into a set of discrete partitions. In this layer, similar patterns are aggregated into larger groups which help the user in getting an overview of available patterns and their location. This is accomplished by clustering of the data and displaying the groups, each annotated with a representative element. Clustering is performed in 2D space, after projection of the data, to prevent clusters from being rendered as fragmented regions. The result of this operation can be seen Figure 4.

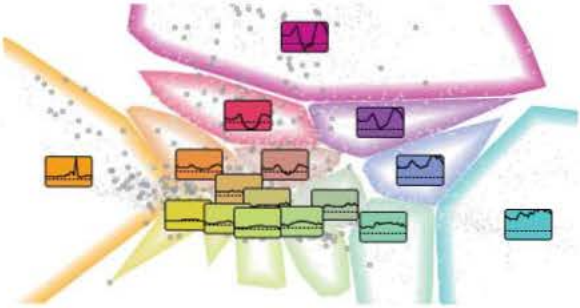

Figure 4: *The projected pattern similarities, clustered and colored using a 2D colormap. Some of the patterns show negative energy consupmtion during daytime (purpelish red patterns at the top). This could indicate that connected solar plants produced significant amounts of energy on that days.*

Generally speaking, the intrinsic property of good clusters is that the elements within have low pair-wise distances while distances to elements in other clusters are comparatively large. If a central element in the cluster can be identified, it would represent the other elements in the cluster with a minimum of lost information. The k-means algorithm creates a clustering based on such cluster representatives. While this algorithm is rather basic, it produces the cluster representatives. This pattern stands for the means of the cluster, i.e. an artificial pattern with the smallest distance to all other pattern in the cluster. In cases where no new element can or should be created, the closely related k-medians algorithm works on existing elements only [JD88]. The common challenge of choosing an optimal number of desired clusters is not a problem in our main use case, because its main purpose is to create a simplified version of the data. The number of clusters is limited by the number of pattern shapes the user is expected to differentiate. In practice, choosing k in the range of 10 to 20 seems to be reasonable.

To improve the readability of the drawing, every cluster

is annotated with a small line chart glyph of the mean pattern that the cluster stands for. This technique is similar to the Micro-Macro Views display [BvLBS09], which uses the rectangular grid of a SOM to derive the 2D position of the entities together with a representative for each of the clusters. In our system, the clustering is separate from the projection, resulting in a non-rectangular layout of the clusters. It is shown as small line chart in the center of every cluster.

Figure 5: *A 2D colormap created by interpolation of four perceptually distant colors. It defines the color of the similarity clusters.*

We emphasize similar patterns using a discrete set of colors to indicate cluster membership. Thus, the color indicates the shape of the pattern without having to show the actual pattern. Patterns of similar color are expected to have a similar shape. Using the 2D position in the similarity plot, the corresponding color of a pattern can be derived from a 2D colormap. This allows us to also use the color as an indicator of similarity. However, using only one color per clusters makes it easier for the user to recognize a certain color as the same in another view if many different slight variations coexist. The colormap must enable intuitive and accurate readings in order to express the metrics of similarity. On the one hand, it should exploit a maximum of different colors. On the other hand, the user must be able to estimate the approximate the distance between two objects correctly, which requires a perceptual uniform interpolation. In contrast to the RGB or the HSV color space, CIELAB is a non-linear colorspace that can be used to extract perceptually uniform 2D planes. However, as presented by Bremm et al. [BvLBS11], these colormaps do not contain many perceptually different colors. Inspired by the work of Ziegler et al. [ZNK07] we use four perceptually distant colors and interpolate between these colors. However, we slightly use a different set of colors, namely yellow, cyan, red and blue. The goal of this selection is to separate the colormap into complementary color tones and also from fully saturated (bottom) to fully intense (top) colors. We use cyan instead of green in order to approximately equalize the perceptually distance between all corner colors. The corner colors are equalized in intensity and saturation in the HSI color space [Kei00] and then interpolated in the CIELAB color space (see Figure 5).

While this view already contains a lot of information on the occurring patterns in the network, the network structure is not visible. Also, it is not immediately clear, which sensor measures which pattern at which time of the year. We overcome these limitations with a second view that displays just that. Tight coupling and interactive linking between the two ensures that the user can bridge the mental gap between two different visual representations of the same entity.

4.2. Network View

This second part of our system has its focus on the network topology. The visualization is a node-link diagram with drill-down functionality that displays temporal information in the node glyphs on demand. In this manner, the user can not only learn about the spatial organization, but also the pattern distribution in different temporal granularities.

Nodes represent sensors and edges denote connections between the sensor. A sound layout should create an intuitive display of the topology, but preserve directions, if possible. The user is interested in an abstraction of local geographic coordinates to reduce the visual complexity of the network. General graph layout algorithms, however, try to satisfy edge length constraints and/or minimize the number of edge crossings. This are typically not problems for sensor networks, as both criteria are not overly important.

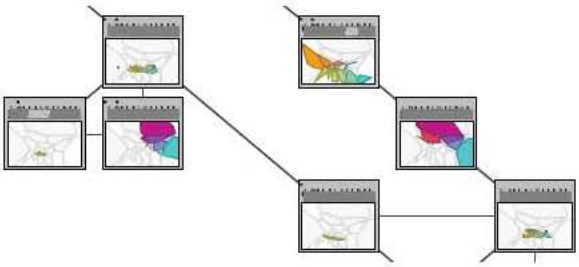

Figure 6: *The network view at the second level of detail. Both calendar and cluster fingerprint view appear.*

A prominent group of methods that achieves this is the octi-linear layout family. These algorithms create a schematic representation that is inspired by the metro map metaphor. Originally, these methods were used to generate layouts of subway lines, which lead to the name *Metro Maps*. They restrict the angles of edges between nodes of the network to multiples of 45 degrees, yielding a stratified version of the original layout. They also try to preserve directions where possible. We adopt one of these algorithms to compute the layout of a sensor network. While different algorithms exist, we chose the work of Nöllenburg et al. [NW06]. In contrast to other works, it favors quality over computation speed. As the layout is static, this can be pre-computed and thus speed is not a major issue.

In order to avoid cognitive overload, this view uses different level of details to adjust the visual complexity. A "virtual camera" that supports zooming and panning enables the

user to navigate in this spatial view. Zooming in and out triggers different levels of detail of the sensor node glyphs. At the most abstract level, all nodes are represented by simple, labeled rectangles (Figure 2). Starting at the second level, higher zoom levels show two small views: the calender and the cluster fingerprint (Figure 6). Every zoom level scales the views in intervals, because only discrete scale factors make sense for the contained calender view.

Focusing on a particular station, the user is interested in its behavior over time. The patterns in the contained time series can be analyzed with respect to different occurrence frequencies. Using van Wijk's Calendar View [VWVS99], we assign a single color value per day based on the cluster the pattern belongs to. This color is then used to colorize a calendar (Figure 7). In this manner, the user can thus identify seasonal patterns. In contrast to radial plots, the calendar view assigns the same amount of screen space to individual patterns, which gives them equal visual importance.

Figure 7: *Calendar view of a partly selected sensor. The calendar maps colored patterns to a cluster. A selection is active which causes unselected elements (mostly in summer) to become smaller.*

Every day in the calendar is colored by the cluster color this day belongs to. While different layouts for calendars exist, we decided to align weekdays on along horizontal axis. Weekdays are ordered according to the international standard ISO 8601 which defines Monday as the first day of the week. This alignment brings Saturday and Sunday together, which facilitates the distinction between workdays and weekends. From left to right, weekly patterns and changes over the year for a given weekday become apparent. From top to bottom, patterns within a week are visible. Looking at a distance on the small calendar, larger seasonal changes are most recognizable. A tooltip shows the actual pattern together with the date and the ID of the cluster. Using cluster IDs serves as an alternative to matching the color across different views, especially for color-deficient people.

The analyst also wants to know which patterns are specific to a particular sensor. We therefore added a small filtered version of the similarity view. All clusters are displayed in light gray to provide context to the current focus (the sensor). Then, a filtered set of clusters that contain only patterns from this sensor is created. In a Focus & Context approach, these reduced clusters are then drawn on top of the faint, unfiltered clusters. Patterns that were recorded by the sensor in focus are highlighted using the same set of colors. This creates a visual fingerprint of the sensor that also has its representation in the similarity view. As in the similarity view, the clusters are drawn using their convex hull, similar to the work of Schreck and Panse [SP07]. Again, tooltips indicate the cluster ID to differentiate borderline cases. See Figure 8 for an example.

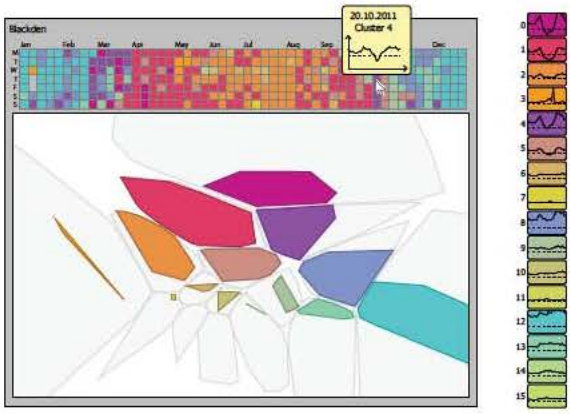

Figure 8: *A sensor shown at the highest level of detail. The calendar maps time to a cluster of patterns. The fingerprint view below illustrates which patterns this sensor recorded compared to the other sensors. Low and even negative consumption patterns are recorded from March to October.*

The system also shows a legend on the right side of the view to facilitate the matching between pattern and color. It is based on the representative pattern of the cluster and the color that is derived from its location in the colormap. It enables the user to see which color relates to which pattern. Again, corresponding IDs are displayed to differentiate borderline cases. The displayed glyph contains the representative pattern of the cluster which is also used in the similarity view. This strengthens the link between the two views.

4.3. Linking the two views

Aside from the visual linking between the two views, interaction with one of them can also affect the other. Selecting a sensor in the network view triggers the selection of all linked time series patterns in the similarity view. Using a single selection color to highlight a selected element would overwrite the cluster association of the elements. We therefore use the color of the corresponding cluster to highlight selected patterns and display the remaining ones in gray.

On the opposite side, we can also select interesting patterns in the similarity view and see their distribution in the network. We use a lasso selection tool that is known from image manipulation software to maximize flexibility. Again, selected patterns are colored while unselected patterns remain gray.

On the lowest level of detail, the network view shows the distribution of selected patterns across the network. We use a progress bar metaphor (blue bar on bright background) to

reflect the fraction of patterns that were selected. As can be seen in Figure 9, the selection affects mostly the sensors at *Hartham* and *Harwick*. About two thirds of the sensor at *Hartham* are selected.

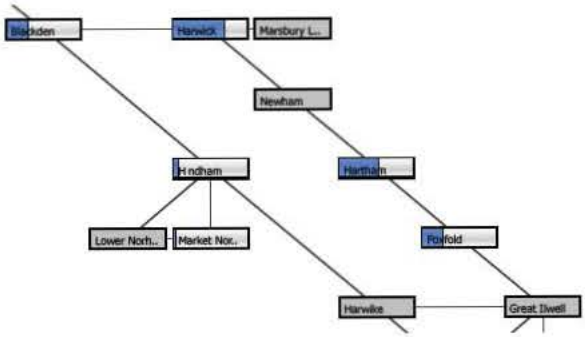

Figure 9: *Selected patterns distributed to their related sensors in the network. Sensors that are at least partly selected are accordingly marked with bluish selection bars on bright background.*

The analyst can also zoom in to also see the selection distributed to individual days. As can be seen in Figure 10, mostly Saturdays and Sundays are selected. In this example, only patterns on the left part of view have been selected. Thus, selected patterns are in different variations of orange. The cyan clusters are not part of the selection and do not appear in the calendar view. Filtered patterns are drawn as miniaturized rectangles to indicate that they are not part of the selection. Missing values are not drawn. The fingerprint view is not affected by the selection.

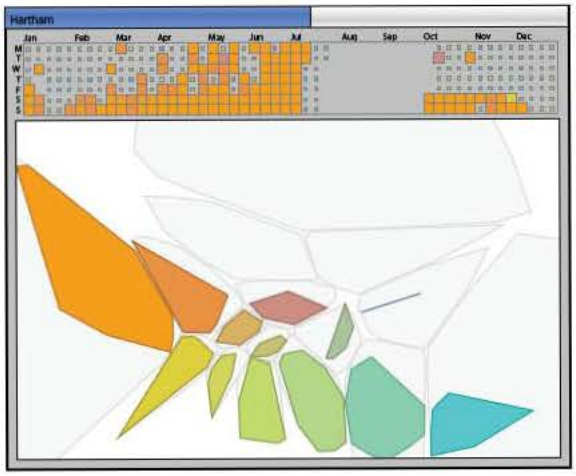

Figure 10: *Partly selected sensor at Hartham. Mostly weekends are affected by the selection. Filtered patterns are displayed only as small rectangles. The months August and September do not contain any data.*

5. Case Study

We performed a guided case study with a domain expert to demonstrate the usability of our approach in a real-world use case. The expert identified two major areas of relevance: monitoring and planning. The first step was to identify interesting patterns with the help of the legend of the network view (Figure 8). In the legend, the pattern that occurred most frequently gave the expert a quick overview on the network. An interesting finding was the prominence of patterns with backflow (i.e. patterns with a significant values below zero) during daytime which is unusual. These patterns indicate an electric flow from the consumers back into the grid – an often undesired result which is due to the high amount of solar panels in the pilot region where the data was recorded.

In the next step, the grid was explored using the Network View. The domain expert first focused on the calendar view, because it was considered the most intuitive one and most similar to the tools the expert uses. Typically, manual lookup of patterns from the previous years is required to derive typical daily patterns, based on the day of week, season of year and other circumstances (e.g. public holidays). For the monitoring task, the focus was on some of the previously identified patterns (see Figure 8). After that, the Similarity View was used to select the interesting parts (patterns with backflow) in the top-left corner using the lasso tool. This selection action highlighted in the Network View that most of the patterns were recorded at only 5-6 stations in the network (e.g. in Figure 9). The expert concluded that only these few stations needed to be investigated further in terms of backflow protection. For the planning task, the interest was on finding the right time to temporarily isolate stations or cables for maintenance. This should be done when power flow is at the lowest for all relevant stations. The expert therefore used the network overview to anticipate the pattern for different station on a given day based on the recordings of the previous year.

6. Design Process

In order to optimize the design choices, we performed the design process in an iterative manner. Different data mappings, visual representations and interactions were explained to a group of 8 non-expert users and two experts from the electrical grid domain. We conducted informal interviews with the running prototype which led to fruitful discussions about the pros and cons of different aspects of the system. In a final round, we gave a video demonstration to two usability professionals to get feedback on the usability of the system.

The first idea was to create a geo-referenced layout that is drawn on top of a thematic or navigational map. A result from the interview with the experts was that geographic reference is required only in exceptional cases. The most important design factors for them were the network topology, followed by simplicity.

The visual representation of the sensor node has changed significantly through the design phase. One idea was to split the calendar into four distinct seasons. The fingerprint view was motivated by the fact that users could not correlate the two views without explanation. Putting the calendar above the fingerprint was motivated by the fact that it was unclear to some of the users where the calendar legend belongs to if put the other way round.

Different concepts to connect the similarity trajectory of a single sensor were proposed. The idea of drawing arrow heads to indicate the direction of the spline was rejected, because the glyphs were often misinterpreted in crowded displays. Aside from line segments and splines, convex hulls [SP07] and bubble sets [CPC09] were evaluated. While they two emphasize areas, they also cover much screen space, especially when outliers are present. Also, the temporal sequence was no longer visible. Combing multiple techniques seemed promising at first, but produced too much overplotting. We conducted a survey with about 15 non-experts with 12 screenshots of the system, each with a different colormap. It clearly confirmed that the four colors we used achieved the clearest color separation.

Using integer IDs for clusters was suggested by one of the users to enforce the ability to recognize the same cluster in different representations, especially for color deficient people. The ID is used in the legend and in the tooltips of the calendar view, the fingerprint view and the similarity view.

7. Discussion & Outlook

In this paper we presented a visualization system for interactive pattern analysis in univariate sensor networks. The focus is on the analysis of similar patterns over different temporal scales, but it also respects the network structure of the sensors. It consists of two strongly linked views that enable the analyst to gain insight into the data set. The cluster prototypes show typical, often occurring patterns. The network view gives an overview over the network topology and the patterns for each sensor. This enables the analyst to compare different sensors and to see seasonal trends.

We considered two types of scalability: the number of stations and the length of measurement data. The application is fairly robust with respect to the number of nodes. The similarity view is not affected by the network complexity and the network view uses a drill-down metaphor to adjust the amount to displayed information. For very large networks, aggregation based on either topology or geography could be used. Currently, only one year of measurements can be analyzed. Comparing yearly patterns requires a different visual encoding of the data.

Future work includes the extension to multi-variate data sets. A challenge is to integrate both multi-variate data and the time domain in one similarity model. Also, an appropriate visual representation of the data is required, in particular for the glyph that represents clusters of similar elements.

We also plan to include semi-interactive clustering to better adjust the representation patterns. This could be achieved by iterative split and merge operations or through adjustment of the distance functions and the clustering parameters. Another aspect is to extend the system to also support live monitoring. As of now, the major restriction is the insertion of new measures in already existing projections. One possible solution would be to insert new measures in an existing projection. This, however, requires a deterministic projection function and the quality may decrease as the projection is computed based on the old patterns only. Another option would be to recompute the projection every time a new pattern is added. While this is a sound approach in theory, it will be challenging to communicate the changes between the old and the new projection effectively to the user, especially if it changes significantly. One lesson we learned in this project is the role of familiar visualizations to enable the learning of new techniques. Here, the calendar view was the anchoring point for the user to understand the rest of the system featuring visualizations which have not been used before.

Acknowledgements

This work has been conducted in the context of the project VASA (grant number 13N11254) funded by the German Federal Ministry of Education and Research (BMBF). We would like to thank the experts at EnBW Regional AG and Siemens AG for their support and constructive feedback. We also thank the Algorithmics Group at the University of Konstanz for providing the MDSJ library.

References

[AAD*10] ANDRIENKO G., ANDRIENKO N., DEMSAR U., DRANSCH D., DYKES J., FABRIKANT S. I., JERN M., KRAAK M.-J., SCHUMANN H., TOMINSKI C.: Space, Time and Visual Analytics. *Int. J. Geogr. Inf. Sci. 24*, 10 (Oct. 2010), 1577–1600. doi:10.1080/13658816.2010.508043. 3

[AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of Time-Oriented Data.* Springer, London, UK, 2011. doi:10.1007/978-0-85729-079-3. 3

[BAF*13] BÖGL M., AIGNER W., FILZMOSER P., LAMMARSCH T., MIKSCH S., RIND A.: Visual Analytics for Model Selection in Time Series Analysis. *IEEE Trans. Vis. Comput. Graph. 19*, 12 (2013), 2237–2246. 3

[BC94] BERNDT D. J., CLIFFORD J.: Using Dynamic Time Warping to Find Patterns in Time Series. In *KDD workshop* (1994), vol. 10/16, Seattle, WA, pp. 359–370. 4

[BP09] BRANDES U., PICH C.: An Experimental Study on Distance-Based Graph Drawing. In *Graph Drawing*, vol. 5417 of *Lecture Notes in Computer Science*. Springer, 2009, pp. 218–229. doi:10.1007/978-3-642-00219-9_21. 3

[BRG*12] BERNARD J., RUPPERT T., GOROLL O., MAY T., KOHLHAMMER J.: Visual-Interactive Preprocessing of Time Series Data. In *SIGRAD* (2012), pp. 39–48. 3

[BRS*12] BERNARD J., RUPPERT T., SCHERER M., KOHLHAMMER J., SCHRECK T.: Content-based Layouts for Exploratory Metadata Search in Scientific Research Data. In *Proc. of the Joint Conference on Digital Libraries* (2012), JCDL, ACM, pp. 139–148. doi:10.1145/2232817.2232844. 3

[BTK11] BERTINI E., TATU A., KEIM D. A.: Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization. *IEEE Symposium on Information Visualization (InfoVis) 17*, 12 (Dec. 2011), pages 2203–2212. 3

[BvLBS09] BERNARD J., VON LANDESBERGER T., BREMM S., SCHRECK T.: Micro-Macro Views for Visual Trajectory Cluster Analysis. In *IEEE Symposium on Visualization* (2009). 6

[BvLBS11] BREMM S., VON LANDESBERGER T., BERNARD J., SCHRECK T.: Assisted descriptor selection based on visual comparative data analysis. In *Computer Graphics Forum* (2011), vol. 30/3, Wiley Online Library, pp. 891–900. 6

[BWK*13] BERNARD J., WILHELM N., KRÜGER B., MAY T., SCHRECK T., KOHLHAMMER J.: MotionExplorer: Exploratory Search in Human Motion Capture Data Based on Hierarchical Aggregation. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2257–2266. 4

[BWS*12] BERNARD J., WILHELM N., SCHERER M., MAY T., SCHRECK T.: TimeSeriesPaths: Projection-Based Explorative Analysis of Multivarate Time Series Data. *Journal of WSCG 20*, 2 (2012), 97–106. 3

[CPC09] COLLINS C., PENN G., CARPENDALE S.: Bubble sets: Revealing set relations with isocontours over existing visualizations. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 1009–1016. 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. *Proc. VLDB Endow. 1*, 2 (Aug. 2008), 1542–1552. 3

[Fu11] FU T.-C.: A Review on Time Series Data Mining. *Eng. Appl. Artif. Intell. 24*, 1 (Feb. 2011), 164–181. doi:10.1016/j.engappai.2010.09.007. 3

[HR02] HINTON G. E., ROWEIS S. T.: Stochastic neighbor embedding. In *Advances in neural information processing systems* (2002), pp. 833–840. 3

[HSCW13] HADLAK S., SCHUMANN H., CAP C. H., WOLLENBERG T.: Supporting the Visual Analysis of Dynamic Networks by Clustering associated Temporal Attributes. *Visualization and Computer Graphics, IEEE Transactions on 19*, 12 (2013), 2267–2276. doi:10.1109/TVCG.2013.198. 3, 4

[JD88] JAIN A. K., DUBES R. C.: *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. 5

[JPC*11] JOIA P., PAULOVICH F. V., COIMBRA D., CUMINATO J. A., NONATO L. G.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2563–2571. 3, 4

[JSMK14] JANETZKO H., STOFFEL F., MITTELSTÄDT S., KEIM D. A.: Anomaly Detection for Visual Analytics of Power Consumption Data. *Computer & Graphics 38* (2014), 27–37. 3

[Kei00] KEIM D.: Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics 6*, 1 (2000). 6

[Kru64] KRUSKAL J.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27. doi:10.1007/BF02289565. 3

[LV10] LEE J. A., VERLEYSEN M.: Unsupervised dimensionality reduction: Overview and recent advances. In *The International Joint Conference on Neural Networks* (2010), pp. 1–8. 3

[NW06] NÖLLENBURG M., WOLFF A.: A Mixed-Integer Program for Drawing High-Quality Metro Maps. In *Graph Drawing*, vol. 3843 of *Lecture Notes in Computer Science*. Springer, 2006, pp. 321–333. doi:10.1007/11618058_29. 4, 6

[SC07] SALVADOR S., CHAN P.: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis 11*, 5 (2007), 561–580. 4

[SFK13] STOFFEL F., FISCHER F., KEIM D. A.: Finding anomalies in time-series using visual correlation for interactive root cause analysis. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security* (New York, USA, 2013), VizSec '13, ACM, pp. 65–72. doi:10.1145/2517957.2517966. 3

[SLH*11] SHI L., LIAO Q., HE Y., LI R., STRIEGEL A., SU Z.: SAVE: Sensor anomaly visualization engine. In *IEEE Conference on Visual Analytics Science and Technology (VAST)* (2011), pp. 201–210. doi:10.1109/VAST.2011.6102458. 3

[SLN05] SARAIYA P., LEE P., NORTH C.: Visualization of graphs with associated timeseries data. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on* (2005), pp. 225–232. doi:10.1109/INFVIS.2005.1532151. 3

[SNLH09] SIPS M., NEUBERT B., LEWIS J. P., HANRAHAN P.: Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum 28*, 3 (2009), 831–838. doi:10.1111/j.1467-8659.2009.01467.x. 3, 4

[SP07] SCHRECK T., PANSE C.: A new metaphor for projection-based visual analysis and data exploration. *Proc. SPIE 6495* (2007). doi:10.1117/12.697879. 7, 9

[Tor52] TORGERSON W. S.: Multidimensional scaling: I. Theory and Method. *Psychometrika 17*, 4 (1952), 401–419. doi:10.1007/BF02288916. 3

[TS08] TOMINSKI C., SCHUMANN H.: Enhanced Interactive Spiral Display. In *Proceedings of the Annual SIGRAD Conference, Special Theme: Interactivity* (2008), Linköping University Electronic Press, pp. 53–56. 3

[VWVS99] VAN WIJK J. J., VAN SELOW E. R.: Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of the Symposium on Information Visualization* (Washington, DC, USA, 1999), INFOVIS, IEEE Computer Society, pp. 4–9. 3, 7

[WG11] WARD M. O., GUO Z.: Visual Exploration of Time-series Data with Shape Space Projections. In *Proceedings of the Eurographics Conference on Visualization* (2011), EuroVis, Eurographics Association, pp. 701–710. doi:10.1111/j.1467-8659.2011.01919.x. 3

[WSM*09] WONG P. C., SCHNEIDER K., MACKEY P., FOOTE H., CHIN G., GUTTROMSON R., THOMAS J.: A Novel Visualization Technique for Electric Power Grid Analytics. *IEEE Transactions on Visualization and Computer Graphics 15*, 3 (2009), 410–423. doi:10.1109/TVCG.2008.197. 3

[ZFH08] ZHAO J., FORER P., HARVEY A. S.: Activities, ringmaps and geovisualization of large human movement fields. *Information Visualization 7*, 3-4 (2008), 198–209. 3

[ZNK07] ZIEGLER H., NIETZSCHMANN T., KEIM D.: Visual Exploration and Discovery of Atypical Behavior in Financial Time Series Data using Two-Dimensional Colormaps. In *Information Visualization, 2007. IV '07. 11th International Conference* (2007), pp. 308–315. doi:10.1109/IV.2007.124. 6