Enhancements for Visualizing Temporal and Geospatial Datasets

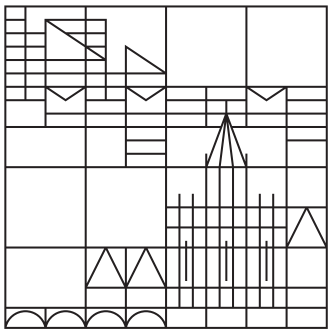
Dissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften

vorgelegt von

Halldór Janetzko

an der

Universität Konstanz



Mathematisch-Naturwissenschaftliche Sektion

Informatik und Informationswissenschaft

Tag der mündlichen Prüfung: 17. Juli 2015

Referenten: **Prof. Dr. Daniel A. Keim, Universität Konstanz**

Prof. Dr. Oliver Deussen, Universität Konstanz

© 2015 - Halldór Janetzko
Some rights reserved.
This work is licensed under a
Creative Commons Attribution-NonCommercial 3.0 license.
http://creativecommons.org/licenses/by-nc/3.0/

Included or cited figures of published works are not touched by this copyright disclaimer and are under copyright of the respective publisher.

Enhancements for Visualizing Temporal and Geospatial Datasets

Abstract

In this thesis, we will discuss enhancements for the analysis and visualization of temporal and geospatial data. Techniques for both research domains have a long history and wide applicability, but at the same time suffer from basic issues as overplotting or hidden patterns. In combination, space and time are even more challenging with respect to the Visual Analytics design however enable new perspectives. The main idea of all enhancements presented in this thesis is focusing on interesting aspects of the data and visually conveying concepts by abstraction. The importance is in our case defined by subject matter experts and consequently our methods are parametrized in a way allowing user influence. In detail, we will improve analysis, prediction, and visualization techniques for time series by mechanisms enhancing the visual saliency of important points in time. Additionally, our goal is to implement inspectable models and explain why our system believes something being important for the analyst. As a second step, we investigate how to enhance geospatial visualizations avoiding and reducing overplotting issues. Overplotting often occurs in geospatial visualizations because of unequal density distributions. We discuss techniques to reduce overplotting in point-based visualizations and present simplifying methods for line-based representations, as in general removing all overplotting lines is not possible. Combinations of both geospatial and temporal data are analyzed in the domain of recorded soccer data. We enhance the way domain experts analyze soccer matches and present methods enabling the expert to focus only on the interesting parts of a match by appropriate Visual Analytics techniques.

Enhancements for Visualizing Temporal and Geospatial Datasets

Zusammenfassung

In dieser Arbeit werden Verbesserungen für die Analyse und Visualisierung zeitlicher und räumlicher Daten vorgestellt. Techniken aus beiden Forschungsbereichen besitzen nicht nur eine lange Entwicklungsgeschichte und ein breites Anwendungssprektum, sondern auch grundlegende Probleme wie die Überdeckung von Datenpunkten oder nicht sichtbare aber gleichzeitig relevante Datenverteilungen. Die Kombination von Raum und Zeit in Form der Bewegungsanalyse ist besonders herausfordernd, aber bietet gleichzeitig auch neue Perspektiven. Der abstrakte, gemeinsame Nenner der in dieser Arbeit vorgestellten Verbesserungen ist die Fokussierung auf interessante Datenaspekte und die visuelle Abstraktion von den ursprünglichen Daten. Das Interessantheitsmaß wird in unserem Fall von Domänenexperten definiert. Folglich sind die vorgestellten Verfahren parametrisiert und durch den Analysten beeinflussbar. Im Einzelnen werden für Zeitreihen sowohl Analyse-, Vorhersage- als auch Visualisierungsmethoden verbessert, indem wichtige Zeitpunkte berechnet und visuell hervorgehoben werden. Zusätzlich zielen unsere vorgestellten Verfahren darauf ab, vom Analysten nachvollziehbar zu sein und zu erklären, warum unser System eine Situation für interessant hält. Als nächstes untersuchen wir, wie im Bereich der Visualisierung räumlicher Daten Überdeckungsprobleme gelöst werden können. Überdeckungen treten häufig in räumlichen Visualisierungen aufgrund ungleicher Dichteverteilungen auf. Wir behandeln in dieser Arbeit sowohl punkt- als auch linienbasierte Visualisierungen räumlicher Daten, wobei die Überdeckung von Linien im Allgemeinen nicht vollständig gelöst werden kann. Kombinationen von räumlichen und zeitlichen Daten werden im Bereich aufgezeichneter Fußballspiele analysiert. Es werden Verbesserungen vorgestellt, welche die Arbeit eines Fußballanalysten erleichtern und es ihm ermöglichen, sich nur auf relevante Aspekte des Fußballspiels zu konzentrieren.

Contents

List of Figures

If you do not know how to ask the right question, you discover nothing.

William Edwards Deming

1 Introduction

Asking the right question is challenging or even impossible without prior knowledge. The essential question is how to acquire prior knowledge in order to ask the right questions? When we start analyzing data we have no prior knowledge nor hypotheses, we usually start with a method called "Explorative Analytics" investigating the data space. We develop rudimentary visualizations and run first statistical and correlation analyses. In an iterative process, we will derive new hypotheses and refine our visualization and analysis techniques. The focus of this process is to enable the analyst asking the right questions.

Explorative Analytics is technically sound and works in practice, nevertheless there are challenges to tackle. Often, first visualization approaches are not perfectly suited to the data types and distributions. As a result, the available screen space is not optimally used and important patterns may be hidden. Implementing visual analysis techniques being robust to unknown data distributions and supporting analysts in gathering first hypothesis and findings is crucial. Filtering, selecting, and visually highlighting manually selected or semi-automatically derived interesting patterns should be enabled as well. In this thesis, we will discuss Visual Analytics

techniques for movement data enhancing the visibility of patterns and dealing with overplotting. Movement can be seen as a combination of two orthogonal dimensions: time and space. Both domains taken alone are already research-wise challenging and many techniques were developed dealing with only one of the two dimensions. We focused in our research on temporal and geospatial data sets and furthermore on their combination in the form of movement data.

1.1 Motivation

The temporal dimension and our perception of time is very fascinating. From our human perspective, time is partitioned into past (our knowledge and experiences), present (our current mood, situation, and sensory input), and future (our plans and next steps). Compared to the infinite amount of past and future, the present we are experiencing and living in is an infinitely small amount of time. Everything we realize and process in our neurons is actually a snapshot of the past. This directly influences how we can cope and interact with temporal data visualizations. Seeing temporal correlations in still data visualizations is not preattentively possible. In animations, for us it is only naturally to see for example correlated movement behavior. But animations do not help humans when remembering single scenes is of importance. Still images (photographies) and animations (videos) have their right to exist as both can convey different kinds of information. Bridging the gap between images and animation is important but unfortunately not trivial at all. Techniques for still images like Small Multiples and Brushing & Linking were developed to connect both worlds to some extent. From the animation side, helping the analyst to guide his attention to important time points and not watching the whole time frame over and over again can be achieved by semantically meaningful keyframe extraction or adaption of the animation speed. The ultimate goal of temporal visualizations is to explain time-dependent behavior and correlations, to support the analyst understanding the current situation, and to enable the analyst in drawing conclusions and actions for future planning. Our research goal is to enable the analyst in assessing the important situations with techniques going further than pure playback techniques by semantically meaningful highlighting. An awareness for such important situations in the past will support the domain expert in defining his next actions.

In the geospatial domain, the very first comprehensive lesson to be learned is that "spatial is special". The long history of visual representations for geospatial data already gives some hints why spatial data are special. Prehistoric signs for maps can be found in cave paintings

and rock carvings depicting significant landscape features as rivers or hills. As there are many artifacts that may show a map-like representation it is not completely clear, when the first map was painted. However, there are two prehistoric maps dating from 25,000 BC (Pavlov map) and 11,000 BC (Mezherich map) being not very geometrically accurate. The first maps rather revealed concepts of how the world was seen and experienced. Usually, historic maps were restricted to the local neighborhood and drawn from a very egocentric perspective. One of the first maps depicting topology on a global scale is the World Map of Babylon (600 BC) representing the Earth by two concentric circles with Babylon being in the center. Increasing trading and the foundation of trading centers increased the need for accurate, geometry-based maps. Cartography and exploration of unknown regions had a high priority during the European Renaissance and research expeditions were quite common and built the basis of our maps today. Obviously, the empty spots on maps have been filled today and Google Earth for example stores 70.5 Terabytes of topological data and aerial images. Today, we are used to the ubiquitously available bird's eye view of the world. In computer science, we can easily employ a two-dimensional representation of geospatial data as reading maps is a skill we learn during childhood. The science of designing, drawing, and beautifying maps is quite advanced, as it is impossible to imagine our everyday lives without maps. When visualizing geospatial data sets and mapping the visual variables to the data space, we will often have to use the variable position to encode the geospatial location. This often limits our design space and using position for geospatial coordinates will often result in overplotting because of dense regions. Our research aim is to convey information of complex and dense spatial data with large amounts of overplotting to the analyst. We inverse the historic evolution of maps and present not the original spatial data but rather conceptualized spatial patterns.

Movement analysis combines both domains the geospatial and the temporal domain. A very famous example for the visual depiction of movement and temporal developments is the map of Napoleon's Russian campaign of 1812 painted by Charles Minard in 1869. This map is an extraordinary case, where spatial and temporal data are conveyed comprehensibly in one single visualization. However, the visualized geospatial pattern is a back-and-forth movement parallel to the x-axis simplifying the visual design. Dealing with arbitrary movement data typically combines not only the geospatial and the temporal domain but also the challenges of both domains. For instance, overplotting resulting from the geospatial domain will propagate to systems dealing with movement data. Furthermore, watching animations of all recorded movements is not efficient for analysis purposes. The challenge in movement data lies in the design of an analysis

system supporting effective and efficient analyses, visualizations, and interactions. However, we can to some extent apply techniques developed for the single domains and connect them in an semantically meaningful way. We tackle the research questions which techniques to combine meaningfully in the domain of movement analysis for soccer games enabling subject matter experts in revealing interesting patterns and findings.

We will focus in this thesis on enhancements for visualizations enabling the analyst in finding, understanding, and interpreting patterns. Our goal is to reduce the effort for detecting patterns by increasing the visual salience of interesting situations and by reducing artifacts in existing visualization and analysis techniques.

1.2 Thesis Structure

The content of this thesis can be seen two-fold: there are sections introducing and describing novel techniques and there are sections combining existing and in this thesis proposed techniques in application-driven Visual Analytics systems. More in detail, we will discuss in Chapter 2 enhancements for visualizations in the temporal domain. We start with general visual boosting techniques, discuss a user-controlled peak-preserving prediction method, and combine those approaches in a Visual Analytics system for investigating power consumption data. In Chapter 3, we will present enhancements for geospatial visualizations. We will discuss an overplotting-free visualization of point data and furthermore simplification and abstraction techniques for lines. The subsequent Chapter 4 deals with Visual Analytics for soccer data and combines temporal and geospatial aspects and techniques. Lastly, we will conclude this thesis and give an outlook to future work in Chapter 5.

1.3 Contributions

The contributions presented in this thesis are mostly enhancing existing visualization techniques and showing their applicability to real-world application scenarios. The enhancements discussed here were usually researched with a specific application need in mind resulting from contacts to subject matter experts. The following list gives for each section an overview of the contributions claimed by this thesis:

- Section 2.1: Description and comparison of state-of-the-art boosting techniques to increase the visual salience of data items

- Section 2.2: Research and evaluation of an peak-preserving, interactive prediction technique
- Section 2.3: Automatic detection of anomalies and presentation of a visual analysis system for hierarchical power consumption time series
- Section 3.1: Discussion of an overplotting-free, enhanced scatter plot based on local correlation patterns
- Section 3.2: Proposing simplifications and enhancements for geospatial data represented as linear segments
- Chapter 4: Discussion of methods suitable for soccer analysis enhancing understanding and visual salience of interesting aspects of a match

1.4 Citation rules

Most techniques described in this thesis are already published in a conference or journal. In order to avoid any suspicion about plagiarism and self-plagiarism, I try to be as transparent as possible concerning the origin of sections. This resulting thesis is a trade-off between a nicely readable thesis (rewriting of all my peer-reviewed articles) and a thesis following the strictest citation rules (quoting all sections being related to a publication). I decided to focus on the content, contributions, and the reader, as I believe these to be most important. For transparency reasons, I will state at the beginning of each section from which publication the content is taken from. In this thesis, I follow the subsequent citation rules:

- For each cited own publication, I list the contributions of all authors in a footnote.
- I differentiate between three different kinds of integrating already published works into this thesis:
  - Quoted paragraphs are not written by myself and contain contributions of co-authors.
  - Sections "taken from" my publications are copied and differ only in slight wording changes. These sections contain my own contributions and I did all writing myself or rephrased the sections during the paper writing process.

– Sections "based on" a publication are mostly rephrased and the content has been modified. These sections contain my own contributions, but had to be changed to fit nicely into this thesis.

The distinction between the past, present, and future is only a stubbornly persistent illusion.

Albert Einstein

2

Enhancing Visualizations for Temporal Data

Contents

THE TEMPORAL DIMENSION is probably the most influencing dimension to our lives. But time is very special with its own characteristics, when compared to the three spatial dimensions we are surrounded with. We cannot influence the current time point we are experiencing and are not able to jump back-and-forth in time. The only possibilities we have are to experience the present or to wait until the future happens. Beside this unidirectional property, time is a hierarchical dimension. The temporal dimension can be for instance partitioned into spans of seconds, minutes, hours, days, weeks, months, quarters, years, decades, and centuries. The hierarchical nature allows analysts to perform nearly arbitrary temporal aggregations. It is possible for instance to compare the sales development of different quarters or predict the hourly power consumption of a city. Often when dealing with temporal data, we make use of the temporal hierarchy. Space-efficient pixel-based visualization techniques, for example Recursive Patterns [KAK95], employ hierarchical layout-nesting for temporal data.

Humans try to learn from past and historic events and experiences. Important events were passed on and conserved in drawn or written form since dawn of mankind. Nowadays, time-dependent variables are typically measured and stored by computers. In science, the analysis

of time-dependent data plays a very important role. Consequently, a whole research field in the area of analyzing and visualizing temporal data has been established over the last decades. An overview to state-of-the-art visualization and analysis techniques for time series data can be found in the book "Visualization of Time-Oriented Data" by Aigner et al. [AMST11]. The most common analysis tasks for temporal data are listed and described subsequently.

Explorative Analysis

When analyzing previously unknown data without any knowledge about trends or patterns, analyses are typically more of explorative nature. Pure information visualization techniques are a good starting point enabling analyses without prior knowledge. Statistics may help to get more hints to data distribution and patterns. Exploration phases are strongly related to hypotheses generation and quick hypothesis validation or falsification.

Similarity Queries

As soon as the analyst identified a certain temporal pattern, he may be interested in re-occurrences of this specific pattern. An analysis framework should query the time series for the desired pattern and show all time frames with similar temporal behavior. The similarity measure can be freely chosen and depends on the application needs. Another variant of similarity queries are correlation queries. Correlation queries are usually applied to a set of time series. The analyst selects both a time frame and a time series and the system will return all other time series being highly correlated to the selected temporal behavior.

Classification

In case of the classification task, the prerequisite is an annotated training data set. In most times, human analysts will annotate a data sample and provide the enriched set to the classification algorithm. The classifier will assign class labels to the unlabeled input times series based on the training set. Classifications can be performed on time series as a whole and furthermore within time series. Whenever applying data mining algorithms within one time series, partitioning the time series in proper time windows is crucial. Note that similarity queries can be seen as a special case of the classification tasks with only two classes, e.g., *similar* and *dissimilar* to the query pattern, and only one training data sample.

Clustering

Clustering is useful to determine all sets of similar behaving time series. An example for such a set could be working hours dependent time series or constant time series. Clustering algorithms are highly influenced by the similarity measure, which can be exchanged easily. As choosing the proper distance function depends on the application scenario, the cluster quality will vary with different distance functions. If clustering is not used between different time series but applied to a single time series, the system will look for often occurring patterns within the time series. The basic assumption is that there exists a certain amount of repetitive patterns. The clustering technique will identify them and return all occurrences. These repeating patterns are sometimes also called events or motifs.

Regression

Pure statistical approaches as regression are beneficial when the statistical model describing the time series is known. Consequently, regressions are often applied after explorative analyses validating human hypotheses about the data distribution. The parameters of the model are fitted to the actual time series minimizing the residuals. Regression is often used in time series analysis in combination with prediction, where prediction models are fitted as good as possible to the observed time series.

Prediction

Prediction is very related to regression, as the first step is to fit the parameters of a prediction model to the recorded time series. There are basically two kinds of time series, namely periodic and non-periodic time series, influencing set of applicable prediction models. After adjusting the parameters, the model is used to predict the next values of the time series. Based on the residuals during the model fitting process varies the accuracy and uncertainty of the prediction. Overall the statement holds true that the less expectable the time series behaves, the less accurate the prediction will be.

Anomaly Detection

The last technique in this enumeration needs some knowledge and understanding of the time series and furthermore regression and prediction analyses. Visualizing the detected anomalies

will guide the analyst to important time points with unusual data values. The residuals of a regression can be a hint to anomalies, though they are highly dependent on how well the model describes the time series. Prediction methods can be used for anomaly detection when computing the difference between actual and predicted values.

We will investigate in the following sections several analysis and visualization tasks in the domain of time series. We will mostly focus on explorative analysis, prediction, and anomaly detection. However, we will also apply the other techniques for further analysis steps. Visual boosting of data items in pixel displays is discussed in the first section. The proposed boosting techniques are of special interest when emphasizing data points in the visualization. In Section 2.2, we will present a peak-preserving prediction technique with interaction capabilities to steer the prediction process. The third section will combine boosting methods with the prediction technique introduced previously in order to support anomaly-driven Visual Analytics of time series.

2.1 Visual Boosting

This section is based on the following publication[1]:

Visual Boosting in Pixel-based Visualizations

D. Oelke, H. Janetzko, S. Simon, K. Neuhaus, D. A. Keim.

Computer Graphics Forum, Vol. 30, Iss. 3, pp. 871–880, 2011. [OJS+11]

2.1.1 Preface

Time series are a very prominent example for long data sets, resulting in the need of dense display visualizations. Pixel visualizations have been developed to support the visualization

[1]Daniela Oelke had the idea to publish a paper about available boosting techniques and provide a guide when to use which technique. Svenja Simon suggested the distinction between image-driven and data-driven boosting for the comparison of boosting techniques. Daniela Oelke focused on the text application scenario, Svenja Simon described a biological usage scenario, and I discussed a geospatial use case. We all together collected the list of possible boosting techniques and discussed in which usage context they work best. Klaus Neuhaus and Daniel Keim helped with fruitful discussions and advices.

of many data points on one single display [Kei00, KAK95, KSS07, LGP+07]. Pixel visualizations are not only capable of displaying time series, but have been applied in many other domains, like document analysis [KO07], geography [PSKN06], or network and sensor analysis [RG10, FN05]. We define pixel visualizations as techniques using small, colored display areas to represent data values. In our context, these areas are allowed to be larger than only one pixel. Typically, pixel visualizations use position and coloring as their main visual variables. Depending on the size of the pixels other visual variables, such as texture or orientation of the texture may be applicable as well.

As pixel-based visualizations represent large amount of data, human analysts might be overwhelmed by the amount of data shown. Guiding the analyst to potentially interesting pixels can be essential for an effective data analysis. In this section, we will discuss several techniques guiding the attention of the analyst to regions of interest. We call this process boosting the visual salience of data points. Basically, we differentiate two kinds of boosting approaches. The first one, called *image-driven boosting*, describes cases where information already available in the visualization should be more visually emphasized. An example would be to enhance the visibility of peak values by highlighting them. The second boosting technique, called *data-driven boosting*, adds additional meta information to the visualization which was not included before. One example for this type of boosting would be highlighting all pixels fulfilling a query.

We will first describe several existing boosting techniques and include a small example figure. Afterwards, we will discuss for each technique the effectiveness and applicability. The overall result of the discussion is materialized in an overview table. We will apply boosting in Section 2.3 in order to show the anomalies in a time series. As the anomaly score is added to the raw time series visualization, we will perform a data-driven boosting enhancing the visual salience of unusual measurements.

2.1.2 Boosting Techniques

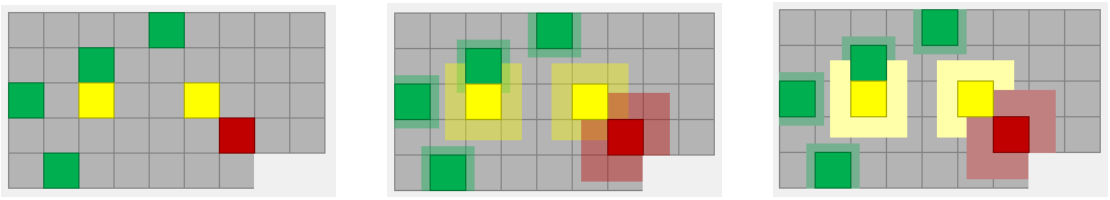
Increasing the visual saliency of data items is very strongly related to the human perception. As perception studies showed [War08], it is in general most beneficial to use another visual channel (e.g., color, shape, motion) for boosting than for encoding data items. At the same time, the human perception imposes several restrictions on boosting data items. Contrast effects resulting from glyphs for instance may influence the perceived color. Furthermore, different boosting techniques should not be applied simultaneously when boosting different data aspects. How-

ever, different techniques may be applied when boosting the same data aspect increasing the visual salience.

The boosting techniques introduced and described here are mainly based on the work of Ware [Waro8]. Ware describes boosting techniques with focus on visual variables resulting in our comparison of boosting techniques with focus on pixel visualizations. Some techniques mentioned below, such as hatching and distortion, require the pixels to exceed a certain size to be effective.

Boosting with Halos

The visibility of pixels can be enhanced by increasing their size. If we do not want to change the layout of the pixels when increasing some pixels, we will have to overplot neighboring pixels. Ware [Waro8] describes this approach as adding a surrounding color. The distinction of data item and surrounding Halo is supported by using translucent colors. Transparency comes along with the problem of mixed colors in areas with overlapping Halos. There are different variants of Halos, being explained in Figure 2.1.1. Please note that Halos are always drawn in background and will never overdraw any data pixels.



(a) Raw pixel visualization **(b)** Translucent Halos **(c)** Opaque Halos

Figure 2.1.1: Halos may be drawn semi-transparent or opaque. In both cases, the painting order is influencing the result. Reprinted from [OJS+11], © 2011 The Eurographics Association and Blackwell Publishing Ltd.

Boosting with Colors

There are two possibilities to use coloring for boosting. We can either improve the visibility of single, important data items or make the global trend more salient.

In the first case, we will apply contrast colors in order to highlight interesting data items. One possibility is depicted in Figure 2.1.2 where red color highlights pixels for a grayish col-

ormap. Additionally, the color wheel can be used in order to determine suitable contrast colors for instance supported by Adobe Kuler [ADO15] or Color Scheme Designer [Sta15]. The perceptual distance between pixel color and chosen contrast color can be calculated in the CIE color space [CIE78]. Depending on the homogeneity of the pixel visualization the contrast has to be lower or for heterogeneous visualizations larger. Using a gray scale colormap allows for instance applying coloring for highlights.

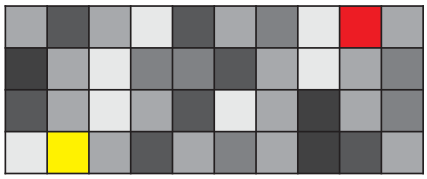


Figure 2.1.2: Using color contrast to visually boost data points.

The second possibility to boost by color is to visually represent the global trend, as shown in Figure 2.1.3. This works especially well for sparse data sets where not all pixels have been occupied displaying data points. These empty pixels can be colored less saturated according to the global trend, such as the average or median. Data pixels with a similar color to the trend coloring will consequently become less visible. Coloring not used pixels should therefore only be used representing an already visible trend.

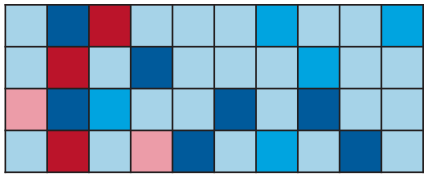


Figure 2.1.3: Background coloring represents the global trend (average value).

Boosting with Distortion

Distortions of the screen space can be used to enhance the visibility of important data pixels and reduce the visibility of uninteresting ones. Applying distortions affect the visual variables

size and position and is only possible if the data points cover a large enough display area. Increasing and decreasing the size of pixels will guide the analyst's attention to interesting areas. Furthermore, distortions increase the scalability as not important areas are decreased offering free space for more data items. A schematic example for distortion can be seen in Figure 2.1.4.

Figure 2.1.4: Distortion of single data points according to their value.

When pixels are layouted in a regular grid, distorting the visualization row- or column-based is easily achievable. For instance, we can count for a column (or row) the number of important pixels and determine the importance of the respective column (or row). Distorting columns is applicable for example when the columns denote points in time and the rows represent different measurements. In Figure 2.1.5, we distort columns according to the average data value (higher values result in wider columns).

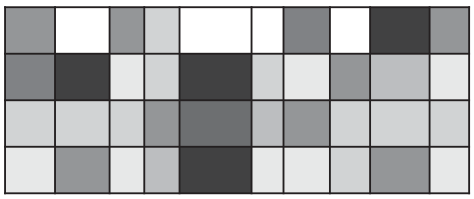


Figure 2.1.5: Distortion of columns according to the aggregated importance.

As soon as the context of pixel is important (e.g., geospatial applications) another kind of distortion should be applied. In this case, the local neighborhood of the boosted pixel should be increased as well, resulting in decreased overplotting in boosted regions. We created in Figure 2.1.6 an example distortion applying the fisheye distortion technique [KR96].

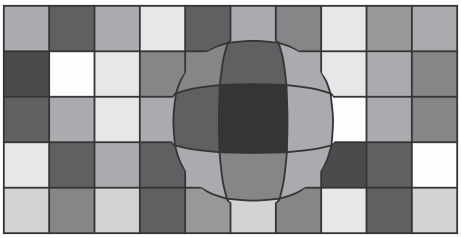


Figure 2.1.6: Using non-linear distortion to emphasize important pixels.

Boosting with Hatching

The visual variables texture and orientation or, more specifically in our case, hatching can only be applied if the area of pixels is large enough. Different orientations of the hatching lines can support the pre-attentive grouping of semantically related data points. In our example Figure 2.1.7, we use the four main directions (horizontal, vertical, and both diagonals). The difference between the horizontal and vertical lines seems to be higher than the distance between the two diagonals. This may result from the reading direction which we are adapted to. Applying hatching would allow us to additionally encode a numerical value by the hatching density. Though, it is not reasonable when hatching very small display areas to additionally vary the amount of hatching.

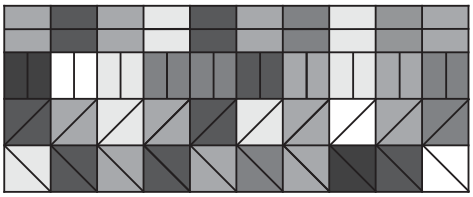


Figure 2.1.7: Different hatching orientations can group related pixels.

Boosting with Shapes or Glyphs

Boosting points of interest on a map with glyphs is one of the most common ways. For instance, showing criminal incidents or ending and beginning of a route are typically marked by flags. Humans are able to easily spot such highlights, though glyphs have a major problem. Glyphs need a larger space than the original data points resulting in overplotting. Using different shapes

representing pixels will also change the area covered resulting in contrast effects and maybe even different colors perceived. Using different shapes require the data points to exceed a certain size. In Figure 2.1.8, we apply both glyphs and different shapes for boosting.

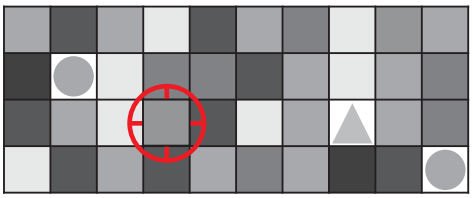


Figure 2.1.8: Adding glyphs to the visualizations allows boosting pixels. Different kind of pixel representations can be also used to emphasize certain data points.

Boosting with Blurring

Kosara et al. [KMH+02] describe in their user study, how blurring can be used to guide the user's attention to important areas. Blurring of not important data points will let the user focus on the unblurred areas. Kosara et al. show in their study that humans detect unblurred items in a blurred context preattentively. Figure 2.1.9 shows one example, blurring unimportant pixels and boosting important ones.

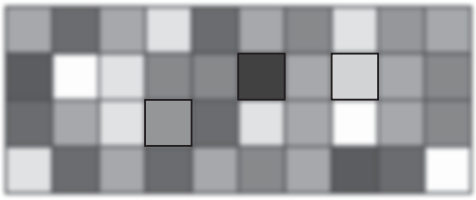


Figure 2.1.9: Unblurred pixels are preattentively in the focus.

2.1.3 Comparison of Boosting Techniques

Depending on the application scenario all the boosting techniques described above are more or less effective. We will present in this section an overview comparison of the approximate ef-

	image - driven boosting			data-driven boosting			trend		effectiveness in boosting	changes to layout	resolution dependency
	pixel		passage	pixel		passage					
	sparse	dense		sparse	dense		sparse	dense			
halos	+	-	-	+	-	-	o	-	+	o	+
background coloring	o	-	-	-		-	+	-	+	+	+
hatching	-		+	-		+	-		+	+	o
color map	+		+	-		-	-		+	+	o
animation	+		o	+		o	-		+	+	o
distortion	o		+	o		+	-		o	-	o
glyphs	+	o	-	+	o	-	-		o	+	-
shape	+		+	+		o	-		o	+	-

Figure 2.1.10: Comparison of the different boosting techniques with respect to data density and amount of boosted pixels. We furthermore investigated the effectiveness in boosting, the amount of changes to the layout, and the resolution dependency. A + symbol represents scenarios in which the respective technique is performing well and a ○ shows medium performance. All combinations of techniques and scenarios marked by - should be avoided. Reprinted from [OJS+11], © 2011 The Eurographics Association and Blackwell Publishing Ltd.

fectiveness. We will consider data density, as the density will affect the amount of empty space in the pixel visualization. Furthermore, we take the two types of boosting into account, namely *data-driven* and *image-driven* boosting, being presented in Section 2.1.1. Lastly, we also consider the possibility to boost the overall trend by the proposed techniques. We assess for each technique the effectiveness in boosting together with the resulting layout effects and the resolution dependency of the respective technique. Instead of conducting a large user study assessing all techniques, we involved perception theories from literature. The result of our comparison can be found in Figure 2.1.10. We see the table as a systematical collection of different boosting techniques and as first step for an exhaustive user study.

In Figure 2.1.10, we differentiate between shapes and glyphs as in our case glyphs exceed the pixel area, whereas shape boosting will only use the pixel's area. Glyphs will be influenced stronger by the data density compared to shapes.

Though animation has a very high effectiveness in terms increasing visual saliency, it may distract and disturb analysts if too many data pixels are flashing. Furthermore, animation cannot be applied for static media. Animation consequently must be applied with special care. Another important point is that the color perception will be influenced as the background of the flashing pixels will be periodically visible.

In the next paragraphs, we will discuss and reason some results depicted in Figure 2.1.10.

Data density

We distinguish between sparse and dense data sets because some boosting techniques require empty space around the data pixels to be successfully applied. Halos, background coloring, and glyphs are the techniques suffering most from dense data sets. These techniques need some space of surrounding pixels and may partly occlude the underlying pixels. We did not distinguish between dense and sparse data in the case of boosting trends, because boosting coherent pixels is by definition not dealing with sparse data.

Image- vs. data-driven boosting

Image-driven boosting will highlight and emphasize information in a visualization that is already visible. Boosting by adapting the colormap will therefore work for image-driven boosting but not for data-driven boosting. If we change the colormap in the case of data-driven boosting, the original visualization will be changed too much to derive the originally encoded information.

The difference between image- and data-driven boosting in the case of boosting by shapes has another background. Using shapes with a smaller size than the original data pixel will influence the color perception. For the image-driven boosting, this is not as bad as all data pixels with the same color are changed simultaneously. However, data-driven boosting will affect arbitrary colors independent of the original color value.

Boosting single pixels vs. passages

Boosting passages will negatively affect techniques needing sparse areas around the boosted pixels, as Halos, glyphs, or background coloring. Although, other boosting techniques might be positively influenced boosting a passage of pixels. Spotting a coherent set of hatched pixels is easier than spotting one single hatched pixel. The same is true for distortion being better visible when a set of pixels is distorted. Glyphs are a special case, as they have to be designed carefully to support boosting of passages.

Boosting trends

Compared to the number of boosting techniques for boosting single pixels or passages there exists only one method for effectively boosting trends. Using background coloring or Halos

boosting the average color can be applied only for sparse data sets. In the case of dense data sets, only some here described techniques can be recommended.

2.1.4 Conclusion

We presented an overview to possible boosting techniques in pixel visualizations and discussed their applicability. The perceptional issues in boosting were discussed and related to the proposed methods. We described three different foci of boosting, in specific, image-driven, data-driven and trend boosting . Depending on what to boost the recommended set of boosting techniques varies. We present the estimated effectiveness for each technique under several conditions in a comparison table.

The work on boosting techniques described in this section were the basis for our work described in Section 2.3 dealing with anomaly-aware visual analysis of power consumption data. In this work, we visually emphasize all data points being visualized in a pixel visualization with unusual values by applying boosting techniques.

2.2 Peak-Preserving Prediction

This section is based on the following publication[2]:

A Visual Analytics Approach for Peak-Preserving Prediction of Large Seasonal Time Series

M. C. Hao, H. Janetzko, S. Mittelstädt, W. Hill, U. Dayal, D. A. Keim, M. Marwah, and R. K. Sharma.

Computer Graphics Forum, Vol. 30, Iss. 3, pp. 691–700, 2011. [HJM+11]

2.2.1 Preface

Deriving information by analyzing the past and extrapolating this knowledge into the future is one important aspect of time series analysis. More specifically, detecting patterns and trends based on historical data and inferring the future is challenging as the only thing certain about the future is uncertainty. Training prediction models is more or less improving the educated guesses about the expected future. Though, predicting unexpected or previously not modeled patterns is impossible.

Prediction methods are already applied in numerous applications, e.g., weather forecasts, warehouse logistics, or power consumption. In the area of data center administration, for example, it is crucial to predict the power and resource consumption in order to budget the resources without exceeding capacities.

There are several state-of-the-art prediction methods with its own characteristics and applications. Statistical methods like ARIMA and Holt Winters [Chao3] or G-TSFE [CSC+05] are model-based. The second group of prediction methods is smoothing, trend, or similarity-based [BAP+05]. Depending on the applied method different patterns can be modeled. In the case of Holt Winters seasonality can be modeled while ARIMA is used for non-seasonal data. The selection of prediction methods highly depends on the application area and furthermore the data analyst's skills driving the prediction are needed.

[2]In this work, Walter Hill proposed to use the Douglas-Peucker simplification for smoothing. I had the idea to use the recursion level of the smoothing algorithm as a weight for the prediction. Sebastian Mittelstädt implemented the new smoothing and prediction into an earlier prototype implemented by myself developed for applying Holt-Winters. Multi-Scaling and Brushing & Linking were also implemented by me. Ming Hao, Umeshwar Dayal, Daniel Keim, Manish Marwah, and Ratnesh Sharma helped with fruitful discussions and advices.

The work described in this section is extending our ideas presented in an IEEE VAST09 poster paper [HJS+09]. We propose a novel analysis and prediction method especially focusing on peaks of the historical input data. We apply our prediction technique in the application domain of data centers, where peaks in resource consumption may be critical, and finally evaluate the quality of our predictions.

Our contributions

We developed peak-preserving smoothing combined with peak-preserving prediction allowing the prediction of seasonal data. Our visual interface allows the user to interactively control the process and integrate his expert knowledge. Together with visual feedback of prediction accuracy and certainty bands the user gets immediate feedback and can adjust the prediction to his needs. Peak-preserving smoothing techniques allow removing noise while retaining peaks. Last but not least, the data analyst can weight the influence of peaks versus the influence of time (e.g., recent data have higher influence to the prediction then older data points).

We discuss the related work in Section 2.2.2, followed by a description of our approach in Section 2.2.3. Section 2.2.4 introduces the peak-preserving smoothing and prediction methods in detail. The next Section 2.2.5 puts our approach in the context of Visual Analytics and describes the possibilities provided by our visual interface. In Section 2.2.6, we apply the presented methods to real-world datasets and evaluate afterwards our prediction results in Section 2.2.7. Lastly, we discuss advantages and disadvantages as part of the conclusions in Section 2.2.8.

2.2.2 Related Work

Predicting time series is a very relevant and actively researched area with many developed methods. We differentiate these methods into two categories, namely pure prediction algorithms and methods combined with visualizations, and describe them below in more detail.

Prediction Algorithms

We mentioned above already two very prominent prediction methods, namely ARIMA and Holt Winters. ARIMA (Auto Regressive Integrated Moving Average) models linear stochastic processes by two terms, the regression and the moving average. Therefore, ARIMA per se can not model periodic or seasonal patterns. An extension to ARIMA was developed by Sadek

[SKC03] which captures both the short- and long-range features by predicting values at different time scales. Furthermore, the extended ARIMA reduces the computational complexity by a simplified prediction scheme. This extension is especially adapted for self-similar time series.

Seasonal or any periodic time series data are supported by Holt [Holo4] and Winters [Win60]. The prediction is performed by exponential smoothing and therefore capable of modeling seasonality. An extension of the Holt Winters technique was proposed by Taylor in [Tayo7]. The aim of this work was to predict supermarket sales on a daily basis by applying exponentially weighted quantile regression. Taylor furthermore integrated the cumulative distribution function resulting in improved prediction results. We applied Holt Winters as one state-of-the-art prediction techniques and compared our prediction results in Section 2.2.7.

Prediction visualization techniques

Visualizing and inspecting the prediction results is the obvious next step after predicting values. In the application domain of predicting the runtime behavior of multi-threaded programs Broberg [BLG99] applied Kalman Filters [Kal60]. The results of this prediction process were visualized by line charts. Multiple visualization techniques were applied by Ichikawa [ITFY02] in order to represent stock price predictions. Ichikawa used line charts and color-encoded time series visualizing several time series simultaneously. Statistical analysis tools like SAS integrate prediction methods as well. The SAS Forecasting System [SAS13] even supports automatic model fitting. Croker [Cro07] showed how to visually present the different confidence bands in a line chart representation using SAS. We extended these ideas and enabled the analyst to assess the prediction quality by using the old data points as an evaluation criteria as described in Section 2.2.5.

2.2.3 Our approach

Comparing the different existing techniques leads to the conclusion that they are sound and advanced methods but lacking one important property. For our use case it is crucial to detect and integrate peaks in the time series. These peaks might represent exceeding of the provided power or, even more dangerous, exceeding of cooling capabilities. Applying prediction techniques performing regression will smooth away the peaks. Furthermore, the distance in time has to be regarded as well. The more recent measurements should have a higher impact to the prediction as the older ones.

Nevertheless, there are usage scenarios where peak-preservation is not necessary or beneficial at all. In sales applications or signal processing peaks are not important or even considered as noise. In these cases smoothing techniques reducing noise and peaks are applied during the prediction process.

We propose a peak-preserving prediction method including a temporal weighting of values by giving recent measurements more importance than old measurements. In order to remove noise without any smoothing of peaks, we integrate a peak-preserving smoothing algorithm as well. The analyst can influence the prediction process by a weighting slider controlling the peak-preservation versus time distance.

The schematic process of our visual peak-preserving prediction is depicted in (Figure 2.2.1). We propose an iterative two-step approach with user control possible in every stage.

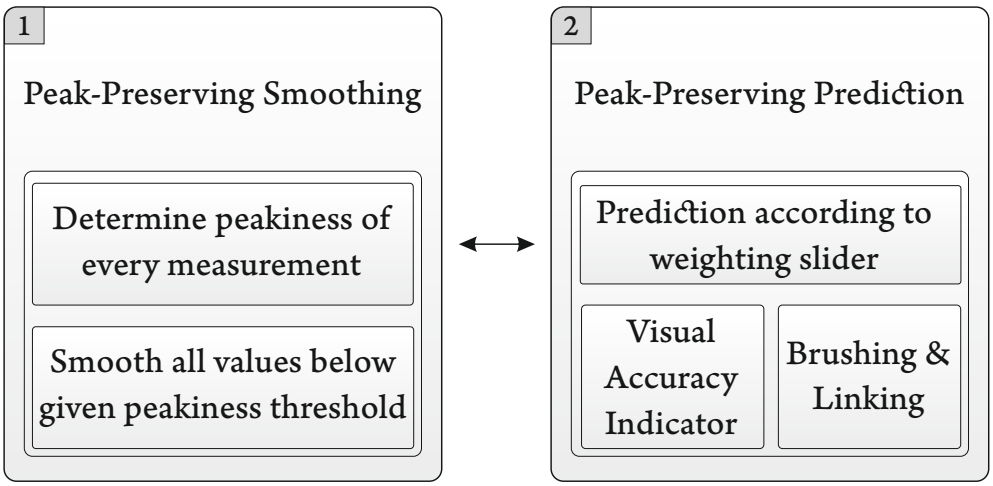


Figure 2.2.1: Visual Peak-Preserving Prediction Process.

1. Applying user-controlled *peak-preserving smoothing* in order to preprocess the time series removing noise. All values that are not sufficiently extreme, e.g., local minimal or maximal, are smoothed.

2. Compute the *peak-preserving prediction* using the user-provided weighting scheme between peak-consideration and time distance. The analyst can freely choose any weight depending on the application. The system provides two visual accuracy and certainty indicators helping the analyst judging the prediction quality. For deeper insights, we

provide Brushing & Linking helping the user in understanding how the prediction was computed.

2.2.4 Peak-Preserving Smoothing and Prediction

The following paragraphs describe our techniques in more detail. We will introduce the peak-preserving smoothing and the peak-preserving prediction technique. Both approaches are automated techniques with parameters being controllable by the analyst.

Peak-preserving smoothing

When we started our experiments with power consumption and workload in data centers, we noticed that existing prediction methods did not lead to results being good enough. The first reason we found was that the raw input data is typically very noisy badly influencing the prediction results. Applying smoothing methods reduces this effect but at the same time may remove potentially valuable information. Furthermore, smoothing should not affect the overall shape, local extremes, and global trends of the time series.

The first results of our experiments are published in a VAST poster 2009 [HJS+09]. We use weighted moving averages for smoothing by applying a Gaussian distribution. Afterwards, we apply time distance based weights for predicting future values. The problem hereby is that the Gaussian smoothing is basically a low-pass filter removing peaks, as they are they have high values in the frequency domain. Following this approach we loose the peaks being important for our usage scenario.

Improving our approach we decided that during the smoothing process we have to somehow conserve peaks, while still removing noise. We consequently adapt the well-known Douglas-Peucker line simplification algorithm [DP73] for our purpose. Douglas-Peucker reduces a line or graph to its most important data points. We exemplify the application of Douglas-Peucker to a time series in Figure 2.2.2. Compared to the original algorithm of 1973, we simplify and speed up the computation by exploiting the fact that time series are simple graphs. We therefore compute distances along the vertical axis and do not use the orthogonal distance measure proposed by Douglas-Peucker. The final results are in the case of time series the same but computed significantly faster.

The first step of the Douglas-Peucker algorithm (Figure 2.2.2 a) is to compute the blue line connecting the first and the very last data point. The data point with the highest distance to the

blue connecting line is determined. The detected point has to be outside the threshold band surrounding the connecting line in order to be considered as a peak point. In the next step, the algorithm partitions the time series into two parts, with both containing the last found peak point as first or last measurement respectively. Recursively, the Douglas-Peucker algorithm looks for peak point in the subdivisions (Figures 2.2.2 b and 2.2.2 c). The recursion terminates when the algorithm finds no more peaks (Figures 2.2.2 c and 2.2.2 d). As a last step, all detected peak points are sequentially connected. The result is shown in Figure 2.2.2 e.

The threshold settings have a high impact on the quality of the simplification results. Unfortunately, the threshold is application dependent and cannot be fixed in advance. We therefore support the analyst picking a good threshold value by immediate visual feedback. The user can set the threshold, which influences basically the amount of simplification, via the peak-preserving smoothing slider.

In Figure 2.2.3, we compare the original input data (a) with the effects of applying moving average smoothing and peak-preserving smoothing. Both smoothing techniques remove noise very well, but the highlighted peak is missing in the case of the moving average smoothing (b). The peak-preservation (c) influences positively the prediction results and still removes noise.

Peak-preserving prediction

Developing the peak-preserving prediction technique, we had two main purposes in mind our prediction should be capable of:

- Predict the global trend and show possible future developments.
- Focus the prediction on peak points reaching critical numerical values.

When predicting time series data it is important to take more than only peaks into account, because peaks do not reflect the development over time. We therefore integrate also the temporal history of measurements, e.g., how recently certain measurements were observed. It is not very likely that very old data points influence current ones, assuming no knowledge about external influences. Including peaks into the prediction and simultaneously taking the temporal dimension into account can be contradictory. It is possible that peaks occurred in the very past and still have to be regarded during prediction. We let the analyst decide how to weight these different prediction foci by an interactive weighting slider. Depending on the slider position either the time-distance or the peak-preservation is weighted higher.

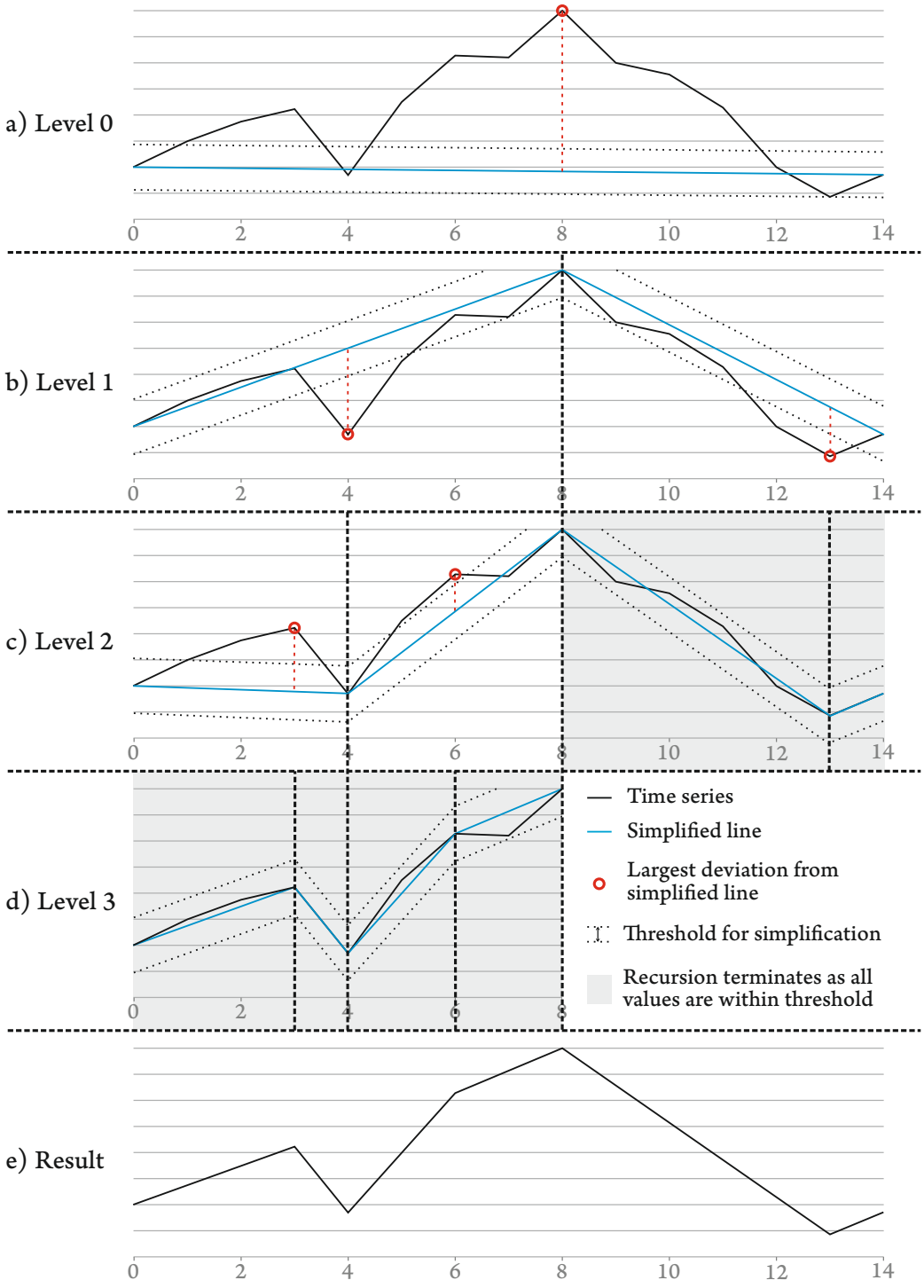
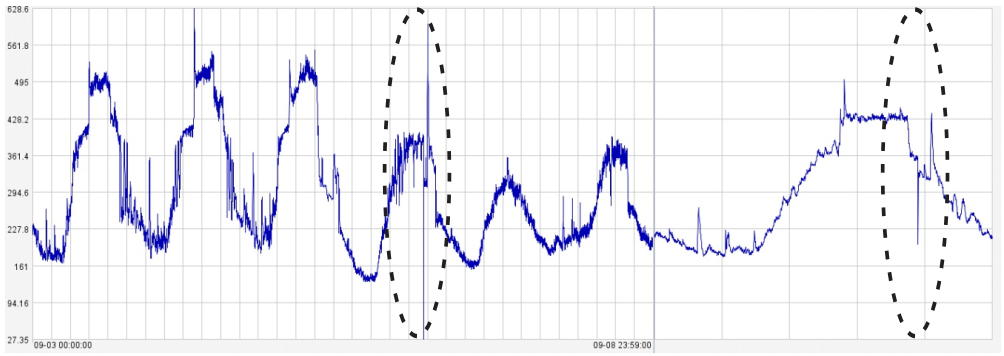
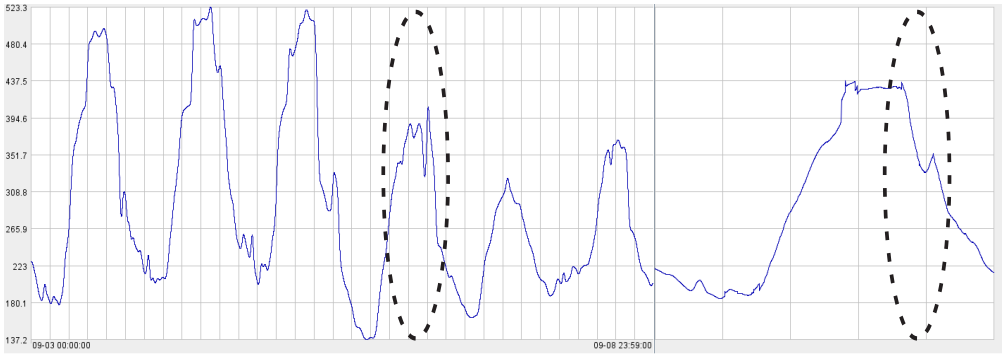


Figure 2.2.2: Schematic explanation of Douglas-Peucker algorithm.



a) Original data



b) Using moving weighted average – peak points are lost in prediction

c) Using peak preserving smoothing – peak is preserved and noise removed

Figure 2.2.3: Comparison of Peak-Preserving Smoothing to Weighted Moving Average Smoothing.

Algorithm 2.2.1: Time series prediction based on daily patterns implementing peak-preservation and development of the time series over time.

```
Input  : double[ ] pastValues ;        // observed time series of the past
Date[ ] datesOfPastValues ;        // dates corresponding to time series
double[ ] importancePeakWeights ;            // peakiness for each value
Output: double[ ] predictedValues ;              // prediction for one day
// create temporary storage:
double predictedValues[ ] = new double[60 * 24];
int counterForEachMinuteOfTheDay[ ] = new int[60 * 24];
// prediction:
double c = calculateConstant(numberOfDays);
for i ← 0 to pastValues.length - 1 do
    Date d = dateOfPastValues[i];
    int minuteOfTheDay = d.getHours() * 60 + d.getMinutes();
    counterForEachMinuteOfTheDay[minuteOfTheDay]++;
    /* Add the current value multiplied with a computed weight
       to the right slot, as we are calculating a weighted
       average                                                  */
    predictedValues[minuteOfTheDay] += pastValues[minuteOfTheDay] *
    combinedWeights(counterForEachMinuteOfTheDay[minuteOfTheDay] * c,
    importancePeakWeights[i], userSetValue);
end
return predictedValues;
```

Our prediction algorithm is shown in Algorithm 2.2.1. The basic idea of the peak-preserving prediction is the predicted values are weighted averages of the historical sums. Giving recent values and/or peaks higher weights is the crucial point for a meaningful prediction. The depicted algorithm is tailored for detecting daily patterns, though it is possible to adapt the algorithm for other periodicities. Different periodicities will be reflected by computing the aggregation slots accordingly. We used for our application a daily grouping because the measured values are mostly influenced by daily patterns. The prediction of the time point 0:00, for example, consists of a weighted average of all measurements made on each day at 0:00. To all of these values, we assign weights according to their recentness and peakiness and finally aggregate them.

In detail, we initialize first some temporary arrays for storing intermediate results and compute a constant *c*, which is described in detail below. The next step is to iterate over all historic values and compute the minute of the day of each measurement. The historic values are added

to their corresponding slot of the temporary storage multiplied by a specific weight explained below. On an very abstract level, the prediction for one minute of the day can be described as follows, where M corresponds to all measurements of the given time interval:

$$pred(minOfDay) = \sum_{m \in M} weightForMeasurement \cdot valueOfMeasurement \tag{2.1}$$

As mentioned before, we have to take the development over time into account and should reflect this by a higher influence of more recent values. We achieve this by computing weights linearly decreasing over time, with an additional assertion: the sum of all weights should be equal to one as these weights are used for an average. The weights prediction one time interval should look like $1 \cdot c, 2 \cdot c, 3 \cdot c, \ldots$ with c being constant, normalizing the weights of the result. The equations below are used to calculate the weights fulfilling our requirements, with n being the number of weights needed:

$$\sum_{i=1}^{n} i \cdot c = c \cdot \sum_{i=1}^{n} i = c \cdot \frac{n \cdot (n+1)}{2} = 1 \tag{2.2}$$

$$\Rightarrow c = \frac{2}{n \cdot (n+1)} \tag{2.3}$$

In order to retain and predict peaks, we compute weights reflecting the peakiness. We use a side outcome of the smoothing algorithm described above determining the peakiness. The peak-preserving Douglas-Peucker smoothing algorithm recursively subdivides the data space. We use the recursion depth of a data point used for splitting in order to approximately determine the peakiness. We use the inverted and normalized recursion levels as weights for our prediction.

We allow the analyst to balance the prediction between time and peak preservation. The two weights computed above are balanced by a weighted average controlled by the user. The method *combinedWeights* used in algorithm 2.2.1 calculates a weighted average of two values with a parameter *userSetValue* (abbreviated to a):

$$combinedWeights(v_1, v_2, a) = v_1 \cdot a + v_2 \cdot (1 - a) \tag{2.4}$$

2.2.5 Visual Analytics Prediction Interface

Our implemented visual interface presenting and controlling the prediction is depicted in Figure 2.2.5. We integrated several interaction techniques and visualizations in order to foster the prediction process. The following describes the applied techniques in more detail.

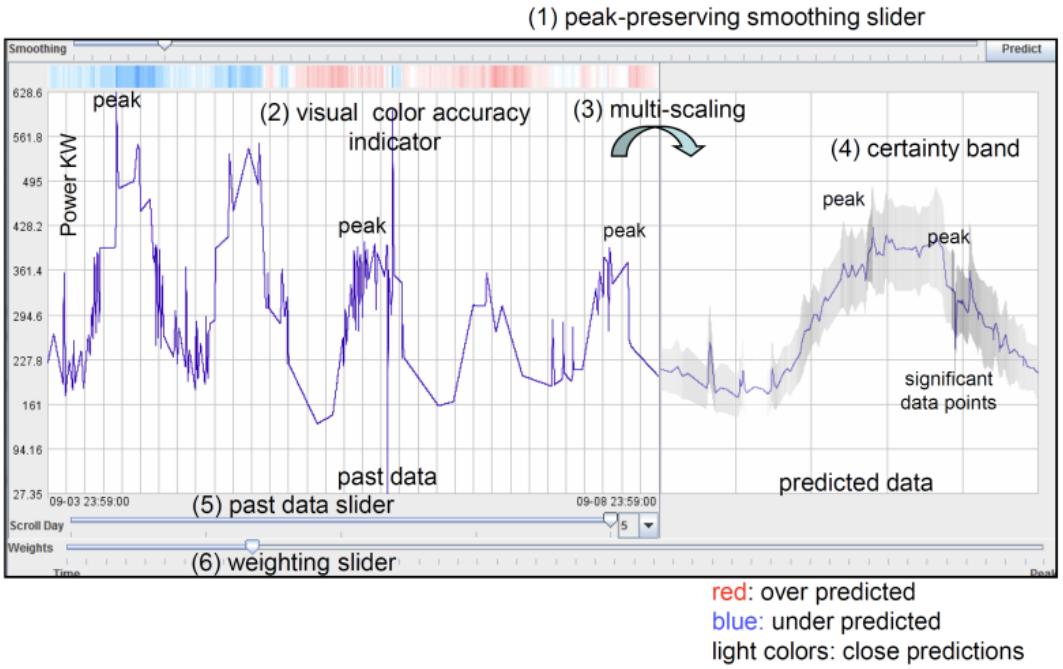


Figure 2.2.4: Screen dump of the prediction interface with observed and predicted values for the power consumption of a server infrastructure. The differences of predicted and real values are shown by the *visual color accuracy indicator*. The certainty band depicts the certainty of the prediction. Reprinted from [HJM+11], © 2011 The Eurographics Association and Blackwell Publishing Ltd.

Accuracy color indicators

Assessing the prediction results it is important to judge the prediction based on the available data. We implemented the *visual accuracy color indicator* shown in Figure 2.2.4 (2). This visualization represents the prediction accuracy of the predictor for the historic values. The differences between actual and predicted values are normalized using the standard deviation. Fully saturated colors indicate larger differences and light colors indicate smaller differences. The hue indicates whether the algorithm predicted too high (blue) or too low (red) values compared to

the actual ones. Figure 2.2.4 (2) depicts the visual accuracy indicator showing at first under predictions (red) and then over predictions (blue).

Multi-Scaling

In order to investigate the predicted data more thoroughly, we integrated multi-scaling of past and predicted data. The user can interactively split the space between historic and predicted values as shown in Figure 2.2.4 (3). Assigning the predicted data more screen space interactively analysts can adjust the visualization to their needs. Multi-scaling is often used by users as the predicted time frame (one day) might be significantly shorter than the history time frame (e.g., one month).

Certainty Band and Significant Data Points

The predicted values come along with uncertainty based on the historic data. The more regular the historic data are, the more certain we are about the predicted values. We visualize the prediction uncertainty by a certainty band as shown in Figure 2.2.4 (4). The band shows the range in which the values can be expected. A narrow band represents points in time where the prediction algorithm is quite certain. We use the standard deviation of the past data to calculate the confidence bands enclosing the predicted values.

We applied shading to the certainty band indicating the significance of the associated data points. High peaks in the past will result in darker areas boosting the visual salience of these important points in time. Lighter areas represent stable or gradually changing curves with peaks being unlikely. Using color saturation guides the user's awareness to the interesting time points with peaks.

Brushing & Linking between Past and Future

Compared to pure model-based prediction techniques, our prediction method is not a black box and the prediction can be visually explained to the user. In order to understand and reason the predicted values, we highlight for a selected predicted value all corresponding past values by Brushing & Linking shown in Figure 2.2.5. At first, the user selects a predicted value for further investigation. The selected value will be highlighted by an unfilled rectangle and all corresponding values in the past will be marked by a filled rectangle. The filling color represents the respective influence of the time point to the prediction, with dark colors being values with

high influence. It is also possible to select a historic value and investigate the corresponding prediction. Consequently, the analyst is able to investigate the prediction of each point in time and to actually see, why a certain prediction has been made.

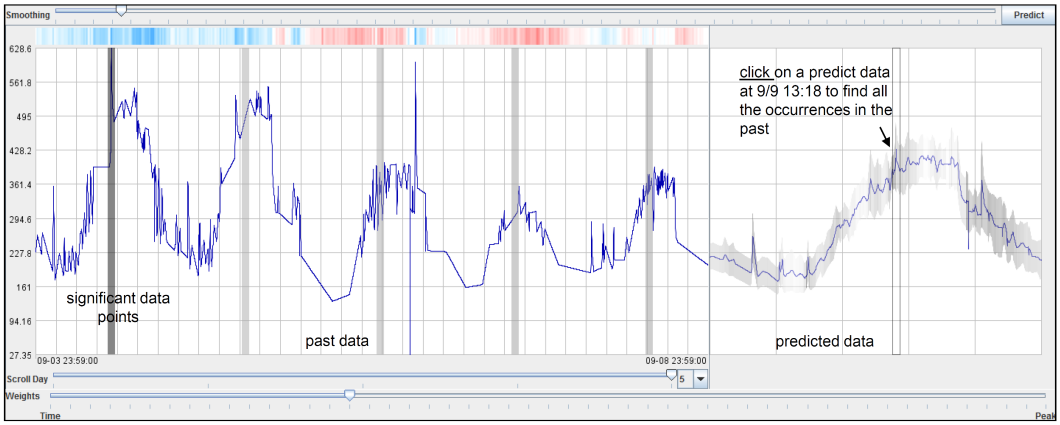


Figure 2.2.5: Brushing & Linking of predicted data to the corresponding historic time slots. Different shades of gray indicate the degree of influence to the prediction (dark: high influence; light: low influence). Reprinted from [HJM+11], © 2011 The Eurographics Association and Blackwell Publishing Ltd.

Past Data Scrolling

Time series data for our application domain is typically too large to entirely fit on one screen when visualized by line charts. We integrated a scrolling mechanism dealing with the limited space available showing only a reasonable part of the whole time series. We show only a certain number of days as shown in Figure 2.2.4 (5). The analyst can control the number of days visible and scroll through the historic data.

2.2.6 Applications

We apply our peak-preserving prediction technique to two data center usage datasets. We will first investigate the daily power consumption patterns resulting from servers and chillers. Secondly, we will examine the usage of servers and predict the application load. These two examples are inherently application scenarios with daily usage patterns being reflected in the implementation described above.

Power Consumption in Data Centers

Data center administrators are interested in the next day's resource consumption based on the previous usage data [PMSR09, SSB+08]. We apply our peak-preserving technique and predict the consumption pattern of the next day. We investigate the time series of a large data center with 2000 racks covering 6,500 square meters monitored in the months July to December 2008.

The power consumption of data centers is periodical with peaks during working hours as shown in Figure 2.2.4. We focused on a shift in the consumption patterns in the historic data. The power consumption dropped significantly from Friday, September 5th, to Saturday, September 6th and increases slowly afterwards. The power consumption is still lower on Monday, September 8th, compared to the end of the previous week. Our prediction results reflect this drop, as the visual accuracy indicator shows over predictions for the latter three days, though getting more accurate over time (lighter colors). Regarding the predicted day, the prediction seems to be reasonable based on the historic data. Using the prediction and configuring low utilized chillers accordingly administrators are able to save a reasonable amount of energy [PMSR09, BPS06].

Server Utilization

The amount of servers dedicated to different applications (e.g., databases, ERP, or backups) has to match the respective utilization. Too few servers will result in long response times or even denials of service. Though, assigning too many servers to an application will boost both the energy and hardware costs unnecessarily. The basic power consumption of an idle server is significant – approximately half of the maximal power usage. Consequently, a server is utilizing power best when it is under full load and idle servers should be turned off. Administrators should analyze the server utilization patterns and relocate and consolidate applications of almost idling servers.

In Figure 2.2.6, we investigate the SAP application resource consumption of three days and predict the usage for the following day. The overall trend of the resource consumption is increasing, which is also reflected by the visual accuracy indicator. The narrow certainty band of the prediction together with the relatively low consumption from 22 pm to 5 am indicates that less computing power is needed during this time. Obviously, there are every morning at 6 am some reports generated and in the evening data integration and backup tasks (circle 2) are performed. As these tasks and corresponding peaks are reflected in the prediction, the admin-

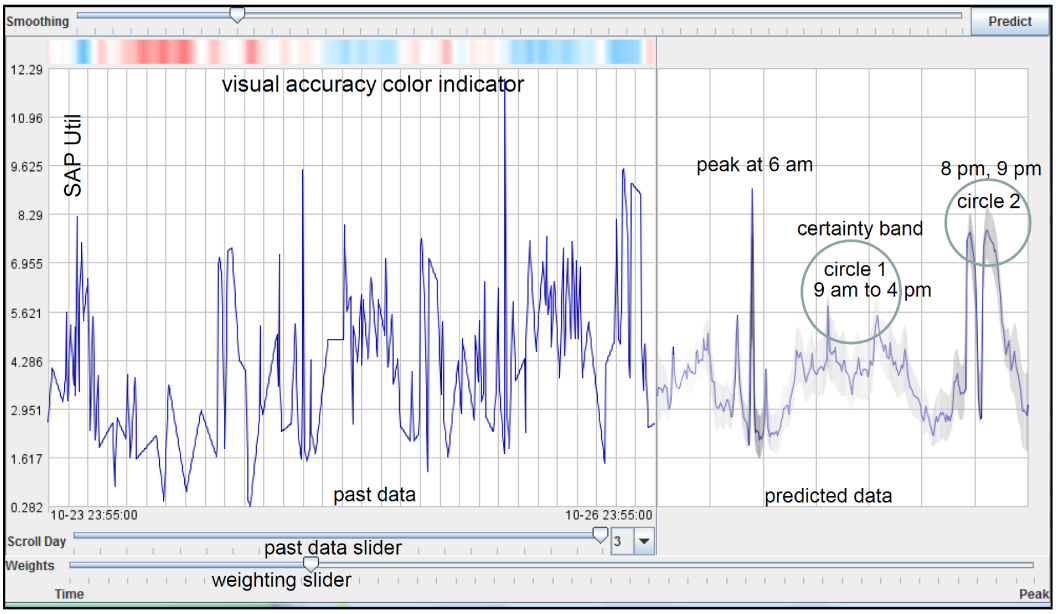


Figure 2.2.6: Peak-preserving prediction of the usage patterns of SAP servers. There are three high peaks and a medium basic load during working hours visible. Reprinted from [HJM+11], © 2011 The Eurographics Association and Blackwell Publishing Ltd.

istrator can reassign servers based on the expected work load.

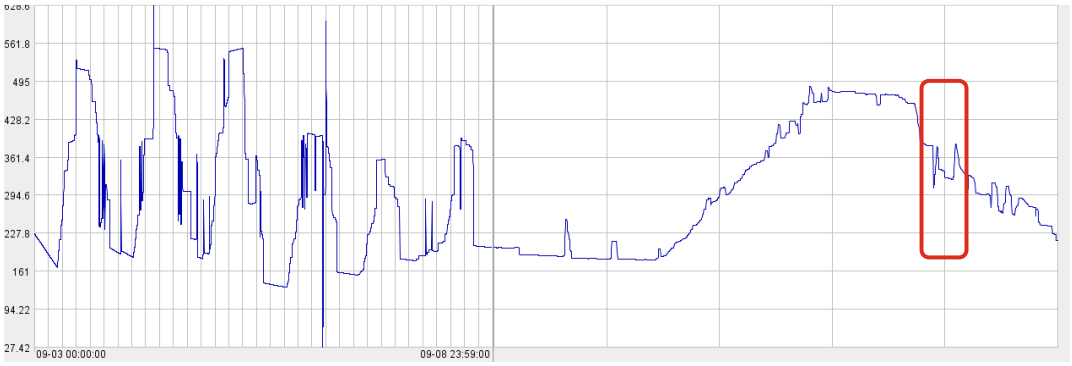
2.2.7 Evaluation

The design goal of the proposed algorithm was to develop a comprehensible, peak-preserving prediction incorporating developments over time. Using weighted averages, we achieved a transparent and peak-preserving prediction. Though, the prediction quality has to be assessed and evaluated. We will first highlight the peak-preservation by comparing our prediction results to the state-of-the-art Holt Winters technique. The second part of our evaluation will focus on the numerical accuracy of the predicted values.

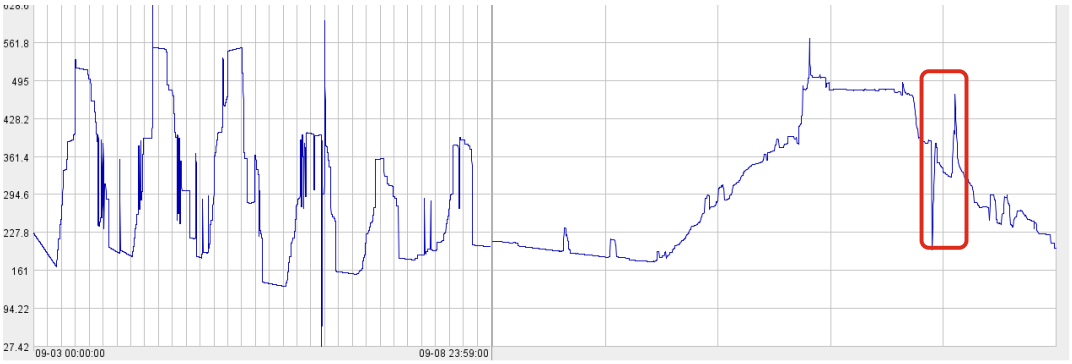
Comparison to Holt Winters

We compare our average-based prediction technique to Holt Winters being a widely used prediction technique for seasonal data. In order to show the peak-preservation by our technique, we set our prediction algorithm to focus only on peaks and disregard time. In Figure 2.2.7, we contrast the existing prediction method Holt Winters with our proposed approach. Our tech-

nique performs better in terms of peak preservation as highlighted by red rectangles. Sudden changes from low to high energy consumption occur typically when administrators re-balance the load of cooling units. These sudden changes in power consumptions have to be considered when scheduling other power demanding tasks.



a) Holt Winters prediction



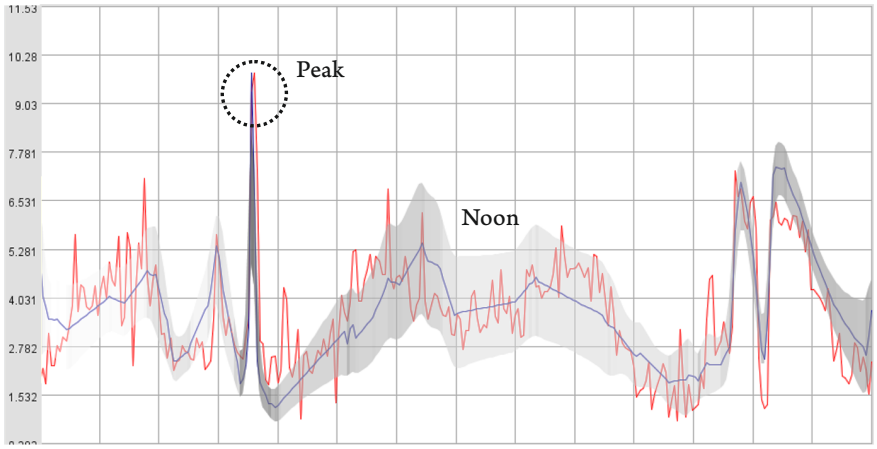
b) Peak-Preserving prediction

Figure 2.2.7: Comparison of Holt Winters technique (a) with our peak-preserving prediction (b). The peaks highlighted in red are better maintained by our prediction technique. Reprinted from [HJM+11], © 2011 The Eurographics Association and Blackwell Publishing Ltd.

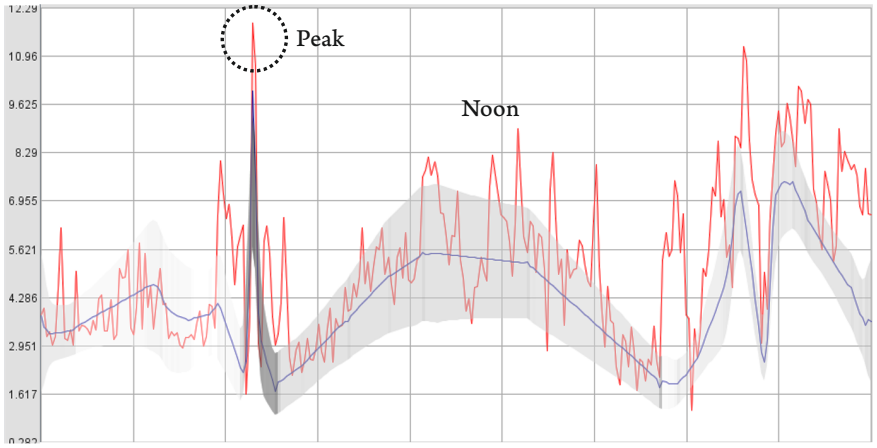
Accuracy evaluation

We assess the accuracy of our prediction technique based on the server utilization from October 6th to 26th (see more description in Section 2.2.6). As it is very unlikely to perfectly predict a numerical value with some decimal places, we decided to assess how often the real value is inside our predicted certainty band. Our evaluation computed a daily accuracy in the range of

70% to 80% with an average accuracy of 75%. In Figure 2.2.8, we predicted two different days and computed the respective accuracy. The left figure shows the predicted values for Thursday, October 14th, with an accuracy of approximately 76%. The right figure depicts Friday, October 22nd, with a prediction accuracy of 74%. Inspecting the figures visually, it becomes obvious that the prediction method does not predict high-frequency changes of values as we applied smoothing beforehand. There is a natural trade-off between smoothing for prediction purposes and the accuracy assessment: the more smoothing is applied, the less possible it is to predict high frequencies which is automatically decreasing the accuracy.



a) Thursday 10/14



b) Friday 10/15

Figure 2.2.8: Visual prediction accuracy comparison between actual and predicted data (blue: predicted values / red: actual values) for two different days. Reprinted from [HJM+11], © 2011 The Eurographics Association and Blackwell Publishing Ltd.

2.2.8 Conclusion

In this section, we described a Visual Analytics approach predicting time series data. The design goal of our prediction technique was to preserve peaks and periodic patterns. We decided to use weighted averages for a transparent and comprehensible prediction. The analyst is able to adapt the influence of peaks and historic developments of measurements tailoring the prediction to his needs. Interactions and visual feedbacks allow steering the prediction process and the estimated quality of the prediction is presented to the analyst as well. We successfully deployed and evaluated our technique in data centers and IT-services centers. The accuracy of the prediction is sufficient and reflects typical usage peaks.

Further work has to be done in order to include external event influences, such as exceptions, holidays, and weather conditions. The prediction algorithm should include conditional computations triggered by specific events or external situations. Furthermore, it would be beneficial if the periodicity of the input data is automatically derived and reflected in the prediction algorithm.

2.3 Anomaly-Driven Visual Analytics of Time Series Data

This section is taken with slight modifications from the following publication[3]:

Anomaly Detection for Visual Analytics of Power Consumption Data

H. Janetzko, F. Stoffel, S. Mittelstädt, and D. A. Keim.

Computer & Graphics, Elsevier, 38(0):27-37, 2014. [JSMK14]

2.3.1 Preface

Commercial buildings consume a significant amount of electricity. According to the Energy Information Administration's 2010 statistics [Uni10], the United States alone consumed an es-

[3]In this work, Florian Stoffel implemented line charts, a flexible way to compute the numerical values for Treemap nodes, and integrated anomaly visualization methods. Sebastian Mittelstädt implemented Spiral Graphs and developed the color boosting technique. I did all the research and implementation work not mentioned above, basically implementing the prototype, computing anomaly scores, and included blurring for anomalies. All sections that were not written by myself are quoted.

timated 1.3 trillion kW. It is about 37% of the total electricity generated. How power is used in a commercial building has a large effect on energy efficiency strategies. The most important energy usage is lighting. Then heating and cooling are next in importance [US 08]. Current approaches for reducing the power consumption for example integrate motion detection sensors for each lamp switching them on and off.

There is a growing interest in understanding how energy is spent in the commercial buildings. Furthermore, building administrators want to know how to reduce the failure rate and detect anomalies. In addition, they want to know how to visualize large volumes of energy consumption data collected by power meters (sensors) in a building to find patterns, trends, and anomalies. In the end, our goal is to find how to automatically discover the anomaly, like unusual power consumption measurements highly differing from old observed patterns, and to reduce the energy cost of a building. For this task, anomalies are of special interest, because they can be caused either by faulty equipment or potentially misconfigured devices consuming significantly more or less energy than required for proper operation.

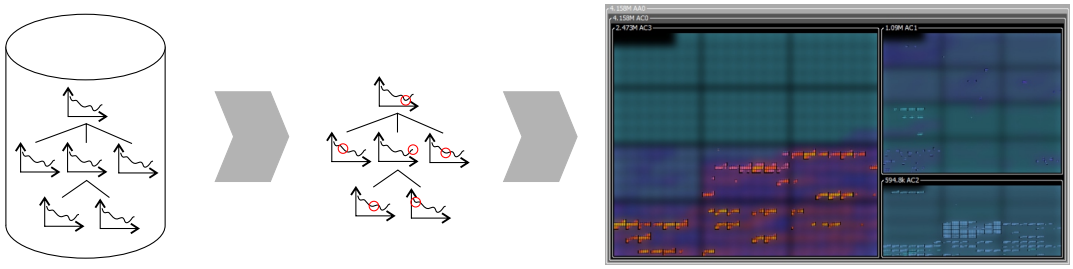


Figure 2.3.1: The input set of hierarchical time series is processed by anomaly detection methods. The resulting anomaly values are visualized together with the time series values by pixel-based techniques. The visualization combines the raw time series with boosting techniques like highlighting and blurring for the anomaly scores. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

In this work, we present an analytical and visual approach to support the building administrators in detecting anomalies and examining energy consumption data as shown in Figure 2.3.1. Our input data consists of a tree of time series reflecting the hierarchical nature of the power meters, e.g., one meter for the whole building and one for each power outlet. In the analytical part, we perform an automatic anomaly detection based on a time-dependent energy

consumption model. We have explored two different anomaly discovery methods. In the beginning, we use clustering-based anomaly detection. Then, we estimate the error rate using the peak-preserving prediction-technique described in the previous section. Both methods have their benefits and drawbacks and are complementing each other.

The last step in our pipeline is the visualization being capable of effectively displaying large amounts of data and, at the same time, allowing quick recognition of anomalous regions in the data. We integrated the three most common time series visualization techniques (line charts, spiral visualizations, and Recursive Patterns) presented in Aigner et al.'s book about time series [AMST11]. Besides giving an appropriate overview of the data, the visualization is also able to support the administrator in a more detailed examination of the data, for example areas with unusual power consumptions by interaction facilities. In addition, the visualization is capable of showing the hierarchical nature of the data set. This is necessary, because commonly the energy consumption of different floors or buildings is independently monitored resulting in an inherent hierarchy in the recorded data.

Our methods rely purely on the recorded power consumption data, which we did not clean in any way as the data was in very good shape. There are many external influences to the power consumption, like the environmental conditions or the number of people working in an office building. The large number and high complexity of external factors prohibit the fully automatic diagnosis of anomalies. Hence, a human subject matter expert is needed to validate found anomalies and investigate the interesting ones. Even though it is possible to think of extensions for an automatic analysis of anomalies like incorporating external factors as weather data and holidays.

It is important to note that our methods are applicable not only to power consumption time series data sets, although they have been developed with a particular application in mind. This is caused by the general nature of time series data and the generality of both, the analytical and the visual methods presented in this paper. The most application-dependent part of this work is the anomaly detection being designed for daily patterns.

2.3.2 Related Work

Reading energy consumption statistics shows that commercial buildings have a high energy usage, which motivates many research projects developed to improve power efficiency. Within the context of our work two main categories can be distinguished: analysis of power consump-

tion data (detecting whether the energy consumption performs normally or abnormally over different locations and time) and visual analysis (visualizing similarities and anomalies with appropriate interaction techniques).

Analysis of Power Consumption Data

Applying data mining techniques for power consumption data is a known approach for identifying abnormal usage behavior. Agarwal et al. [AWG09] examined 6 months of data from the UCSD campus, including aggregate power consumption of four buildings. Agarwal et al. focus more on the setup of power meters and provide only simple visualization methods like line charts. Catterson et al. [CMM10] used an approach to monitor old power transformers. Their goal is to proactively search for abnormal behavior that may indicate the transformer is about to fail. Similarly, McArthur et al. [MBMM05] searched for anomalies to detect problems with power generation equipment. Jakkula and Cook [JC10] compared several outlier detection methods to find which is better at identifying abnormal power consumption. Seem [See07] used outlier detection to determine if the energy consumption for a day is significantly different from previous days' energy consumption. This is a known approach for identifying abnormal system behavior.

The work conducted at Lawrence Berkeley National Laboratory [MPKP11] focuses on demand response. Mathieu et al. used a time-of-week and piecewise-linear modeling approach to analyze commercial and industrial electric load data. To our knowledge, the unsupervised anomaly detection algorithms from prediction and clustering described in this paper differ from the Mathieu et al. method in two aspects: finer granularity and weighted by time distance (recent data weights more than old data).

The review of several prediction methods for power data performed by Zhao et al. in [ZM12] investigates the effectivity and efficiency. Neural networks and Support Vector Machines were performing better than statistical approaches. We though decided to use the prediction technique developed in [HJM+11] as peak-preservation is one of the main strengths of this technique.

Visual Analysis

Visualization of building energy consumption has not yet been a major focus of research thus far. Most of the energy consumption visualizations have been time series line charts, scatter

plots, and maps [IBM13, UCE07, GPGP09]. Recently, Many Eyes [IBM13] allows analysts to choose a visualization type for analyzing public building electricity consumption. The Google PowerMeter [Goo13] recently provides a free energy monitoring tool for people to view home energy usage.

In addition to these existing tools, improving visualization techniques for time series data is ongoing research work. In SAVE [SLH+11], Shi et al. presented a sensor anomaly visualization engine that guides the user to diagnose sensor network failures and faults using multiple coordinated views. In this paper, we map multiple sensors' time series in a single view to enable users to visually analyze energy usage and identify anomalies. Lin et al. describe in [LKL+04] a visual interface querying and data mining large time series. The focus of Lin's work is the interactive mining of realtime time series to support analysts. In SAGA Dashboard [BRR11], Buevich et al. provided a visual interface for interaction with the sensor network. They require the user to use a device that tracks and visualizes home energy usages. We extend the home energy consumption visual analysis to large commercial buildings with dozens of sensors. We therefore restricted ourself to space-efficient visualizations like pixel-based Recursive Patterns. Furthermore, no pre-defined devices and sensor types in our methods are required. Another related work being capable of visualizing hierarchical time series data are the TimeEdgeTrees introduced by Burch and Weiskopf in [BW11]. The technique shows the time series as one-dimensional, color-coded timelines instead of drawing the graph edges. The hierarchy is preserved better by this approach while the space-efficiency is worse compared to the pixel-based approaches we use. We chose the pixel-based techniques as periodic patterns are easier perceivable. Additional discussions on related work concerning anomalies detection and boosting methods can be found in sections 2.3.3 and 2.3.4.

Our contribution

To leverage the prior work and to support analysts in understanding power consumption data, we combine automated anomaly detection algorithms with interactive time series visualizations. The resulting anomaly score is used to highlight unusual power usages in the time series visualizations. Our contributions in the visual analysis process of power consumption data are:

1. In the anomaly detection process, we
    - detect power consumption anomalies based on either a clustering-based approach or a time-weighted prediction.

- compare the prediction-based method with a similarity based anomaly computation.

2. In the time series and anomaly visualization process, we:

   - map the hierarchical time series onto a Treemap and embed in each Treemap cell the corresponding meter's time series visualization.
   - provide different time series visualization techniques dependent on the analysis purpose.
   - visualize the anomaly score by visual boosting of the raw time series representation.

Furthermore, we provide an advanced visual interface enabling the user to visually analyze the power usage. Histograms for viewing the frequency and power usages of important meters; visual queries for analyzing correlation and similarity; and various options on visualization types, Treemap layout, colormappings, and anomaly score computations enable the analysts to tailor the visualization to their needs.

2.3.3 Anomalies Detection

Detecting and exploring of anomalies in time series is a very important aspect, especially when dealing with power consumption data of physical infrastructure. Saving cost and energy are the main motivations for observing and analyzing consumption data. But when dealing with infrastructure that may be even system-critical, the number of failures must be reduced to an absolute minimum. Early signs of failure should be visible in abnormal power usage patterns. In our main usage scenario abnormal behavior is defined as a difference from the expected daily pattern. Both methods described below assume a daily power usage pattern which, of course, can be different for each day of the week. Both techniques are not limited to daily patterns, but can be easily adapted to the periodicity of the underlying data set. The first described method is based on a weighted prediction, where recent measurements have a higher impact than older measurements. The latter approach is transforming the observed daily pattern in the frequency domain and looking for dissimilarity in a transformed space.

Prediction-based Anomaly Detection

The basis for prediction is an observed pattern and the assumption that it is reoccurring (with slight modifications) in the future. If this assumption does not hold true, the predicted values

may be far off the measured values. Considering this fact the other way round, observed values far distant from the expected ones tell us the model used does not explain the observed values. There might be two reasons, the first one is that the model quality is not good enough and the second one is that the values are really differing from the expected and explainable behavior. We assume our data follows a regular underlying pattern and therefore also assume that the model describes the usual behavior well. Detecting anomalies using prediction follows this idea and is related to the statistical measure of residuals.

The prediction method used is crucial for the reliability and expressiveness of the computed anomaly scores. As already stated above we assumed daily patterns and included developments over time into the prediction process. We decided to use a prediction method developed and introduced in the previous Section 2.2. Basically, this method predicts a value for each minute of the day by taking all previous measurement at the same time of the day. As an example, assume we predict the value for a Tuesday at 11:05 am. We would now average all previous observed values of a Tuesday at 11:05 am. Taking just an average would have the disadvantage of neglecting recent developments in the time series. We therefore used a weighted averaging scheme with higher factors for recent values and linearly decreasing influence weights for older values. This prediction method works very well for weekly patterns and will neglect holidays or other external events. The prediction model will adjust to seasonal changes, but alternating behaviors cannot be modeled by this approach. Furthermore, power usage patterns randomly distributed over a day will negatively influence the prediction quality.

After predicting for each point in a time series the expected values based on all values occurring before this point in the time series, we can compute the difference between predicted and observed values. The difference is an indicator for the abnormality of the point in a time series but needs for higher expressiveness some kind of normalization. From the choice and the design of the prediction method we are assuming a model which may not being applicable to all observed time series. We counterbalance for this fact by calculating the average fitting of our model. More in detail, we compute the average deviation from the predicted values for the whole time series. If a whole time series is highly unpredictable, the differences between predicted and actual values are less meaningful compared to a case when a time series follows perfect daily patterns with small deviations. The computation of the anomaly score is summa-

rized by Equation 2.5.

$$anomaly[time] = \frac{|predVal[time] - obsVal[time]|}{avg_{t \in Time}\left(|predVal[t] - obsVal[t]|\right)} \tag{2.5}$$

The variable *time* is the point in a time series for which the anomaly score is calculated. At this position the difference between the predicted and observed value is computed and afterwards normalized by the average deviation from the model.

Clustering-based Anomaly Detection

The second approach for detecting anomalies in time series data is similarity-based. We assume often-observed patterns to be the usual behavior and rarely occurring patterns to be abnormal. Following this idea, we first have to define and compute the similarity of patterns in order to detect whether a pattern occurs more than once. The approach described in this section is proposed and presented by Bellala et al. in [BMA+11, BMA+12]. The time series is first partitioned into days and afterwards transformed by a Fourier transformation into the frequency domain. Each day of the time series is resulting in a k-dimensional vector in the frequency domain with k being a parameter of the transformation process. The next step described by Bellala et al. is a dimension reduction by multi-dimensional scaling into a two-dimensional space. The density distribution in the reduced MDS space is now interpreted as an anomaly score. Points (time series of a single day) being in a high-density area with many (similar) neighbors are assumed to reflect the usual behavior. Outliers in the 2D space can be seen as days with unusual values and are assigned a high anomaly score. This technique only takes the frequency domain into account and does not integrate external effects like weather data or week of the day.

Comparison of Anomaly Detection Methods

We previously described two methods for computing and detecting anomalies and both come with their advantages and drawbacks. Comparing both methods the most obvious difference is the resolution of the anomaly score. The prediction-based method computes for each point in a time series one anomaly score, whereas the clustering-based method returns only one anomaly value per day. It is of course possible to extend Bellala's technique to cope with hours or even minutes of a day, but noise might influence the clustering approach. This behavior is inherited from the computation of the anomaly scores. The clustering-based technique uses daily time

series and uses them as one data item in the clustering process. An anomaly value is assigned to each data item based on the density distribution. Therefore, there is no possibility to assign different anomaly scores to temporal sub-units of a day (i.e., minutes, hours).

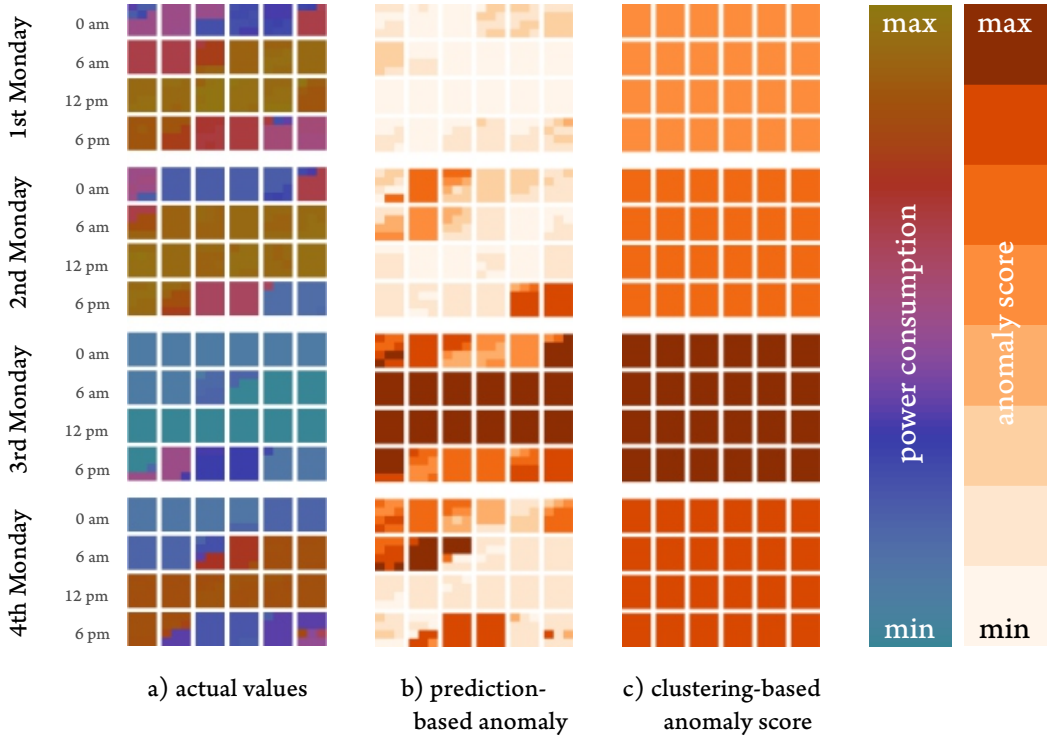


Figure 2.3.2: Comparison of the resulting anomaly scores based on the proposed methods. The third Monday shows an unusual behavior being reflected in the anomaly scores. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

The second essential difference can be seen in the complexity of the methods. Transforming the dataset in the frequency domain and applying MDS results in a data space with axes hard to interpret. But the frequency domain is typically less prone to noise and induces some robustness to the observed time series. Though the transformed data space is complex, there exists the possibility to extract models of typical behavior by computing cluster representatives. Furthermore, the clustering approach allows supporting several typical 'behaviors' of a time series. Just assume a time series alternating between a day-work and a night-work pattern. The clustering approach will assign both low anomaly scores as both patterns are observed often, whereas the

prediction-based method will assign each day a very high anomaly score as averages are computed.

In Figure 2.3.2, we present a visual comparison between both anomaly computation methods. The first column shows a visualization of the observed values for four consecutive Mondays. The exceptional behavior of the third Monday is obvious. This anomaly is reflected in all computed anomaly scores, while the higher temporal resolution for the prediction-based method is visible. Altogether, the clustering-based approach is good for cases when time series switches between different typical behaviors and the prediction-based approach is good for cases when the behavior slightly changes over time following a (seasonal) trend.

2.3.4 Anomalies Visualization

The anomaly scores computed in the previous section are used to highlight important time intervals of the input time series. The visualization for the time series is influencing the design possibilities depending on the visual variable encoding the numerical values. We implemented for comparison three well known, state-of-the-art methods to visualize time series data: Recursive Patterns [KAK95, LAB+09], Spirals [WAM01], and the traditional line chart. These techniques are configured to visualize only one time series.

We will discuss the different design alternatives and motivate our design decisions in the following sections. We focus hereby on the possibilities to encode the time series and the anomaly values simultaneously. We describe all state-of-the-art techniques visualizing time series, namely Recursive Patterns, spiral visualizations, and line charts.

Recursive Pattern

The Recursive Pattern is a pixel technique using coloring to visually encode numerical values using a parameterized space-filling layout. Recursive Patterns are capable of displaying large amounts of time series data in a space-efficient way. By setting proper parameters the resulting display can be calendar-like pixel visualizations. The layout for one day is shown in Figure 2.3.2 and the values of the series are clearly recognizable and furthermore patterns or exceptions are easy to identify.

We integrated borders into the Recursive Pattern layout improving the readability. We extended the recursive algorithm presented by Keim et al. in [KAK95] and added the possibility to specify borders in the same manner as the *widths* and *heights* values. The borders in

x- and y-dimension will be defined in two arrays called *xBorders* and *yBorders*. The Function `drawRecursivePattern` is depicted on the next page and shows the corresponding pseudo code. The presented function is an extension of the recursive algorithm proposed in [KAK95]. We added all lines of code marked with • and changed lines marked with ••. In our application, the borders allow us to build a calendar-like layout.

Function drawRecursivePattern(x, y, level)

```
    if level == -1 then
        SetPixel(x, y, color);
    else
        next_x = ∏_{i+0}^{level-1} widths[i];
        next_y = ∏_{i+0}^{level-1} heights[i];
•       next_border_x = 0;
•       for i = 0; i < level; i++ do
•           next_border_x = next_border_x * widths[i] + (widths[i] - 1) * xBorders[i];
•       next_border_y = 0;
•       for i = 0; i < level; i++ do
•           next_border_y = next_border_y * heights[i] + (heights[i] - 1) * yBorders[i];
        for h = 1; h <= heights[level]; h++ do
            if level == -1 then                                          // odd row
                for int w = 1; w <= widths[level]; w++ do
                    // recursive call of the algorithm
                    drawRecursivePattern(x, y, level - 1);
••                  x += next_x + next_border_x + xBorders[level];
            else                                                         // even row
                for int w = 1; w <= widths[level]; w++ do
••                  x -= next_x + next_border_x + xBorders[level];
                    // recursive call of the algorithm
                    drawRecursivePattern(x, y, level - 1);
••          y += next_y + next_border_y + yBorders[level];
```

In order to incorporate the anomaly score in the Recursive Patterns, we present a color boosting technique that highlights data points by manipulating the intensity of color values according to the anomaly score. The boosting techniques applied are a subset of the techniques presented by Oelke et al. in [OJS+11] and discussed in the previous Section 2.1. Out of the presented

techniques, we used only color highlighting as the visualization will not be overcrowded with visual cues. Color boosting techniques bias the visual impression, which might lead to misinterpretations of the visualization. To keep these artifacts at a minimum, we created a perceptual uniform colormap that only varies over hues without variation in intensity. Since the change in intensity does only minimally shift the hue, the original color tone can be reconstructed mentally.

> "It is known that RGB and HSV are not perceptually uniform and that linear interpolations within these models do not produce color scales with equal or monotonically changing lightness [Kei00]. CIE LUV and CIE LAB have already been proven useful in former Visual Analytics research [Hea96, WK12]. By varying over the color opponents (a and b) but maintaining the same lightness value L, a perceptually uniform colormap can be created in the CIE LAB color space. However, interpolations in CIE LAB can lead to undefined RGB signals and thus, this color space cannot be used in the final application. Therefore, we use the HSI color space [KK95] for intensity manipulation. This color space is an extension to the HSV color space that allows monotonic changes in lightness.
>
> Two proposed color encodings for the anomaly values can be seen in Figure 2.3.3. The first row depicts the original time series without any anomaly scores. We use different intensity levels to encode the anomaly scores and highlight important areas. The effect of the intensity boosting can be seen in the second row of Figure 2.3.3. For further visual boosting we combined blurring and intensity highlighting shown in the last row of Figure 2.3.3." [JSMK14]

We added another highlighting technique, in order to direct the analyst to the anomalous regions of the time series. This highlighting imitates the human perception regarding a focus and the context area, where usually the focus area is sharp and the context area is blurry. We used a similar approach to Kosara et al. and Giusti et al. in [KMH01, GTC+11]. Since the anomaly score is available for every element of the visualization, we are capable to determine the important areas of the time series analytically corresponding to the focus area of the analyst. The implementation adapts locally the blurring according to the anomaly value of each element in the Recursive Pattern. Low anomaly values are more blurry than areas with a high anomaly score. This adaptive blurring technique utilizes the human depth intuition guiding the analyst to

the interesting areas first in a pre-attentive way, depicted in the bottom row of Figure 2.3.3. The blurring will affect the visibility of pixel borders, and it influences the comparability between highlighted and non-highlighted areas. We though believe that the pre-attentive focusing on anomalies helps the analyst in assessing interesting points in time at a glance.

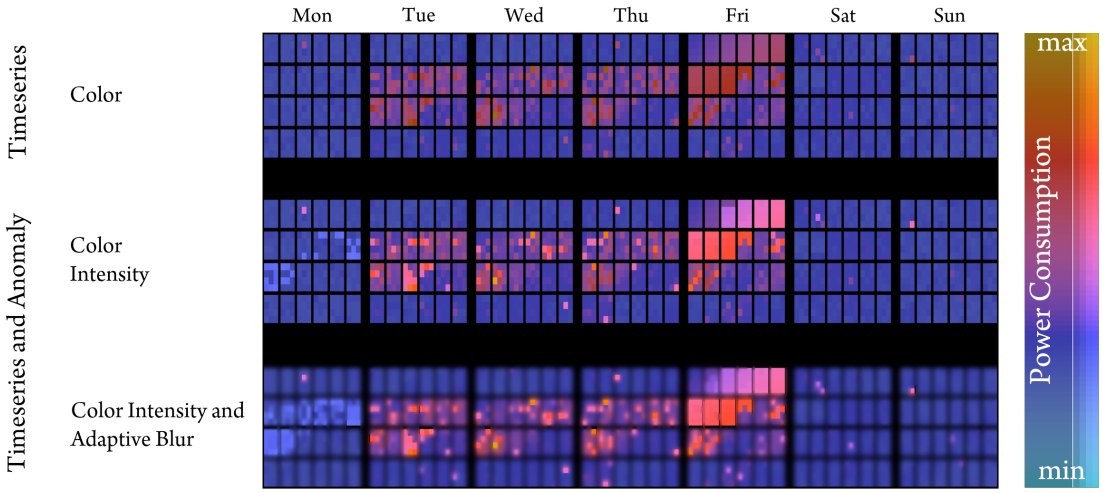


Figure 2.3.3: Different methods to display the anomaly value. Top row: the time series values without anomaly values. Second row: the intensity of the color is adapted to the anomaly value. Third row: color intensity representing the anomaly score combined with adaptive Gaussian blurring. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

Spiral Visualization

The spiral visualization is a technique to display recurring time series data with a fixed periodicity. Our implementation is based on an Archimedean spiral, where the radius grows proportionally to the spiral angle, which leads to a uniform expansion of the spiral over time. In our implementation, each round of the spiral is used to display one day of data. The proportional growth of radius and spiral angle, combined with the absence of any border between each circle makes it possible to build a space-efficient visualization. Comparing the value of the same time span on different days is possible, because these values are on a straight line going from the center of the spiral to the outermost part of the spiral. Each polygon along this line displays the same time span of different days.

To show the anomaly score of each of the displayed time spans, we apply the same color

manipulations as described for the Recursive Pattern above. The right spiral in Figure 2.3.4 shows the described color saturation and brightness adjustment to highlight the anomalous values of the time series. By comparing the left with the right spiral the highlight of the outer ring of the right spiral is clearly visible. There is a time range with unusual numerical values beginning after one fourth of the day and lasting for one quarter of a day. Besides that, some little colorful spots are visible in the right visualization, which were not that visible when modifying only brightness or saturation.

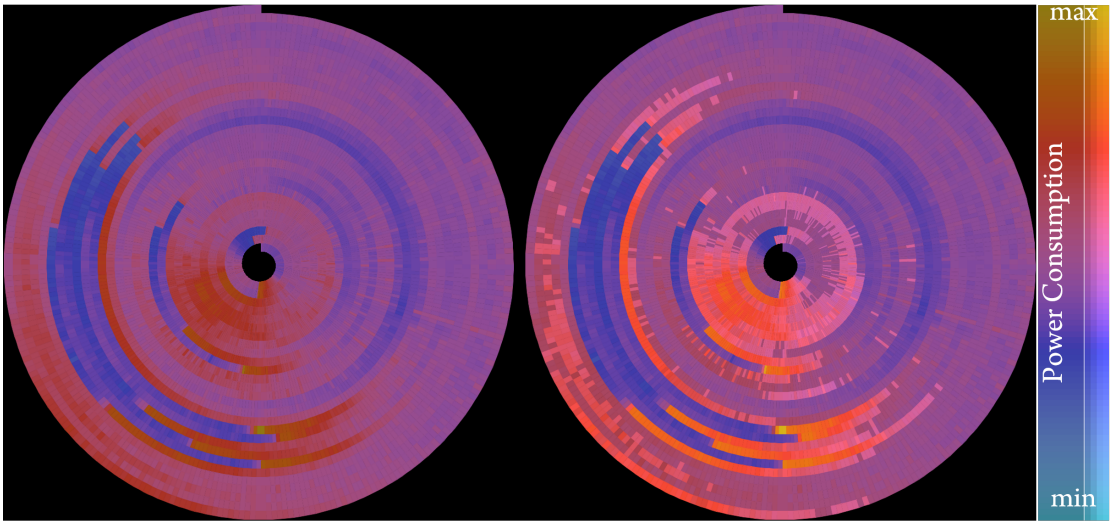


Figure 2.3.4: Spiral visualization of time series. The left spiral shows the actual time series data, the right spiral shows the time series data with brightness and saturation value adapted to the anomaly score of the corresponding polygon. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

Line chart

The most common visualization of time series data is undoubtedly the line chart. The main difference to the Recursive Pattern or spiral-based visualization can be found in the encoding of the actual time series value. In the latter two, the series value is shown by colored polygons, which have a spatial extent. In contrast, encoding the value in a line chart is done by the position on the y-axis. The brightness and saturation-based techniques adding the anomaly value into

the visualization makes no sense in such a positional encoding, having only a very small area available for the coloring. Coloring segments of the line and applying the same techniques to enrich the line with anomaly score information as before is not helpful as line segments are very hard to see. To use coloring a larger line stroke would be necessary, which would introduce high amount of over-plotting and visual clutter. It may be fine for one single line chart displayed on a large screen, but as soon several line charts are displayed the technique does not work anymore.

In order to show the anomaly value simultaneously with the time series values, we used the empty space in the background of the line chart as shown in Figure 2.3.5. For each data point, we plot a red stripe in the background. The anomaly value is mapped to the opacity of the stripe in a way, that for the lowest anomaly value it is completely transparent and therefore not visible. In contrast, the highest anomaly score causes the stripe to have the highest opacity resulting in a clearly visible, red stripe.

To reduce the visual clutter introduced by coloring the background, we also support a minimized view. In this view, the anomaly stripes are only plotted above and below the line chart, which keeps the visualization distraction-free, but still shows the anomaly values. A comparison of both anomaly visualization techniques for line charts can be seen in Figure 2.3.6.

Treemap Integration

We integrated all visualizations in a Treemap display [JS91, Shn92, SKM06, HDKS05] (see Figures 2.3.5 and 2.3.7). The hierarchical nature of our time series dataset is consequently reflected in the visualization. Treemaps are showing the leaves of each selected branch and the nesting depth by borders. The selection of visualized nodes can be achieved twofold, either by interactive roll-up or drill-down operations in the Treemap visualization or by an additional vertical tree representation. Our design choice using Treemaps, though they visualize only the leaves of each branch, was implied by the application needs. The analysts are mainly interested in finding the root-causes of anomalies first and later on in analyzing the impacts by traversing the hierarchy to the root node. Further details concerning the used time series can be seen in the application section 2.3.5.

Each cell of the Treemap contains the visualizations of the time series building one branch of the hierarchy. The border of each of the cells is furthermore drawn in white to allow a clear

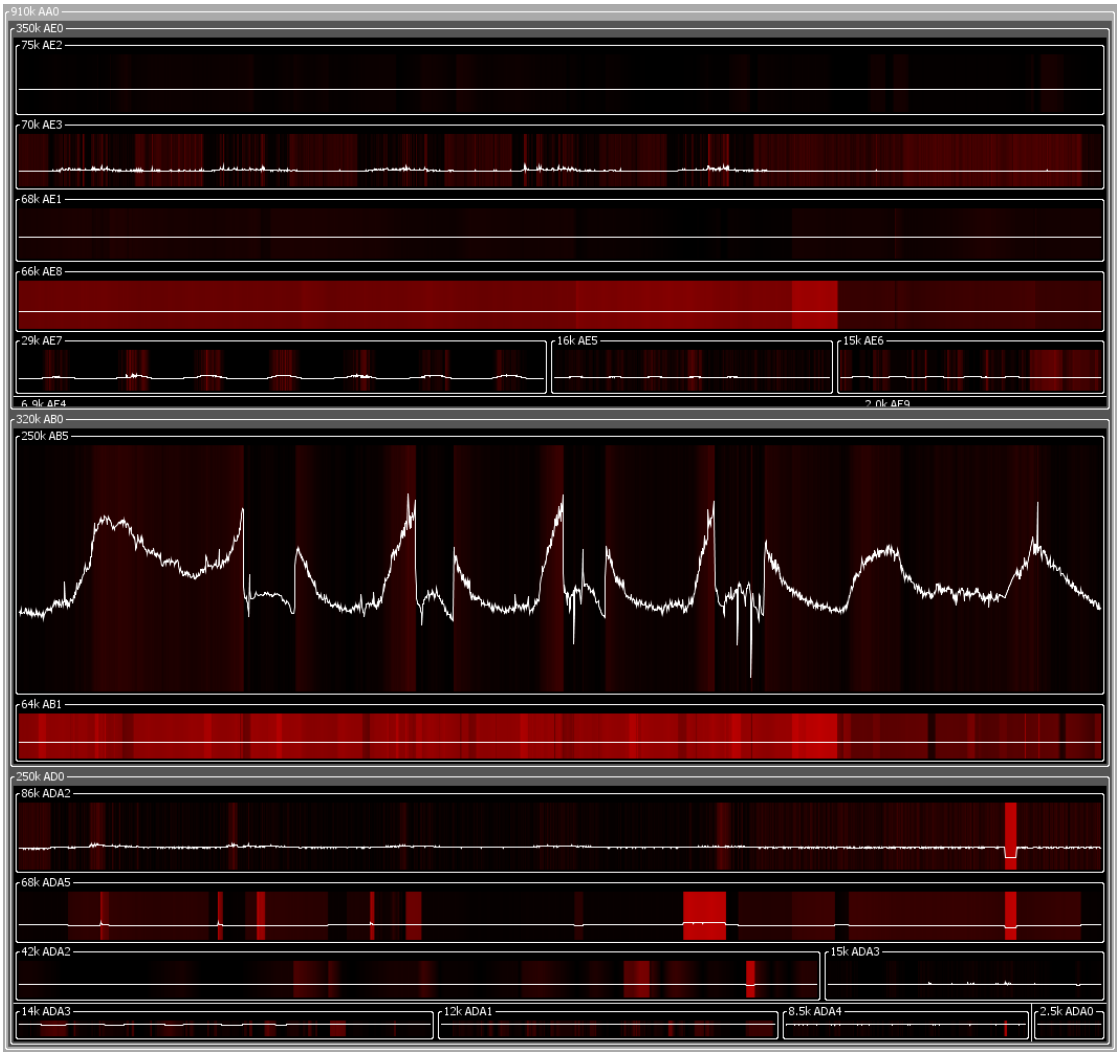


Figure 2.3.5: The line chart visualization in a Treemap with a horizontal strip layout using an aggregated anomaly score to determine the cell size. The anomaly scores are represented by reddish stripes in the background. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

distinction in terms of the hierarchy. The caption of each Treemap cell is used to display the numerical value used for layout and the cell label.

The numerical value is used by the layout manager to compute the final Treemap layout and directly influences the size of a single Treemap cell. The computation of the numerical values is critical for the expressiveness of the visualization since the size of a cell has a large influence on the perception. The size of a Treemap cell can be computed by different measures. Given the interest of an analyst to quickly recognize unusual or highly anomalous time series, the Treemap

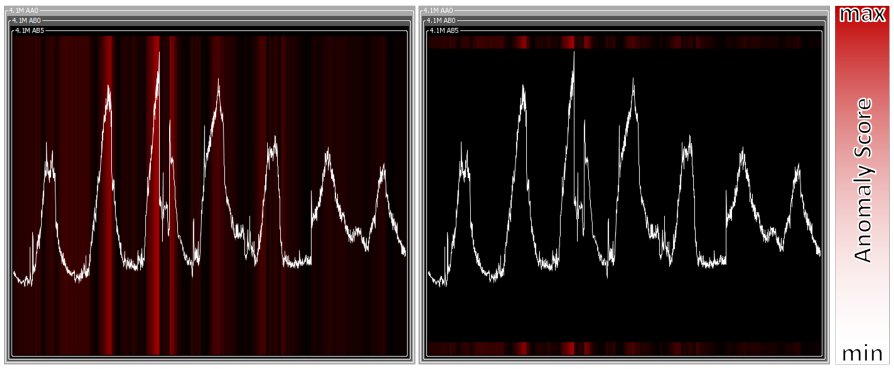


Figure 2.3.6: Comparison of the anomaly visualization technique for line charts. On the left, the whole background is used to show the anomaly scores, whereas on the right, only a small stripe on the top and bottom of the chart background is used to display the anomaly score, which reduces the clutter from the background coloring. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

layout can be adjusted to support these tasks by computing the layout score in different ways. For example, the analyst can choose between the statistical variance, sum, or the arithmetic mean of the anomaly score. To incorporate the level of the anomaly, there is also the possibility to compute the layout based on the product of the anomaly score and the time series value. In addition to anomaly score-based layouts, the sum and the statistical variance of the time series values can be used to compute the layout. Having these choices, the visualization can be adapted to the priorities of the analyst independently of the visualization technique. We also added the possibility to assign the same importance value to each node resulting in a regular layout enabling easy comparisons. Beside the general layout the actual width and height (the aspect ratio) of a single cell is an important factor when using different time series visualization techniques. For that reason, we implemented different layout algorithms for the previously described visualization methods.

A Recursive Pattern has a rectangular shape and we consequently apply a squarified layout [BHvWoo] to the Treemap. This layout algorithm results in a square-like cell, which obviously leads to an efficient space usage of the overall display. In addition, we framed the Treemap cells to improve the overall structure perception of the Treemap and the hierarchical representation.

The circular shape of the spiral graphs combined with the squarified Treemap layout leads to the best readability and space efficiency. We hereby maximize the size of visualization and at the same time use as much space of the Treemap cell as possible. Creating the layout for

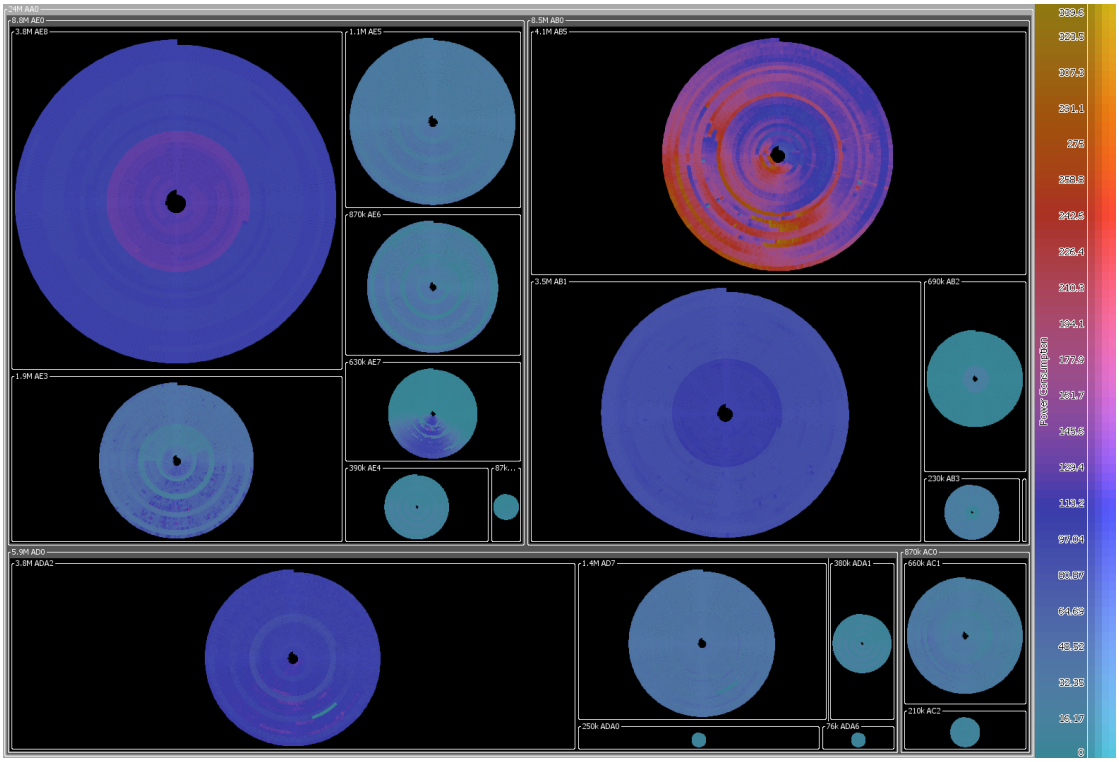


Figure 2.3.7: Treemap visualization of 19 time series, each time series has four weeks of data. Interesting spots or patterns in the data are highlighted and can be therefore easily detected. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

Treemaps containing line charts comes with a fundamental difference to the Recursive Pattern and the spirals: the width of a line chart is much larger than its height as our observed time span is quite long. Consequently, this leads to the conclusion that a squarified layout is not the best choice. Instead, we implemented a so called strip layout [BE95], which ensures that the line charts get more space on the horizontal axis than on the vertical (see Figure 2.3.5). Otherwise, the line charts would be very hard to interpret and this would be an unfair comparison to the pixel-based techniques. Note that the size of each Treemap cell still reflects the numerical value used for layout.

Comparison of Anomaly Visualizations

We have presented three different state-of-the-art visualization approaches for time series and visual extensions to show time series and anomaly score simultaneously. All techniques have their own advantages and disadvantages. The Recursive Patterns presented first have the abil-

ity to visualize large amounts of data in a very compact and space efficient way. Regardless of the shown time range, lasting from weeks and months to years, the Recursive Patterns are always capable of showing the data readably revealing patterns. The visualization is designed such that the value representation by color enables the analyst to easily spot interesting areas or regular patterns, nearly independent of the actual size of the visualization. In Figure 2.3.7, patterns and outstanding time spans are visible, even in the compact Treemap representation of 19 different time series. Having spotted regular patterns Recursive Patterns enable also the cross-comparison in different time series, since the relative position of one point in time is well-aligned. Using Recursive Pattern in Treemap is more difficult to compare the same hour of a day, for example, as the position of the same hour varies through the visualization.

Comparing the same hour is an advantage of the spiral visualization as the periodicity was set to daily patterns. The angular encoding of the time of a day enables these comparisons as a straight line from the spiral center to the outer spiral connects these data values. With such visualizations, it is easy to explore the value of the time series over time. In addition, comparing time ranges and/or spot longer lasting trends is a simple task, since the analyst has only to follow the continuous spiral over time. This is an advantage compared to the non-continuous time display of the Recursive Patterns, where layout breaks are needed, as with any space-filling curve. Line charts are great for detailed visual explorations of continuous data for single time series. For the usage scenario of anomaly visualization, there exist only a few application possibilities, since condensed visualizations are needed as limited screen space is an issue. The low space efficiency of line charts leads to our proposed solution to re-use the empty space in the background to visually encode the anomaly value. We avoid the arising visual clutter by applying the stripe based anomaly visualization, which keeps the anomaly information but reduces the colored area distracting the analyst.

2.3.5 Applications

Our prototype integrating all the presented analytic and visual techniques focuses especially on the detection of anomalies and their temporal occurrence. With this task in mind, two general use cases can be identified. First, general browsing and exploration of the data is important to get an overall impression of the power usage. All different visualization techniques presented above can be chosen to gain from their individual strengths. The second task is the examination of a specific issue, like unusually high or low power consumption. Our system can provide the

analytical and visual insights necessary to find the source of the unusual energy consumption. All visualizations are integrated in the same analytical framework, but use different methods of displaying the power consumption and the anomaly values.

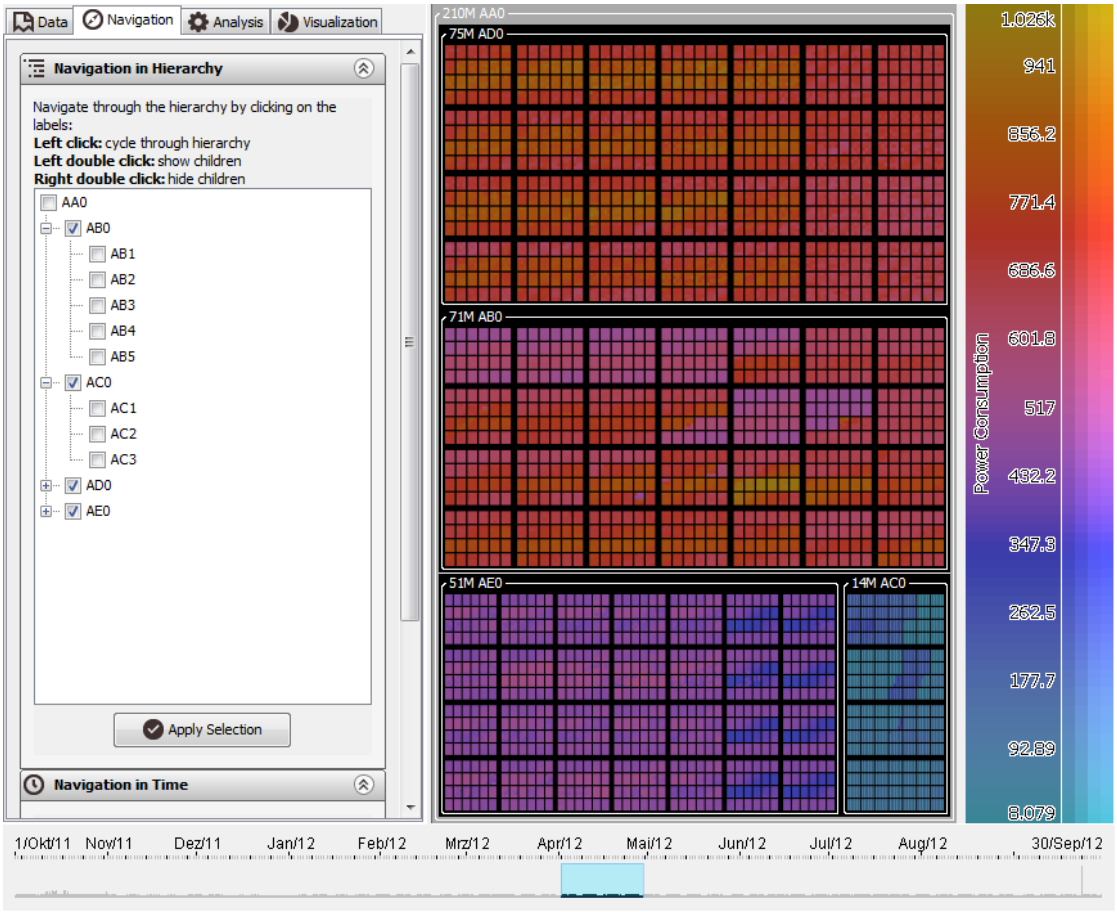


Figure 2.3.8: Screenshot of our prototype showing the hierarchical and temporal selection capabilities together with the visualization panel. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

Analytical Framework

Our prototype consists of three parts reflecting the different dimensions in the data set as depicted in Figure 2.3.8. The left panel allows the navigation through the hierarchy of the sensor graph by selecting the nodes being visualized. In the center, the visualization panel combines a Treemap visualization together with a colormap legend. The panel at the bottom of the win-

dow allows navigating in time and selecting the time range being visualized. We additionally augmented the timeline by visualizing the total amount of power usage over time.

Beside animation, we furthermore implemented interaction techniques like dragging the selected time range (blue rectangle in the timeline visualization) left and right causing immediate updates to the visualization. The visualization offers three interaction possibilities. The first interaction is a tooltip allowing inspecting the underlying data values invoked by mouse hovering. We directly support drill-down and roll-up operations in the Treemap visualization, allowing the analyst to keep his focus on the visualization during traversing the sensor graph. Lastly, the analyst is able to select a region in the visualization and query the system for similar time series sharing the selected behavior by means of distance or correlation calculations. Switching the visualization technique, colormap, value normalization, anomaly calculation, or the weights for the Treemap layout is possible by choosing the respective option.

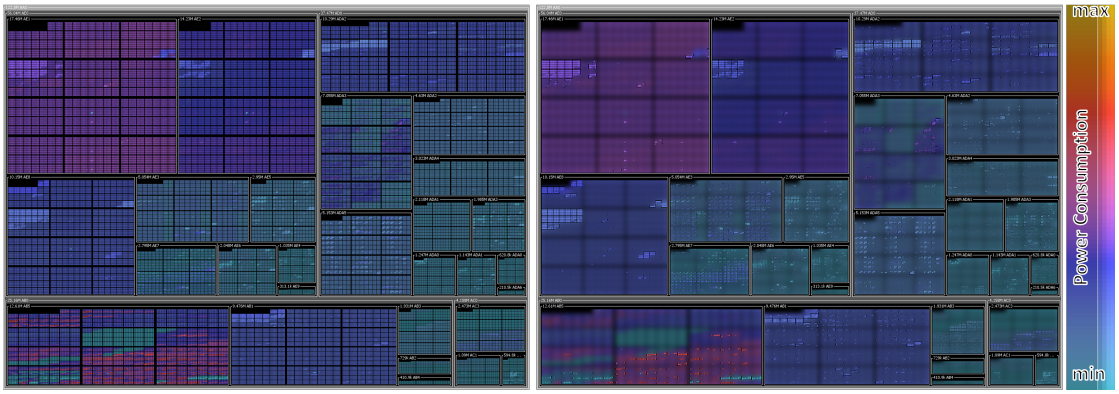


Figure 2.3.9: Overview of the power consumption data from 28 sensors during 48 weeks. Despite the huge amount of data, patterns are still clearly visible. On the right, the same dataset is presented, but with adaptive blurring highlighting unusual power consumptions. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

2.3.6 Visual Inspection of Anomalies

In this use case, the building administrator gets the information, that in February 2012 the overall power consumption and energy costs of a building was higher than expected. The investigation starts by getting an overview and some contextual information about the general energy consumption of the building. Undoubtedly, the most suitable visualization for this task is the Recursive Pattern Visualization, which can be seen in Figure 2.3.9. The blurring approach at

the right side highlights the anomalies further compared to the left figure, where we visualized anomalies only by color intensity. The resulting visualization points directly to one time series, which can be seen in Figure 2.3.10 on the right. Both, the left and the right visualization show the power consumption data beginning on 6 February 2012. Each of the bigger rectangles contains the data from one day, starting with Monday on the left. In total, there are four weeks of data visible, starting on 6th February and ending on 4th of March.

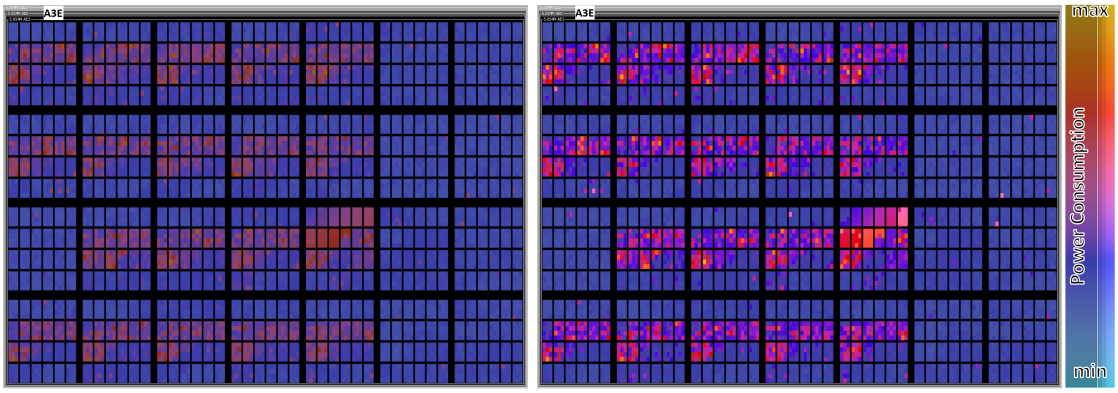


Figure 2.3.10: Power consumption measured by sensor AE3 from 6th February to 4th of March 2012. On the left, only the power consumption is visualized. On the right, the intensity is additionally reflecting the anomaly score. Due to the high intensity, an area in the fifth column of the third row stands out. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

In the visualization, there are some single, outstanding spots. Those look relatively random and last only one pixel representing a time span of five minutes. Although the color is quite intense and reddish, they are far too few and do not last long enough to have a large influence on the overall power consumption. Besides these spots, an area in the fifth column of the third row stands out. The intensity seems to increase from pixel to pixel over a long time. Having in mind, that one small black-framed rectangle of the Recursive Pattern stands for one hour, the anomaly score seems to increase over ten hours, until suddenly the anomaly score drops again. Due to the long duration of the anomaly and the intense red color, the actual energy consumption in this time frame is very high. This makes this anomaly a candidate for the cause of the higher energy costs in February.

The building administrator found an anomaly in the given time frame with the Recursive Pattern visualization. To identify potentially correlated time series, our prototype implements

a top-n time series similarity search. The query can be triggered by clicking on a part of the visualization and selecting the query area with the mouse. Afterwards, the desired similarity measure can be selected. The system supports the standard Euclidean distance and positive, negative, and unsigned Pearson Correlation for different analysis tasks. In this case, selecting the positive Pearson Correlation or the Euclidean Distance is appropriate. The result of the query can be seen in Figure 2.3.11.

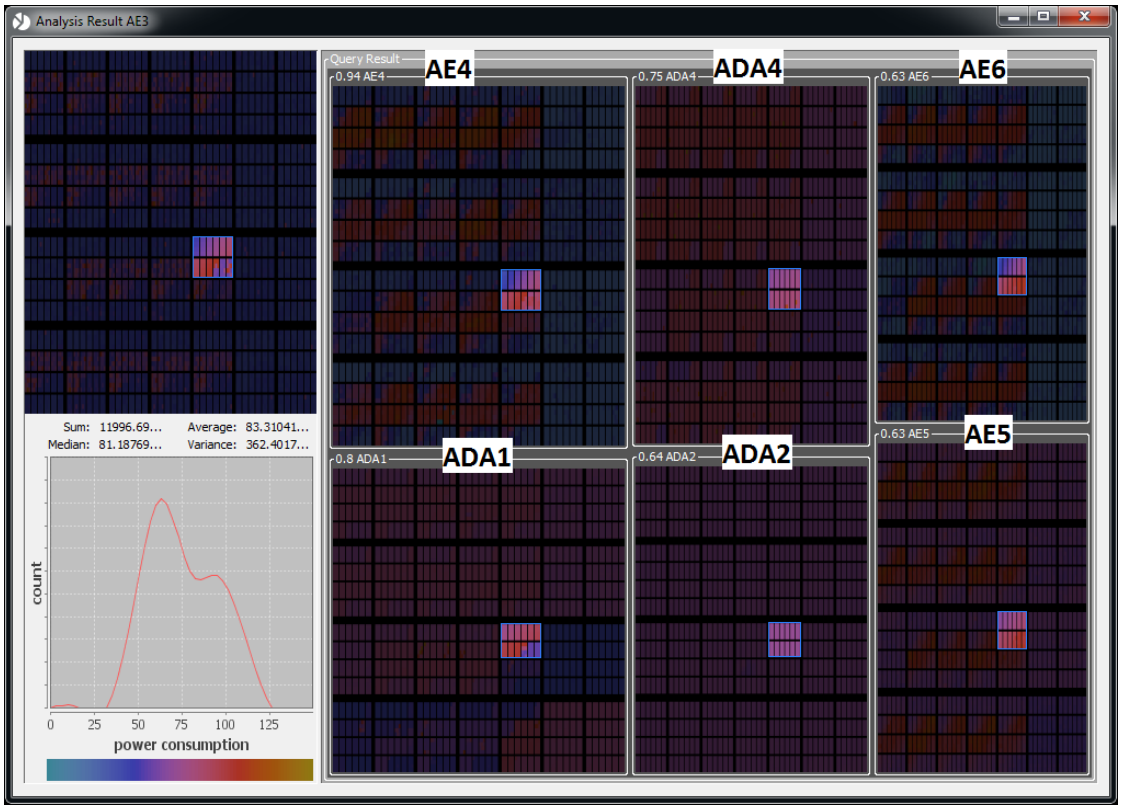


Figure 2.3.11: The time series query result window. On the top left, the query time series is displayed, on the right the top-n query results are shown. The query range is highlighted. Reprinted from [JSMK14], © 2014 Elsevier Ltd.

The query results show three very similar series: AE4, AE5, and AE6. All three sensors are part of the same subtree of the sensor hierarchy. This means they are located in the same building as sensor AE3, which logged the time series identified as anomalous by the Recursive Pattern visualization before. With this additional knowledge, the building administrator can conclude that the anomaly affected not only one, but at least four parts of the building, where the sensors

have been installed.

The quality of the conclusions drawn from the visualizations and analytical methods depends heavily on the sensor deployment. If each of the sensors monitors a single machine or office, the building administrator has a concrete subject of further examination. When they are deployed in a more general way, for example per building floor or even per building, the shown analysis allows narrowing down the investigation of power consumption to the affected units.

2.3.7 Evaluation

We showed the applicability of our proposed technique in the previous application section, but it is very important that real expert users rate our approach effective and helpful. We therefore presented our approach to the target user group in a big company. We had contact to two analysts and interviewed them first about their state-of-the-art technology. The company develops sensor networks measuring the power consumption for large buildings and is experienced with power management. The current state-of-the-art technology they are using is a visualization based on line charts. They are able to select arbitrary time frames and inspect the temporal power distribution. Further analysis steps are yet impossible to perform. In later meetings we explained our approach to the experts and afterwards let them interact with our system and investigate the time series data. We asked them to describe their typical way of analyzing data and furthermore to comment on our proposed technique by thinking aloud using our prototype. We got very valuable and interesting feedback from the experts regarding the benefits and room for improvement.

First of all, they validated the temporal patterns shown in the pixel-oriented visualization techniques with their knowledge of typical power consumption patterns. Their proof-of-concept was that the daily periodic patterns were visible at a glance, at the same time reflecting their expectations for the time series. After they found the patterns like low power consumption at nighttime and weekends they started to look for anomalies using our visual boosting techniques. At first, obvious patterns like holidays or the Christmas vacation have been found. Afterwards, less obvious patterns have been investigated. During their analysis, we asked the experts to comment on our techniques and give feedback related to visualization and analysis methods.

The first point they commented on was the helpfulness of the overview visualization in the form of the Recursive Patterns. Compared to line chart based visualizations they are very fa-

miliar with, the calendar-like representation of the power consumption was highly appreciated. Furthermore, the possibility to interactively change the visualization type helped them a lot to get familiar with the pixel-oriented techniques. The coloring of pixels was intuitive to them and they could interpret the visualization easily.

From an analysis point of view, a very interesting point was their comment on our prediction-based anomaly computation. They agreed with our definition of anomaly: "The anomalous day is likely to deviate from the daily pattern in some way." As shown above, our anomaly method is very fine grained, but to the experts a single time spot with a high anomaly score is not important. They were more interested in longer periods of unusual behavior, starting at approximately one hour duration. On the other hand the related anomaly computation method based on days was too coarse-grained for them for this kind of analyses. An aggregation of the anomaly values might help to let the analyst focus on the severe anomalies. The visualization of the anomaly scores together with the time series was mentioned very positive, especially with respect to the Recursive Pattern. The overview calendar-like visualization with intensity highlighting and adaptive blurring let them focus on the interesting spots. They had the impression that their attention was guided to the anomalies, while the unimportant, common daily patterns were pushed in the background. As soon as they found some unexpected anomalies they applied further analysis techniques.

The experts very much appreciated the possibility to select a region in the time series and query for other similar time series. When they selected a leaf in the hierarchy of time series they would look for the impacts of the anomaly on the parent nodes. The other way around, querying for anomalies on higher levels would show the root-causes for the unusual power consumption.

A possibility for improvement mentioned by them is the integration of external events into the application. Sometimes managers know in advance of extraordinary events that will cause unusual power consumptions. It should be possible to include this information whenever available and to reflect the additional events in the visualization. Overall they found the integration of different time series visualization techniques combined with an anomaly representation very helpful and wanted to integrate our techniques in their management tools.

2.3.8 Conclusion

Analyzing and interpreting unusual patterns in time series data is a very important task. In this paper, we applied novel analysis and proven visualization techniques to a system, which sup-

ports analysts finding those patterns visually. We supported the analysis process by computing anomaly scores of the given time series data with an anomaly detection algorithm which produces very fine grained results. This also allows the creation of detailed visualizations resulting in a fine grained pixel-based date representation. Furthermore, the algorithm is very efficient in terms of required computing power, because it does not require expensive transformations nor does it rely on elaborated analyses of the time series data.

Having the anomaly scores, different visualizations can be used to get deep insight into the time series and the anomaly scores, depending on the task to fulfill. Recursive Patterns create overviews of large time spans and large amounts of data. Spiral views provide the possibility to quickly detect and analyze periodic patterns. If a more common visualization was wanted, the classical line charts would be also available for further investigations of the data set.

The double encoding of time series values and anomaly scores is solved in different ways. The novel adaptive blurring, which generates a focus and a context area by blurring the visualization according to the anomaly scores, guides the analyst directly to interesting spots of the visualization. This makes the technique a particular advantage in overview visualizations, where irrelevant areas of the time series are losing their level of detail by a strong blur, whereas interesting, high anomalous areas are clearly visible and attract the focus of the human eye. To support the display of multiple visualizations, the well-known Treemap approach is extended by layouts based on space efficiency and specific visual properties of the visualization.

The use case of power consumption data shows the applicability of the methods shown in this paper. The general nature of the analysis and visualization methods makes it possible to apply these techniques to time series not only from the application domain of power consumption data. In the future, we want to integrate external knowledge like known events influencing the time series like weather information. It would be also interesting to automatically determine the visualization method, colormap, and possible enhancements like the adaptive blurring based on the displayed data.

Listen within yourself and look into the infinitude of Space and Time. There can be heard the songs of the Constellations, the voices of the Numbers, and the harmonies of the Spheres.

Hermes Trismegistos

3

Enhancing Visualizations for Geospatial Data

Contents

Geospatial data are a very rich and valuable source for research questions. Geospatial research questions are often very application-driven, with subject matter experts either having hypotheses of the hidden movement patterns or being just overwhelmed by the amount and resolution of data collected. Besides application needs, geospatial data are challenging from a visualization and Visual Analytics perspective. Advances in technology allow capturing more and more movement data in higher temporal and spatial resolution. We nowadays need to support the analysts during their explorative analysis during several steps:

Hypotheses building

Without any prior knowledge of the collected geospatial data, it is often impossible to define hypotheses to investigate. Geospatial visual analysis methods should help in the very first explorative investigation steps. Following Shneiderman's visual information seeking mantra [Shn96], giving first an insightful overview to the user is crucial for a successful analysis process. The challenge is to deal with a possibly infinite amount of input data and a very limited visualization space. Problems as visual clutter and overplotting occur easily and often simple mappings from data to screen space are not sufficient. We propose in this section visual methods for point- and line-based representations actively reducing overplotting issues by replacement, aggregation, and abstraction.

Explaining observed geospatial patterns

Often geospatial patterns occur and although we may be able to identify and visualize them, we are not per se able to explain them. Visual support and guidance to explaining features and

variables are needed, in order to support the analyst generating insights. We usually derive attributes and additional context data and visually present them to the user. The Visual Analytics methods employed should take existing domain knowledge into account to ensure an effective analysis process. Our techniques described in this work cover both the presentation of additional, external information as land-use categories and integrating the analyst's domain knowledge during the analysis process.

Exploring and analyzing reoccurring patterns

Once interesting movement behaviors are identified, analysts may want to look for similar patterns. A good visual analysis systems should be able to compare this set of situations with automatically deriving and presenting similarities and differences. Automatically detecting why a certain situation is interesting to the analyst and finding based on this information similar situations is quite challenging. Usually, relevance feedback, as introduced for example by Rocchio in [Roc71], will be employed to improve the results. Giving some kind of formal or informal description to the system, why a situation is important is crucial for good results. Basically, this feedback loop is a step to transfer domain knowledge from the subject matter expert to the Visual Analytics system. We will especially focus on the detection of certain kind of situations and the integration of user feedback in the next chapter dealing with soccer data.

We will discuss in this chapter techniques dealing with the two most important types of geospatial data related to movement. Movement can be either seen as a sequence of discrete sampling points or as a sequence of linear segments approximating the original movement. Point-based representation are usually applied when the overall density distribution is of interest or when analysts are investigating additional measurements (e.g., weather information or health parameters) recorded synchronously with the geospatial tracking. Line-based visual representations are applied when the sequence of locations, at which the moving object was tracked, are of interest. Usually, line-based visualizations help to identify the direction of movement (for example, by tapered line segments) and similar movement behavior within groups of movers. As lines need more screen pixels than points, they are visually more salient and can potentially encode additional information (e.g., speed by the length of segments in the case of regular sampling). At the same time, lines introduce more overplotting and reducing the amount of overplotting is more challenging for lines than for points.

Consequently, we will discuss in this chapter methods enhancing the visualization of point-based and line-based geospatial data. We see scatter plots as a very related technique visualizing geospatial point-based data. With scatter plots being widely used, we decided to show general applicability of our approach and use geospatial data sets only as one of many possible application scenarios. Nevertheless, the techniques described in the next section with focus on general scatter plots are easily transferable to the geospatial domains and point-based visualizations. The line simplification algorithms described in this section are not tailored to a specific application domain, although the use cases are often in the area of animal movement. Animal movement has the advantage of lacking any privacy issues and is complex enough to provoke research.

3.1 Enhanced Scatter Plots for Point-based Visualizations

The following section is based on the following publication[1]:

> **Enhancing Scatter Plots Using Ellipsoid Pixel Placement and Shading**
>
> *H. Janetzko, M. C. Hao, S. Mittelstädt, U. Dayal, D. A. Keim.*
>
> *Proceedings of the 46th Annual Hawaii International Conference on System Sciences, pp. 1522–1531, 2013.* [JHM+13]

3.1.1 Preface

Exploring two-dimensional relationships and correlations is the key feature of scatter plots. Scatter plots are therefore often used because of their wide applicability and intuitiveness. Unfortunately, one severe drawback exists impairing the effectiveness of scatter plots: overplotting will occur in dense data regions resulting in a significant number of points not shown to the user. Two reasons exists, why overplotting occurs. Either the data points share the very same coordinates or the screen resolution is not sufficient to place the points on distinct pixel coordinates. Without overplotting, we can even colorize the points in a scatter plot to visualize a third

[1] The idea of using ellipses for pixel placement and applying lighting was developed by myself. Sebastian Mittelstädt implemented the techniques into an existing prototype previously developed by myself. Ming Hao, Umeshwar Dayal, and Daniel Keim helped with fruitful discussions and advices.

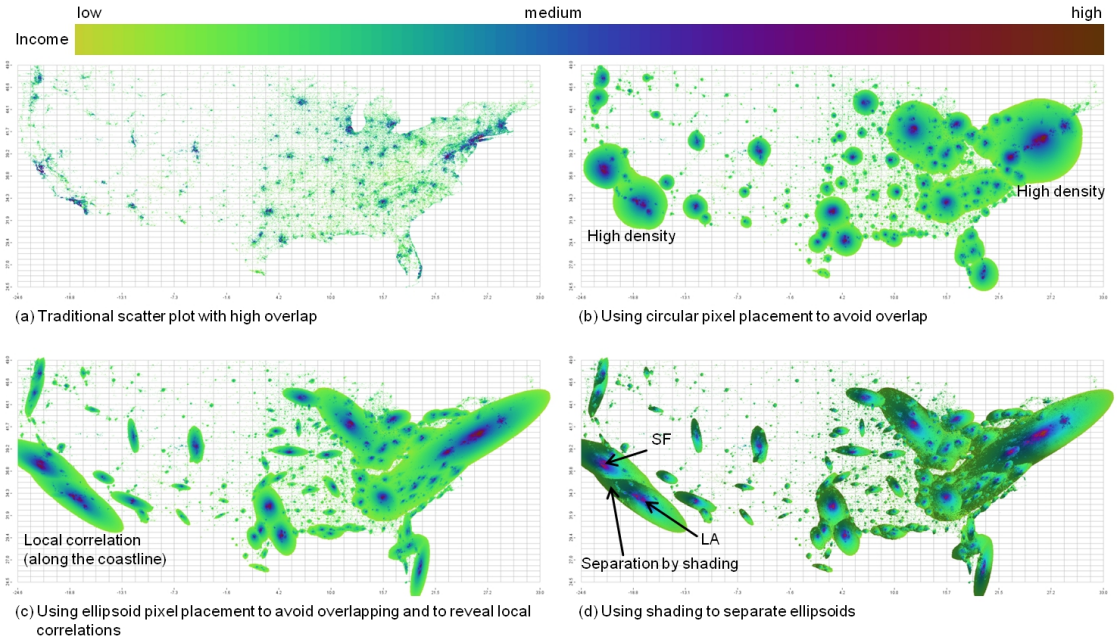


Figure 3.1.1: In this figure, we compare four different scatter plots for US Census data. Figure (a) depicts a traditional scatter plot with a high overplotting degree in densely populated areas. The other three figures show results of (b) circular pixel placement, (c) ellipsoid pixel placement without shading, and (d) ellipsoid pixel placement with shading. Note, that there are no overplotting points in figures (b), (c), and (d). (x-axis: Longitude, y-axis: Latitude, color: Median household income) Reprinted from [JHM+13], © 2013 IEEE.

dimension.

We exemplify the drawback of scatter plots in Figure 3.1.1 (a). The traditional scatter plot should display 333,488 data points, but due to overplotting the scatter plot is only able to visualize less than ten thousand data points. In a previous work, we described *Generalized Scatter Plots* [KHD+09] and proposed user-controllable solutions to reduce the amount of overplotting. Figure 3.1.1 (b) shows the result of circular pixel placement being a technique described in [KHD+09]. We will explain this approach in more details in later paragraphs of this section. We enhanced the circular pixel placement by applying ellipsoid pixel placement reflecting local correlation patterns depicted in Figure 3.1.1 (c). In order to visually separate nearby ellipses, we applied lighting to the scatter plot resulting in Figure 3.1.1 (d).

We present in the following sections a novel approach to enhance scatter plots in two ways. We apply a ellipsoid pixel placement removing overplotting in the scatter plot and simultane-

ously expressing local correlation patters. Furthermore, we added illumination to the visualization in order to enable a better separation of nearby density clusters. Additionally, lighting enable the analyst to determine the original position of a relocated data point. We will first discuss existing approaches in Section 3.1.2 followed by a short introduction to the previous Generalized Scatter Plots technique, in Section 3.1.3. We present our novel techniques in Section 3.1.4 and discuss possible lighting options in Section 3.1.5. Our new techniques are applied in three different application scenarios showing the wide applicability.

3.1.2 Related Work

Overplotting in scatter plots is not a new challenge and there were many techniques developed solving overplotting issues. The first set of techniques computes and visualizes the data density. Cleveland presented in 1984 for example a glyph-based representation depicting the data density called sunflowers [CM84]. Glyphs may introduce other visual artifacts, such as overplotting of glyphs or binning artifacts. Semi-transparency of data points is used to convey a notion of density and is described in [WWR+06]. In most cases using transparency will help analysts to investigate the density distribution. A severe drawback of transparency is the limited readability of resulting density visualizations. Transparency is not correlating linearly to the number of points painted at one single position. Setting the transparency value to the best one is very challenging and depends highly on the data set and the task. Other density visualizations as heatmaps or density contour maps are often better to depict the density distribution. Such density visualizations display aggregated information, but not the raw point data as traditional scatter plots. A mixture of density visualizations and scatter plots are HexBin scatter plots [CLNL87, Hex15]. Hexbin scatter plots partition the data into hexagonal bins and count the number of data points per bin. The density counts can be visualized for instance by different brightness levels, as done in the statistical toolkit R. Later techniques proposed by Bowman and Azzalini [BA04, BA03] built smooth contour scatter plots showing overlaps with different shades. Continuous scatter plots invented by Bachthaler and Weiskopf [BW08] derive from the discrete input data a continuous model. Continuous scatter plots do not suffer from overplotting points as they are aggregated when building the model, but the analyst loses the notion of how many data points lead to a local pattern. Anti-aliasing and a greyscale representation is used by the Information Mural in order to cope with overplotting. Jerding and Stasko [JS98] focused in the Information Mural especially on cases where the number of data points exceeds

the number of available pixels. Mayorga and Gleicher discuss in [MG13] a technique combining density aggregation with point-based representations in scatter plots. The authors apply perception-based color blending for density surfaces and sampling for visualizing a subset of the original data points.

The second set of techniques distorts and transforms the display space in order to reduce overplotting. Interacting with small screens suffers from the very same overplotting issues as scatter plots. Büring et al. [BGR06] proposed two interaction techniques: a geometric-semantic zoom supporting transitions between overview and detail and in order to contextualize the current viewport the authors proposed a fisheye distortion. Other distortion techniques such as cartograms [KNPS02] or HistoScale [KPS+03] can be applied as well. We developed previously a technique called Generalized Scatter Plots [KHD+09] allowing analysts to interactively control the amount of distortion and pixel placement. We will describe this approach in more detail in Section 3.1.3. Another technique actively reducing the amount of overplotting was introduced by Trutschl et al. [TGC03]. The authors present a SOM-based approach replacing overplotting data points according to the points' similarity. A very simple, row- or column-based pixel placement algorithm is presented by Aris et al. in [AS07]. This approach will introduce visual artifacts resulting from the row- or column-based replacement of data points.

In this paper, we are adapting the pixel placement to local trends in the data set. We highlight therefore two example approaches dealing with detecting and visualizing local trends in scatter plots. In order to visualize trends in multi-dimensional data Robertson et al. [RFF+08] propose to use animation and Small Multiples. Visualizing local correlation patterns by a flow-based visual representation is presented by Yu-Hsuan Chan et al. in [CCM10]. The authors visualize both the direction and the strength of the local correlation.

In Figure 3.1.2, we compare six different approaches being related to our proposed technique. We use a data set resulting from a telephone conference system, with a total of 37,787 records. We relate the length of the calls (x-axis) to the charge of the respective call (y-axis). As depicted in Figure 3.1.2 a), the traditional scatter plot suffers from overplotting, but still reveals some linear relations between length and charge of a phone call. According to the data distribution, we also applied logarithmic scaling to both axes simultaneously resulting in Figure 3.1.2 b). We applied a density-equalizing distortion to the data set in Figure 3.1.2 c) enlarging the space for dense areas and shrinking the display space for sparse areas. The very same distortion is applied to all subsequent visualizations, for a fair comparison, because our new technique is

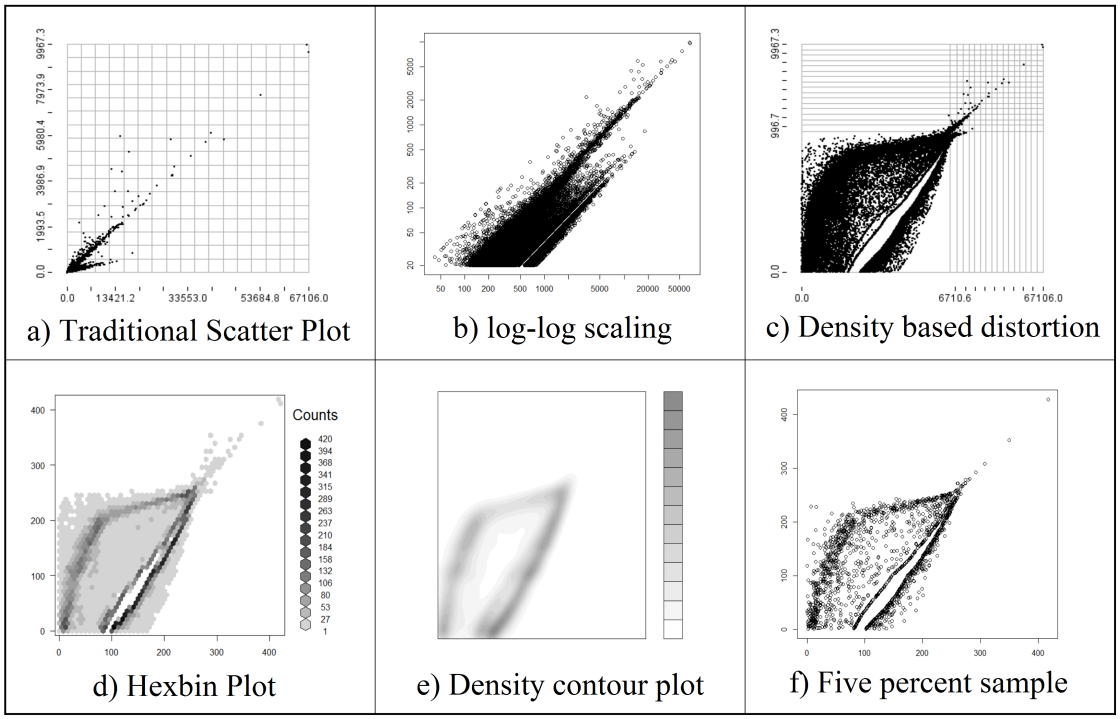


Figure 3.1.2: Six different existing techniques visualizing the same telephone conference data set. The amount of distortion of figures c) to f) is the same as in the application Section 3.1.6. (x-axis: duration of call, y-axis: charge for call) Reprinted from [JHM+13], © 2013 IEEE.

applied to the equally distorted data set in Figure 3.1.9. Density visualizing methods such as HexBin scatter plots, Figure 3.1.2 d), or kernel density estimations [BA04, BA03], shown in Figure 3.1.2 e), do not suffer from overplotting as they do not display the original data points. We performed a five-percent sampling in Figure 3.1.2 f). Sampling will reduce the amount of overplotting, but low-density patterns might disappear.

Applying lighting to visualizations is not new and has been already proposed in various works. Cushion Treemaps [VWVdW99] for example apply lighting to each treemap cell adding visual clues to distinguish neighboring cells. Lighting has been also applied to aggregated trajectories by Willems et al. [WVDWVW09b]. This method uses bump mapping related techniques in the same way as we do. Our approach can be seen very related to a three-dimensional representation of a kernel-density estimation [Par62]. The main difference between kernel-density estimation and our approach is that we do not aggregate the data at all. The result of our technique is a two-dimensional pixel-based visualization, with every data point represented by one

pixel. Consequently, we can encode a third variable by color and interact with each single data item.

3.1.3 Generalized Scatter Plots

We introduced in some previous work *Generalized Scatter Plots* [KHD+09] enabling the analyst to interactively control the amount of distortion and pixel placement applied. An example result is depicted in Figure 3.1.1 (b). As the ellipsoid pixel placement and lighting techniques build upon our previous work, we will briefly introduce the Generalized Scatter Plots in this section.

The basic assumption of this technique is that a fully distorted view without any overplotting may not be the best possible view to the data set according to some optimality criteria. We introduced two criteria, namely the displacement error (points should not be moved too far from their origin) and the overplotting error (no points should overplot each other). Obviously both criteria are not possible to satisfy simultaneously. We consequently describe techniques being user controllable allowing arbitrary stages of distortion and pixel placement. The range covered starts from traditional scatter plots to fully distorted and pixel placed visualizations without any overplotting but with circular visual artifacts.

Technique

We employ two techniques to reduce the amount of overplotting in scatter plots. The first technique we implemented is a density equalizing distortion enlarging dense areas and shrinking areas with low density. Consequently, the distortion will already reduce the amount of overplotting. In detail, we bin the data space independently in both dimensions and count the number of points per bin. As shown in Figure 3.1.3, the relative density of each bin is used to determine the corresponding bin size in image space.

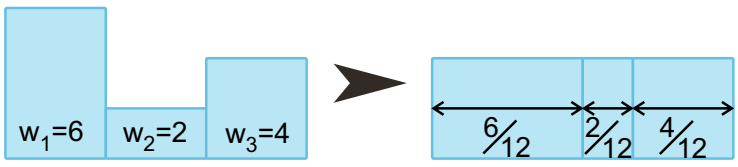


Figure 3.1.3: Example of a one-dimensional density equalizing distortion based on relative density. Reprinted from [JHM+13], © 2013 IEEE.

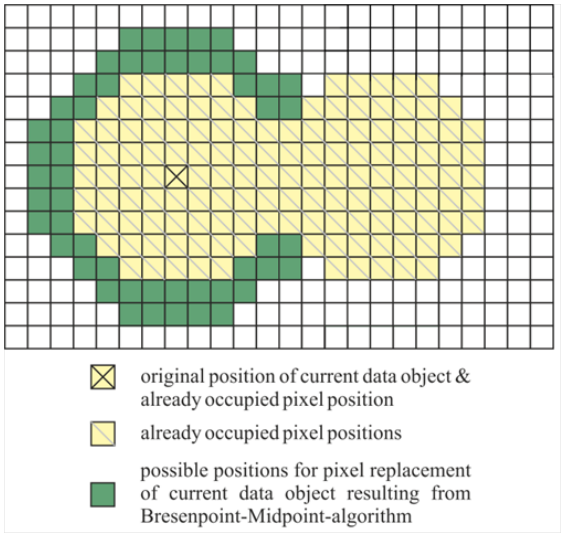


Figure 3.1.4: The new position for an overplotting data point is determined in a circular fashion. The nearest empty pixel is chosen from the green candidate set. Reprinted from [JHM+13], © 2013 IEEE.

Additional to the density equalizing distortion, we introduced a circular pixel placement algorithm relocation all overplotting data points to a nearby empty pixel position. In Figure 3.1.4, we show an example relocating step of our algorithm. Our algorithm uses a sorted input lists of all data points, with the points usually being sorted by the third dimension – the dimension being represented by color. We iterate over the sorted lists of points and for each data point we check whether the original screen position is empty. If the pixel position is empty, we will place the data point at this location. In the other case, we will relocate the data point to next free position. In order to find the next free position, we compute a circular area around the original location (green area in Figure 3.1.4). Out of this candidate set of possible positions, we chose the nearest empty pixel. Following this iterative algorithm, we can assure that there are no overplotting data points, if the number of pixel exceeds the number of data points. More details of the pixel placement algorithm with respect to the parametrization allowing intermediate states between data-induced overplotting and no overplotting can be found in the original paper [JHM+13].

Merits and Limits

Without overplotting in scatter plots, we are not only able to show all data points but we can apply coloring to the data points expressing a third dimension. Furthermore, we assessed the quality of the resulting visualization and could show that a combination of pixel placement and a semi-distorted view is the best with respect to overplotting and displacement error.

However, Generalized Scatter Plots do not take the local distribution of data points into account and always introduce visual artifacts by the circular pixel placement. The very salient circles in the resulting visualizations are for instance visible in Figure 3.1.1 (b). Scatter plots are employed to investigate the relationship of two variables. It is consequently not very beneficial to visualize the data in a way that local patterns of zero correlation are induced.

Another drawback of the method is that the original location of data points is lost, as points are moved to the nearest empty position. This is especially bad in cases, when there are to circles visually intersecting, as it can be seen in Figure 3.1.4. In the intersection area, it is not obvious to which origin a data point belongs to.

3.1.4 Enhancing Generalized Scatter Plots

As Generalized Scatter Plots have some limits, we first improved the pixel placement algorithm and developed a new ellipsoid pixel placement. The local correlations will be reflected by the rotation and the width-to-height ratio of the ellipses. Consequently, the pixel placement result will retain local data distribution patterns. We will describe the approach in more detail in the next section.

We furthermore propose a solution to visualize the original position of a relocated data point. We followed the idea of Bump Mapping [Bli78] and introduced an artificial 2.5D impression to a two-dimensional surface. The lighted surface will give visual clues to the analyst at which original location a point was before the pixel placement. This approach will be presented in detail after the description of the ellipsoid pixel placement.

Ellipsoid Pixel Placement

As already mentioned above, scatter plots are employed to reveal global and local trends and correlations. Applying circular pixel placement will hide local patterns decreasing the effectiveness of scatter plots. Exchanging the circular pixel placement by an ellipsoid replacement allows us to visualize local correlations. We can show both the strength and the direction of the

correlation by rotating the ellipse and adjusting the aspect ratio according to the correlation strength.

We have to define locality before we can compute local correlations. We therefore partition the input data set into disjoint groups and compute for each group the linear correlation. The proper partitioning is crucial, because the local correlation may vary in the data set. A simplistic, straightforward method is a partitioning by a regular grid. But depending on how many grid cells are chosen the results may look very different. Consequently, regular grid could result in new visual artifacts not being data supported. As we wanted to reflect the arbitrary data distribution, we decided to employ a partitioning clustering technique and compute the correlation per cluster. We use OPTICS [ABKS99] in our prototype, in order to partition the data set. As OPTICS just orders the points without already assigning cluster IDs, we support the analysis by visualizing the reachability and core distances. The analyst can interactively set the epsilon threshold and gets immediate feedback of the resulting clustering.

After we partitioned the data set according to the data distribution, our next steps involves the computation of a correlation measure. We calculate both, the correlation strength and the correlation direction, by applying Principal Component Analysis (PCA) to each cluster. The input for the PCA is a 2×2 covariance matrix. The PCA will return in our case the two most dominant correlation directions in terms of two eigenvectors and the correlation strengths in terms of eigenvalues.

The most dominant correlation direction is depicted in our example in Figure 3.1.5 (b). The first half axis of the ellipse is parallel to the most prominent correlation direction, as shown in Figure 3.1.5 (c). The ratio of the two eigenvalues corresponds to the strength of the correlation. We adjust consequently the ratio of width and height of the ellipse according to the ratio of the two eigenvalues. In order to avoid very narrow ellipses in perfectly correlated cases, we restrict ourself to a minimal ratio of 1:4.

As soon as we have determined the proper shape and rotation of the ellipse based on the local correlation, we are able to compute the pixels of the ellipse accordingly. We apply an approach based on Bresenham midpoint algorithm for ellipses [Ake84] and rotate the resulting pixel positions according to the desired rotation. These pixel positions are then used to replace an overplotting data item.

We show the overall pixel placement on an abstract level in Algorithm 3.1.1. The conditions

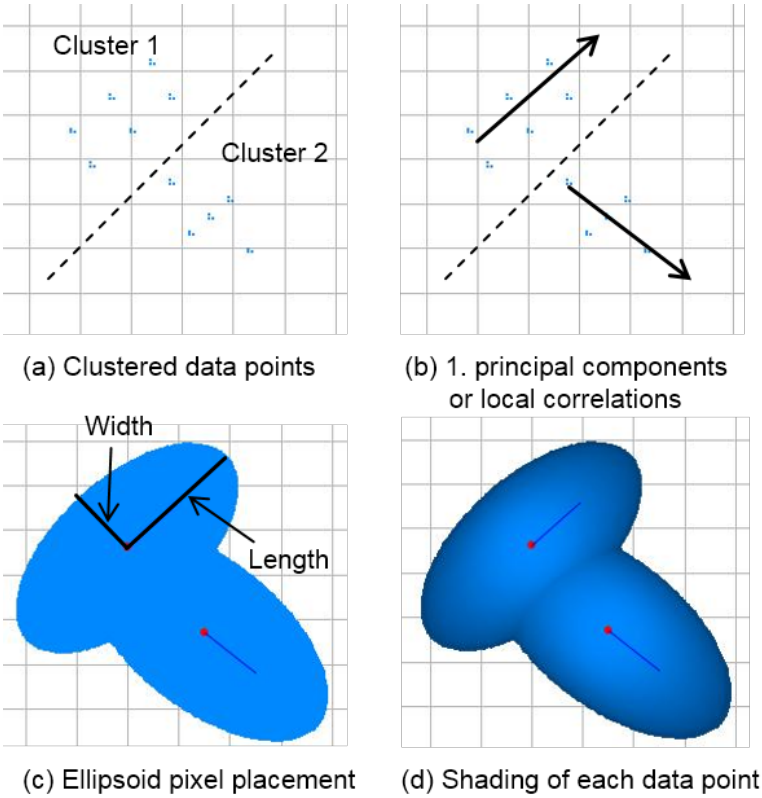


Figure 3.1.5: Starting from a set of points, we perform partitioning clustering to separate the data set (a). We compute for each cluster the most prominent correlation direction (b) and use the correlation direction and strength to determine a corresponding ellipsoid pixel placement result (c). Lastly, we apply shading according to an illumination model (d). Reprinted from [JHM+13], © 2013 IEEE.

marked by • do not only allow to remove overplotting at all, but do also allow for intermediate stages between overplotting and no overplotting. The line marked by •• covers the call of the ellipsoid Bresenham algorithm and some counter variables. With each call of this line for one specific location the algorithm must ensure that a new pixel position will be returned. Consequently, the width and length of the ellipse will be increased gradually as soon as all border pixels of a ellipse have been returned. From a technical perspective, we store for each pixel position the last returned pixel together with the parameters of the last used ellipse.

Algorithm 3.1.1: Abstract pixel placement algorithm

```
Input: Ordered list of all data items
foreach DataItem cur in allDataItems do
•   if cur can be placed at original position then
        assign original position to cur
    else
        repeat
••          get next ellipsoid pixel position for original location of cur
•           if ellipsoid pixel position can be used for cur then
                assign ellipsoid pixel position to cur
        until new position for cur found
```

Combining Shading and Pixel Placement

As pixel placement techniques replace data points to some nearby locations, the origin of a data item may be obscured. Especially, when the origin is ambiguous, for example in the intersection area of Figure 3.1.4, it is impossible to infer the original coordinated of a data point. We therefore visually enhance the result of our pixel placement algorithm by applying a technique based on Bump Mapping as depicted in Figure 3.1.5 (d).

The basic idea is that all points originating from the same location belong to a small hill. This hill can be used to defer normal vectors used for illumination purposes. We show our approach schematically in Figure 3.1.6. The shading allows both, the visual separation of nearby ellipses and localizing the origin of a data point. The normal vectors are the input illuminating the scene by Phong shading [Pho75]. We compute the amount of light being reflected by the data point according to light direction and normal vector and blend this with the point's color. Lighting might decrease the effectiveness of coloring the data points. We consequently enable the user to control the strength of the illumination to focus on the third dimension's value or to focus on the origins of the data points.

The computation of the normal vectors is closely related to the result of the prior pixel placement. The distance of the point's final location to its origin is weighted by the half-axis if the corresponding ellipse. The higher the distance is the more parallel to the *xy*-plane the normal vectors become. The normal vectors and their correspondence to their relative location can be seen in Figure 3.1.6.

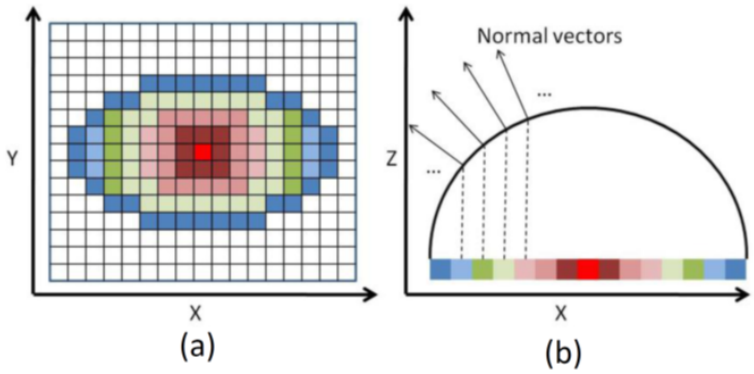


Figure 3.1.6: The ellipses resulting from our pixel placement approach are seen as hills. Normal vectors are computed correspondingly (right). Reprinted from [JHM+13], © 2013 IEEE.

Applying illumination to scatter plots allows us to adjust lighting to the analyst's needs. We can influence both the number of light sources and the respective light direction. In our prototype, we propose three different settings reflecting the most important scenarios and additionally implemented manual control over the lighting conditions. The different settings can be inspected in Figure 3.1.7. In most application scenarios, lighting orthogonal to the main correlation vector is sufficiently expressive. Nevertheless, the user can choose the best suiting lighting condition.

3.1.5 Discussion

We developed the ellipsoid pixel placement technique to enhance the visual salience of correlation patterns. We compare in Figure 3.1.7 our new approach to circular pixel placement being a representative for methods not taking data-inherent patterns into account. The data set visualized is artificially generated and contains several clusters with random position, size, and correlation. The circular pixel placement, Figure 3.1.7 a), is performing worse with respect to correlation visibility compared to the ellipsoid pixel placement shown in Figure 3.1.7 b).

Besides enhancing the pixel placement algorithm, applying shading to a scatter plot abstracts further from the original view. But applying lighting comes with the advantage that we can differentiate nearby clusters better. While we lose some information about the values encoded by color, we are able to hint to the original position of a data point. We visually compared the most promising lighting options and discuss them below. In Figure 3.1.7 c), we applied one light source for each cluster parallel to the main correlation. Each cluster is illuminated orthogonal

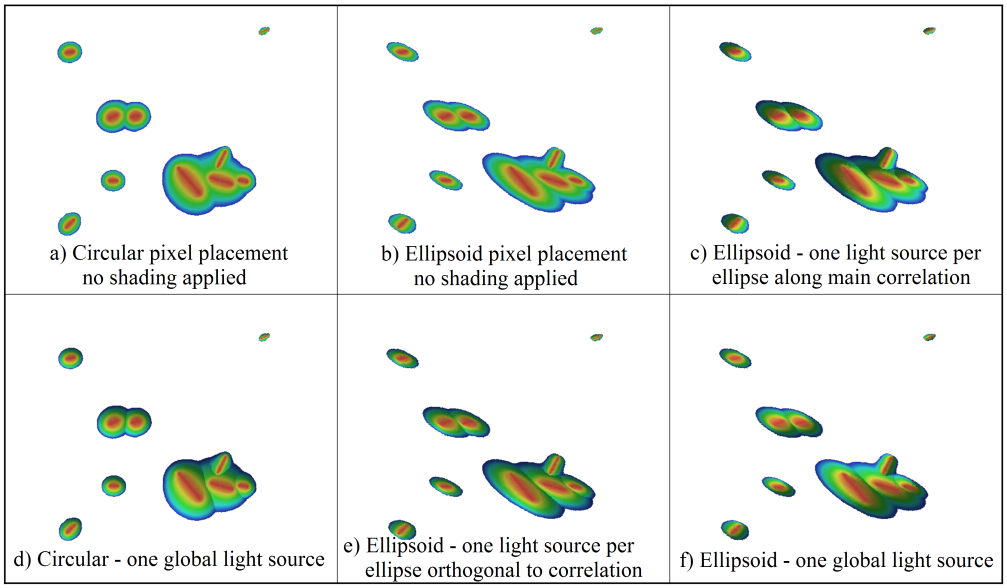


Figure 3.1.7: Visual comparison of possible illumination variants. All figures visualize the same, artificially created data set without overplotting. Reprinted from [JHM+13], © 2013 IEEE.

to the main correlation in Figure 3.1.7 e). Our last shown option is one global light source orthogonal to the main correlation of the whole data set, in Figure 3.1.7 f).

When we started developing our technique, we expected that option c) would result in the best visualizations, since it emphasizes the direction of the local correlation. However, in our informal evaluation options e) and f) performed significantly better. The visibility of the different local correlations depends on the respective applications but seems to be best in e) and f). We added for completeness the results of the circular pixel placement in Figure 3.1.7 a) and the correspondent illuminated result d).

Consequently, we illuminate the scene by default per-cluster and orthogonal to the correlation, while enabling the analyst to adjust the lighting to his needs. Furthermore, the strength of the lightning can be controlled or even switched off. We designed our colormap in such way that it is hue and not intensity based. Because of the illumination process, intensity based colormaps could result in data points being identically colored though different in the encoded third dimension's value.

3.1.6 Applications

We applied our ellipsoid pixel placement algorithm combined with the shading technique to several real-world data sets. We will first introduce a financial scenario followed by a telephone service usage analysis. Lastly, we discuss a geospatial use case and investigate the visualization results of a census of the United States of America.

Financial Analysis

Investors and financial analysts are interested in developments of funds and the best time for their purchases and sales. Based on historic data, investors try to find the best funds minimizing their risks and maximizing their profit. Risky funds are typically determined by a high standard variation in the respective price time series. We applied our techniques to a data set of approximately 130,000 American funds collected over 15 years with focus on a performance-risk analysis. The performance of each fund is computed on a yearly basis resulting in 14 one-year intervals. The data set contains therefore for each of the 130,000 funds 14 data points representing the performance for one year. In Figure 3.1.8, we visualize the impact of the risk (x-axis) on the performance (y-axis). The color of each data point represents the respective one-year interval applying a rainbow colormap. We chose a rainbow colormap supporting our lighting technique and enabling the analyst to distinguish between different intervals in an ordinal way.

We compare in Figure 3.1.8 circular pixel placement (a) and ellipsoid pixel placement (b) both with illumination applied. In both figures, in the middle of the data set is a sparse area visible. As known from financial research and also visible in our visualization, increasing risks have either strong positive or negative impact on the performance of financial funds. Exploring the data set further, we highlighted four high-density clusters in Figure 3.1.8 (b). A very prominent cluster is C1 described by high risks and very low performance. Cluster C1 corresponds to the "Dotcom" crisis in 2000 and the global financial crisis in 2007 to 2009 and shows a strong negative correlation of risk and performance. Cluster C4 shows quite the opposite behavior: directly after the global financial crisis in 2009, the funds recovered in 2010 quickly. The other funds of 2010 are contained in cluster C2, showing low risks and medium performance. The years 2005 to 2007 are shown in cluster C3. These years were succeeding the "Dotcom" crisis from which the market slowly recovered.

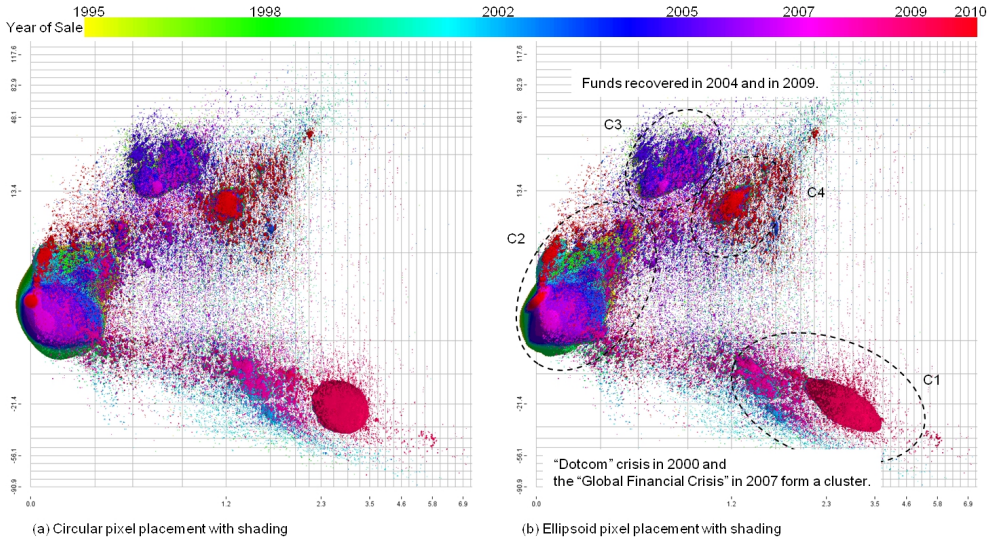


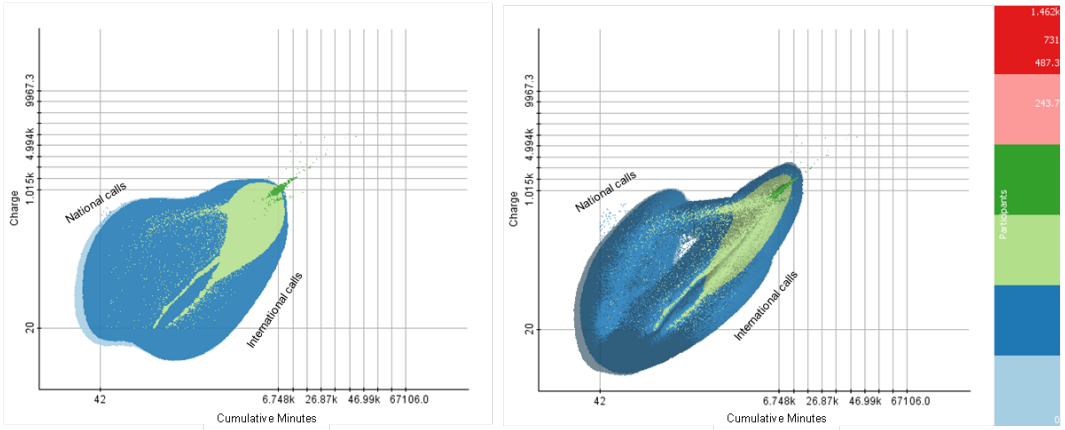
Figure 3.1.8: Risk-performance analysis of 130.000 American financial funds from 1995 to 2010. The risk of each fund is represented by the x-axis and the respective performance is shown on the y-axis. The color encodes the time of a one-year investment period, for example, red represents purchases in beginning of 2010 and sales at the end of 2010. Reprinted from [JHM+13], © 2013 IEEE.

Telephone Service Usage Analysis

We investigate in this application a telephone service usage data set with 37,787 record entries. The set was collected by IT service managers and is analyzed with respect to correlations and usage patterns. We analyze the charge for a call (x-axis), the duration of a call (y-axis), and the respective number of telephone conference participants (color) in Figure 3.1.9. For comparison reasons, we applied the same amount of distortion as in Figure 3.1.2.

In both pixel placements variants, the overall call distribution can be seen. Nevertheless, the circular pixel placement in Figure 3.1.9 (a) visualizes less local patterns in dense areas compared to the ellipsoid pixel placement (b). The ellipsoid pixel placement is able to partition the large set of data points into two different phone rates representing national and international calls. Furthermore, our illumination allows to see the origin of a data point and to visually assign data points to a phone rate.

The interaction on a data point level is enabled by our pixel placement allowing us to inspect meta-data, for example provider or timestamps. Based on our analysis, we could derive the following insights:



(a) 60 % distortion , circular pixel placement

(b) 60 % distortion , ellipsoid pixel placement

Figure 3.1.9: Visual analysis of a telephone conference usage data set consisting of 37,787 records. We applied in both visualizations the same amount of distortion, but compared the previous pixel placement algorithm with our novel techniques described above. The new ellipsoid pixel placement is able to split the high-density patterns in the lower left, while they are merged in the circular pixel placement algorithm. Reprinted from [JHM+13], © 2013 IEEE.

1. There is a variety in charges resulting from different providers and the locality of the conference call. The national calls represented by the left curve are the most expensive calls per minute and occur quite often. Surprisingly, the international calls are less expensive than the national ones. The slope of the national calls decreases with increasing call length seemingly resulting from a quantity discount.

2. The date points of the lower right patterns represent the international calls. The are two green linear patterns visible resulting from two different service providers, namely AT&T and Sprint.

3. Comparing the national and international calls, there seems to be a more clear rate structure for international calls. The distribution of national calls varies stronger and results from external influence factors, as for instance time of the day.

4. Our novel ellipsoid pixel placement enhances the visual salience of local patterns and the illumination separates nearby high-density point clusters. Although both Figures 3.1.9 (a) and (b) result from the same amount of distortion, the merged clusters of (a) are better separated in (b).

U.S. Census Data Analysis

We round up our presentation of application scenarios with a geospatial data set. We analyze census data of the United States shown in Figure 3.1.1. The census data set contains 333.488 entries and was collected in 1999. We visualize the medium household income with 500 households being aggregated to one single data point. In our visualizations, the x-axis and y-axis represent longitude and latitude and coloring encodes the medium household income. To ensure the compatibility with our illumination technique, we again use a hue-based colormap. The traditional scatter plot depicted in Figure 3.1.1 (a) suffers from the unequal population density distribution. The empty area in the Midwest cannot be used for visualization purposes, whereas the coastal areas have high degree of overplotting. We removed all overplotting by the circular pixel placement in Figure 3.1.1 (b) and by ellipsoid pixel placement in (c). We applied additional illumination to the results of our ellipsoid pixel placement in Figure 3.1.1 (d). Comparing the different visualizations, it is obvious that the traditional scatter plot shows very few data points. Both pixel placement algorithms enable analysts to see the whole data set. The circular pixel placement introduces visual artifacts in terms of circular shapes and hides local patterns. The ellipsoid pixel placement represents local patterns better and shows for instance the coastal line. Applying shading enhances the visualization even more and helps to separate nearby population centers.

3.1.7 Conclusion

We presented in this section two enhancements for scatter plots enabling a overplotting-free visual representation. By analyzing local correlations in the data set and using this information to adjust the pixel placement process, we are able to enhance the visual salience of local patterns. Furthermore, we added illumination to the pixel placement result in order to better separate visually merged clusters. We support different lighting approaches based on the respective analyst's needs. Our techniques have been applied to three different application domains showing the wide applicability of our approach. We plan as one of our next steps to further improve the pixel placement and adjust the used shapes to the local point distribution.

3.2 Reducing Overplotting for Line-Based Visualizations

This section is based on and partly cites the two following publications. The first publication, SimpliFly, proposes and discusses several methods to simplify and enhance trajectories[2]. The second publication complements the simplification methods of our SimpliFly paper by clustering dense movement patterns and deriving a graph-based visualization[3].

SimpliFly: A Methodology for Simplification and Thematic Enhancement of Trajectories

K. Vrotsou, H. Janetzko, C. Navarra, G. Fuchs, D. Spretke, F. Mansmann, N. Andrienko, and G.Andrienko.
IEEE Transactions on Visualization and Computer Graphics, 2015. [VJN+15]

Visual Abstraction of Complex Motion Patterns

H. Janetzko, D. Jäckle, O. Deussen, and D. A. Keim.
SPIE 2014 Conference on Visualization and Data Analysis, 2014. [JJDK14]

3.2.1 Preface

With the ongoing development and advances in technology, today we are able to track more movement data than ever. GPS receivers and other environmental sensors are shrinking in both size and weight every year and simultaneously get better in precision and battery life. Today, researchers are enabled to track smaller animals, such as crabs or hummingbirds, not possible previously. The trajectories recorded by such devices consist of a timestamp, a geospatial position, and optionally additional meta-information either environmental (e.g., precipitation or

[2] In this work, Katerina Vrotsou and Carlo Navarra invented and implemented the property-based simplification and enhancement. Katerina Vroutsou integrated my density-based simplification approach into a huger picture and was the main author of the publication. David Spretke implemented the Douglas-Peucker algorithm and the geospatial framework, in which I could integrate my density-based simplification technique. Georg Fuchs, Florian Mansmann, Natalia Andrienko, and Gennady Andrienko revised and co-authored the publication.

[3] The graph-based visualization and the temporal partitioning was implemented by Dominik Jäckle. I had the idea to perform clustering and use the resulting clusters in a graph-based visualization. The clustering and the heatmap representation of a cluster was implemented by myself. Daniel Keim and Oliver Deussen helped with fruitful discussions and advices

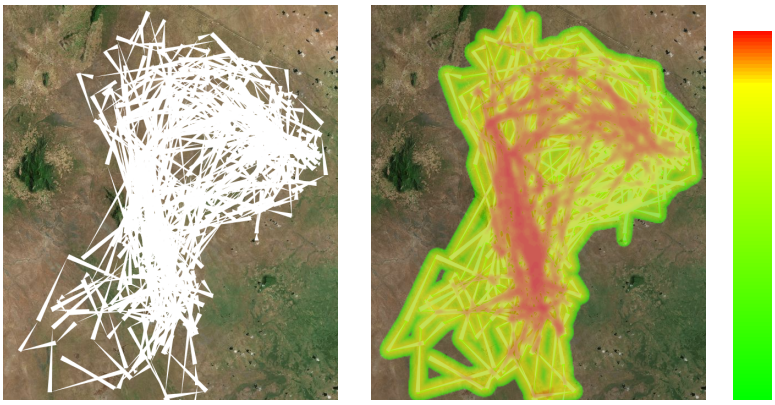
wind direction), medical (e.g., heart rate or body temperature), or movement related (e.g., posture or acceleration forces).

Consequently, the amount of data being recorded grows quickly in terms of temporal and spatial resolution and in terms of the number of moving objects being tracked. Furthermore, new additional dimensions as described above are measured and can reveal important insights. Unfortunately, there are severe challenges in analyzing and visualizing movement data due to the large amount of data with high resolution covering long periods of time. The three crucial points that have to be dealt with for an effective and successful analysis are:

- Tiny patterns or minor variations are hard to observe because of perceptional limitations.
- The cognitive load of the analyst should not exceed his capabilities – keeping track of all data is impossible and focusing on important patterns is crucial.
- Rendering performance issues should not detain the analyst from an interactive Visual Analytics process.

Although the recorded movement data is sampled and could be seen as point-based data, we usually interpret such data as trajectories and interconnect the measurements with lines. Such line-based representations of movement suffer even more from overplotting than point-based visual representations. When we investigate real-world movement of animals, we can observe different kinds of movements. There are territorial animals such as wolves or lions. Territorial animals stay in their territory and visit other places only in exceptional cases. In the opposite, migrating animals have areas where they stay for foraging or breeding but also cover long distance travels. Analyzing movement on very different scales is quite challenging and visualizing such data without distortion retaining both local details and the global context is difficult.

Movement data is not only challenging because of the large amount of data available but also because of the dynamic nature of movement patterns. Even the most regular movement patterns vary over time influenced by external factors as for instance weather or traffic jams. We exemplify a temporal shift of movement patterns with the help of a real-world data set in Figure 3.2.1. We enhance the visibility of the temporal movement shift by visualizing the density distribution (b) instead of visualizing the raw movement data (a). Low density areas are encoded by green, whereas medium and high densities are represented by yellow and red colors.



(a) complete trajectory **(b)** density visualization

(d) first third **(e)** second third **(f)** last third

Figure 3.2.1: Movement data of a lion over a period of two years. We partitioned the data set into three thirds visualized in Figures (c) to (e) showing a temporal shift. The temporal shift is not seen in the complete trajectory depicted in (a) and (b). Low density areas are encoded by green, medium density by yellow, and high density by red colors. Reprinted from [JJDK14], © 2014 SPIE.

The temporal movement shift is very salient, as we partition the recorded trajectory into three parts. The first third shown in (c) shows some kind of circular movement pattern in the northern region. The north-south transition pattern becomes obvious in the last third depicted in (e).

We propose in this section two techniques dealing with overplotting and reducing the number of details presented to the analyst. First, we introduce a novel line simplification algorithm being efficiently adjustable to the current zoom level. We compare our line simplification approach to two related techniques and discuss the merits and drawbacks. The second technique

described in this section consists of a temporal and geospatial abstraction, in order to visualize movement patterns in a visually less overwhelming way. We combine the abstracted graph-based visualization with land-use information incorporating some context information to further support the analyst.

This section is based on two complementing approaches and is structured as follows. We will first discuss methods simplifying movement data represented by linear segments. Instead of applying pure geometric-based simplification approaches, we derive descriptive features of the input and use this information to adapt the simplification to the data set. Later on, we will use the density of filtered movement records to show movement patterns from an abstract node-link-diagram view. All the techniques presented here will be applied to real-world data sets and the results will be discussed.

3.2.2 Related Work

Movement data has been in research focus for a long time and many analysis and visualization techniques have been proposed. Summaries of previous works in the geospatial domain can be found for example in [AAB+13a, GP08, GS05]. The goal for movement data is usually to enable analysts detecting and understanding movement patterns. As stated above, visualizing large and dense data sets is challenging and often standard techniques are not feasible at all. Sampling is a statistical approach often applied to reduce the number of data points while preserving statistical features of the data. In our example, presented in Figure 3.2.1, sampling would not be sufficient as the density is by far too high and overplotting issues too severe. We will present in the following section three different methodological approaches dealing with the analysis and visualization of movement data. At first, we will introduce simplification-based approaches followed by aggregation-based methods. We will furthermore discuss segmentation-based techniques partitioning the trajectories into meaningful units.

Simplification-based Approaches

Line reduction and simplification is one possibility to enhance the readability of geospatial movement visualizations. Several approaches in the domain of geometric line simplification have been proposed [DP73, VW93, LIS12], with the Douglas-Peucker algorithm being one of the most prominent ones. The recursive Douglas-Peucker algorithm determines successively significant points of the line and removes all other points. Smoothing techniques with focus

on maps are proposed by Burghardt et al. in [Buro5]. Smoothing is applied in order to reduce the visual clutter of angular lines (e.g., for railroads or buildings), sinuous lines (e.g., for rivers or coastal lines), and contour lines (e.g., isolines in maps). Curve simplification with focus on decreasing the runtime complexity is described by Agarwal et al. in [AHPMWo5]. Besides to computational geometry techniques simplifying lines, there exists a number of techniques reducing the temporal resolution of a trajectory. Laube and Purves discuss in [LP10] the influence of different temporal sampling on derived attributes as speed or sinuosity. The technical aspects of random sampling are discussed and described by Ellis and Dix in [EDo2].

The simplification methods described above are only applicable to a certain extent to movement and trajectory data. For example, coastal or contour lines do not contain self-crossings and consequently have less overplotting issues than geospatial movement. The most prominent simplification technique is probably the Douglas-Peucker algorithm and we chose it as the representative for simplification-based approaches despite its limits. Besides other drawbacks [VW90], the simplification result of Douglas-Peucker is highly sensitive to outliers. We propose in this section two additional simplification methods enhancing the pure geometrical simplification approaches. The proposed simplification methods take data features into account and are either density-based or property-based.

Segmentation of Trajectories

Partitioning complex movement data into coherent parts enhances the readability and understandability. Determining coherent segments allows visualizing trajectories in a more expressive way. Segmentation of trajectories into *episodes* is an ongoing research topic. Movement episodes are defined as parts of a trajectory having relatively coherent properties (e.g., speed, heading or sinuosity) and being cut by sudden changes in these properties [DMo3]. Usually, a new episode ranges from one cut to the next cut position and consequently all movement points belong exactly to one episode. A number of spatio-temporal cut criteria are discussed by Buchin et al. [BDvKS10]. Concerning performance, a variety of methods implementing cut criteria have been proposed [AVH+o6, GGM10, HBK+o7, PPK+11]. We use a slightly modified approach for segmentation as we are interested in regions where the movement of animals fulfills specific properties that label them as sleeping places or foraging areas.

The results of the segmentation process can be used for further analysis and visual presentation. Similar episodes of moving objects are detected and grouped by Laube et al. [LIWo5].

The authors look for specific motion patterns of several moving objects. Motion modes are automatically classified by Dodge et al. [DWF09] by performing an analysis of motion characteristics within trajectory episodes.

The episodes of a trajectory can be clustered in order to reveal further movement patters characterized by similarity in multiple attributes. Lee et al. [LHW07] describe in their work a framework based on partition-and-group clustering. They enable analysts to detect common sub-trajectories in multiple trajectories. Applying incremental clustering to detect common sub-trajectories is discussed and proposed by Li et al. in [LLLH10]. Lee et al. [LHL08] present classification techniques in order to classify movement according to properties with the help of generating a hierarchy of discriminative features.

The presented methods segment trajectories into coherent episodes with respect to given properties. Our approach is inspired by these techniques as we also partition trajectories according to user-chosen data properties and differentiate between interesting and not interesting movement episodes. Our clustering differs from these methods in that it is applied to arbitrary attributes chosen according to the analysis goals and is used for supporting visual exploration of movement patterns. In addition to the presented techniques, we use data statistics, namely density information, and integrate this information into the segmentation process.

Visual Data Representation and Exploration

Visualizing geospatial data can be naturally achieved by using map-based representations. Movement can be represented on both static and animated maps as discussed by Vasiliev in [Vas97]. Visualizing the temporal dimension simultaneously with the movement patterns is possible with a technique called space-time cube [Kra03, Kwa00]. Space-time cubes use a third visualization dimension to encode the temporal information of two-dimensional movement data. Multiple trajectories of aircrafts are visualized by Hurter et al. [HTC09] in a way allowing iterative queries by animated transitions between different projections of the data.

Aggregation-based methods avoid over-plotting at all and furthermore allow also a large amount of complex trajectories. There are several techniques applying aggregation to movement data. Density visualizations using surfaces represent often visited regions by heatmaps [DM03, FH00, Mou05, WVDWVW09a, SWvdWvW11, SWvdW+11]. Similar movements are grouped together in Flow Maps by partitioning the spatio-temporal space and visualize movements as a directed graph [Guo09, PXY+05, Tob87, BBBL11]. Spatially ordered treemaps introduced by

Wood et al. [WD08] can be used to display trajectories by aggregation. Methods for spatio-temporal aggregation are proposed by Andrienkos [AA08], an overview of existing approaches can be found in the framework by Andrienko et al. [AA10]. Another very related application area is the visual analysis of eye tracking data. Andrienko et al. [AABW12] and Li et al. [LÇK10] present different approaches based on clustering and the space-time cube. More related to the area of animal movement data also using aggregation-based techniques is the work of Grundy et al. [GJL+09].

In many application domains text data arises and such textual information needs to be displayed. Word Clouds [VWF09] are a prominent way to visualize the importance of textual fragments made popular by `wordle.net`. Word clouds have been applied in very related approaches. Nguyen et al. [NS10] display textual data on top of maps in a Word Cloud fashion. Important terms are displayed in larger font size than less important ones. The tags do not have their very own geospatial location but share the same location. It is therefore not possible to assign each tag their desired position. In [KKEE11], Kim et al. use Word Clouds to enhance node-link diagrams by visualizing Word Clouds instead or on top of nodes and edges. This approach is very related to our work besides the node-link diagrams have no spatial reference but result from document analyses. Another work using Word Clouds for analyzing land-use data is presented by Ferreira et al. in [FLF+11]. Word Clouds are here used as smart lenses displaying the land-use categories covered by the smart lens.

Visual data representation is essential for our methods presented in this section. In comparison to the techniques discussed above, we use density information of movement data for our simplification and visualization methods. We do not take three-dimensional visualization techniques as space-time cubes into account as overplotting is too severe for large movement data. Aggregation-based methods have to be tailored to the analyst's needs with focus on the respective task. In this section, we combine the partitioning of trajectories with density visualizations and furthermore provide a flow-based graph with further context information all based on the density distribution of the analyzed movement data.

3.2.3 Density-Based Line Simplification

The purpose of density-based simplification is to remove details from trajectory portions that exceed screen resolution and/or perceptual limits: dense clusters of data points can, in principle, be replaced by a single cluster representative. This kind of simplification is, hence, an

inherently viewpoint-dependent operation based on screen resolution and the current zoom level (in 2D) or virtual camera position (in 3D). Smooth interaction with the representation requires its application at interactive frame rates. The naïve approach of density-based clustering directly in screen space is thus hardly feasible as it implies re-calculating clusters for all visible trajectories based on changed node densities for every update of the viewpoint. We therefore propose a more efficient technique using real world/object coordinate-based densities allowing interactive simplification even of large trajectories.

The key idea here is to capture the relative densities of trajectory points in object space only once in a clustering preprocessing step. As the viewpoint changes these cluster results are transformed into screen space to obtain absolute (pixel-based) densities, which is computationally far less demanding and can be achieved in linear time. Figure 3.2.2 shows a schematic overview of this approach. Its individual steps are detailed in the following subsections.

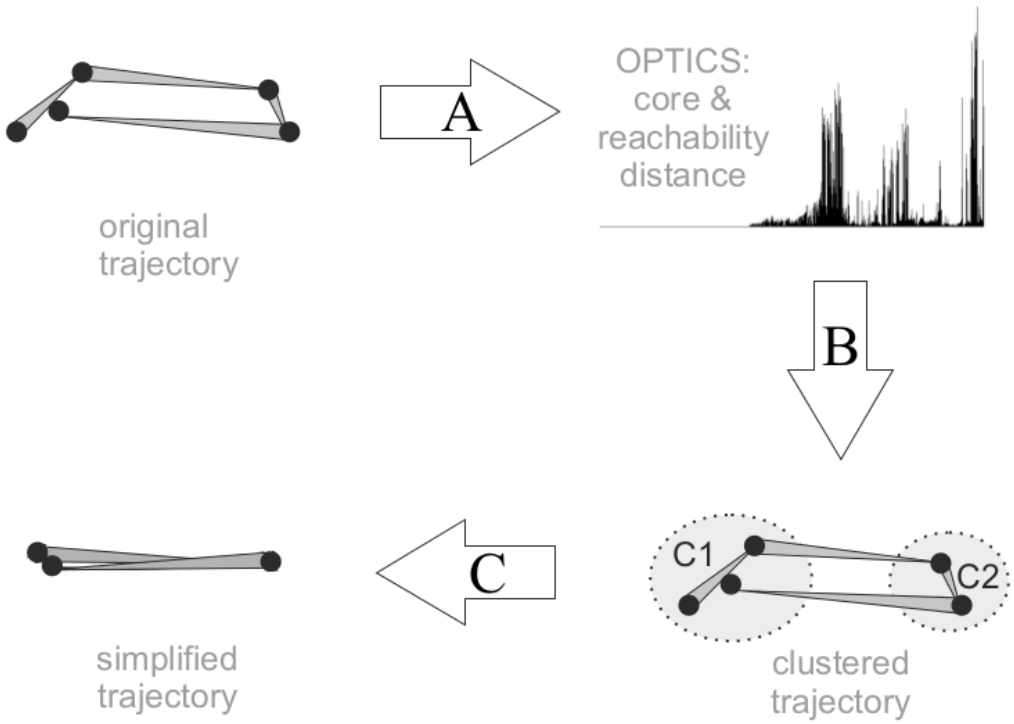


Figure 3.2.2: Schematic depiction of the density-based simplification approach. The input trajectory points are ordered into a core-reachability distance plot using OPTICS (A). From this plot a cluster partition is extracted depending on current on-screen size (B). Finally the clustered data points are processed sequentially to generate a simplified trajectory from cluster representatives (C). Reprinted from [VJN+15], © 2015 IEEE.

Clustering in data space

The primary notion of density-based clustering is that of so-called core objects that have at least *MinPoints* neighboring points within a maximum neighborhood distance threshold ε according to a defined distance metric. Points that are in the neighborhood of at least one core point are called density-reachable. Core points and density-reachable points constitute dense regions or clusters, whereas other points are considered noise not belonging to any cluster. Both *MinPoints* and ε are user-selected input parameters to the clustering process [HK06].

OPTICS [ABKS99] is an extension to this general density-based approach applying a specific sorting method to the input points prior to clustering. Beginning with an arbitrary core point, it first builds a core-reachability distance plot for a given value of *MinPoints* and a maximum distance value, d_{max}. From this plot cluster partitions of the point set can be extracted for different neighborhood distance thresholds, $\varepsilon \leq d_{max}$.

We apply OPTICS, as a one-time preprocessing step, to generate a core-reachability distance plot for a trajectory's data points (Figure 3.2.2A). The plot is built for a neighborhood size $MinPoints = 2$. This allows extraction of clusters of only two trajectory points as the smallest possible simplification step. d_{max} is chosen as the length of the trajectory's bounding box diagonal. Density clustering the set of trajectory data points with this neighborhood distance threshold guarantees that for any point all other points are within its neighborhood and thus, all belong to a single cluster. Therefore, the highest degree of simplification reached by our density-based line simplification approach is to collapse the entire trajectory into a single point.

Clustering result transformation

To arrive at an actual, screen resolution-dependent clustering of a trajectory's data points, the core-reachability distance plot of that trajectory is evaluated for a specific distance value, ε, according to the OPTICS algorithm [ABKS99]. This allows selection of an overall simplification level for the trajectory: higher values of ε result in larger and fewer clusters being extracted from the plot (and thus, a more simplified trajectory representation), whereas smaller values of ε generate smaller and more clusters retaining more of the original trajectory's details. Note that the evaluation of the core-reachability distance plot has linear complexity with respect to the number of trajectory points, making simplification level selection possible at interactive rates even for large trajectories.

In our approach, ε is determined for the current display size of the trajectory based on the

width in pixels of the primitive used to represent trajectory segments (e.g., simple lines or triangles, cf. top-left Figure 3.2.2). For this, an inverse projection of the pixel diagonal length at the current zoom level into data coordinate space is performed. For 2D maps ε is determined by finding the geographic distance covered by the diagonal of a pixel in the map representation multiplied by the primitive width in pixels, thus yielding ε as the primitives' width in data coordinate space. This results in segments of the simplified trajectory that are at least as long as they are wide in screen space. Shorter segments would only add variations in the trajectory's path which are hard or even impossible to perceive, since the corresponding bends between segments would be masked by the resulting overplotting.

Simplification

The final step is to derive a simplified trajectory representation from the view- and resolution-dependent data point clusters by replacing each cluster by its representative point, thus reducing the number of trajectory line segments (Figure 3.2.2C). In our approach, we use the arithmetic mean point of a cluster.

Algorithm 3.2.1: Building representative points from a clustered trajectory

Data: List of geographic data points of one trajectory (ordered by time) associated with a cluster ID (noise has a negative cluster ID)
Result: Simplified trajectory following a semantic zoom approach

```
Trajectory result = new Trajectory( );
int lastClusterID = -1;
DataPoint curRepPoint = new DataPoint( );
foreach DataPoint p in listOfClusteredObjects do
    if lastClusterID < 0 then
        result.insertPoint( p );
    else if lastClusterID == p.getClusterID( ) then
        curRepPoint.addPoint( p );
    else
        result.insertPoint( curRepPoint );
        curRepPoint = new DataPoint( );
        curRepPoint.addPoint( p );
    lastClusterID = p.getClusterID( );
return result;
```

However, a trajectory is a temporally ordered sequence of points, whereas data points are clustered only with respect to their spatial positions. This creates the risk of the temporal aspect being lost as points from temporally disjoint trajectory segments may be assigned to the same cluster (cf. cluster C_1 in the lower-right of Figure 3.2.2). In order to avoid losing the temporality of the trajectories, we address this risk explicitly by sequentially processing the trajectory data points from beginning to end, as depicted in Algorithm 3.2.1. Consecutive points assigned the same cluster ID are aggregated to obtain their mean representative, and as the next ID in sequence is encountered, the current simplified trajectory segment is finalized. Thus, if a cluster ID is re-encountered later in a trajectory the corresponding points from the cluster are associated with a different segment. Figure 3.2.2C shows an example for this: points from cluster C_1 are aggregated into two distinct vertices for the simplified trajectory.

Also note that singleton data points, or 'noise' in the selected cluster partition, are never merged with any other trajectory points, because they are sufficiently far from any other location at the current simplification level. On the one hand, there is no need in merging since these points do not contribute to clutter and overplotting; on the other hand, these point may convey important trajectory shape and object location information that should be preserved.

(a) 614 segments **(b)** 322 segments **(c)** 9 segments

Figure 3.2.3: Paragliding in Lake Chelan. A glider's trajectory simplified using density-based simplification at various levels of detail resulting in: (a) 614, (b) 322, and (c) 9 segments. Reprinted from [VJN+15], © 2015 IEEE.

We show in Figure 3.2.3 the simplification results of our technique introduced above applied to a paragliding trajectory. Depending on the screen resolution different aggregation levels are chosen. Starting from Figure 3.2.3 (a), with decreasing resolution our simplification algorithm

would aggregate more and more ending at the result shown in (c). The global movement pattern is visible in all of the three different simplification settings and even representing the trajectory by nine segments seem to be sufficient for a coarse but expressive shape.

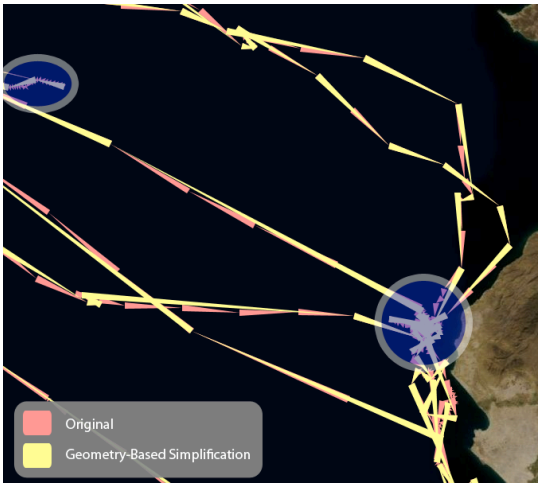
Comparison and Discussion

We will compare our simplification technique introduced above with two other techniques suitable for line simplification. We chose the prominent Douglas-Peucker algorithm as one state-of-the-art simplification algorithm to compare our technique with. The second technique discussed here represents another approach for line simplification applying property-based clustering. In the property-based approach, all data points are clustered according to the similarity of trajectory properties, like sinuosity, speed, or additional meta-data as temperature, and not with regard to geospatial density. Consequently, data points having exceptional property values will be shown to the user while others with frequent values may be replaced by a cluster representative according to the current zoom level. The efficient zoom level dependent simplification is achieved by hierarchical clustering and a quality-aware aggregation level selection. The aggregation level is chosen by analyzing the variance in the properties used for clustering.

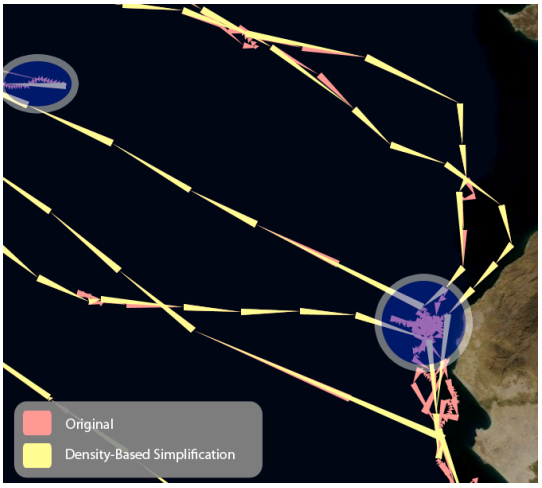
In Figure 3.2.4, we compare the three different simplification algorithms. They all preserve different aspects of the underlying movement patterns exemplified by a Galapagos albatross flight data set.

Geometry-based simplification algorithms such as Douglas-Peucker use only topological and structural properties of movement and do not take further attributes into account. Geometry-based techniques are typically applied for simple lines in maps and are not perfect for more complex lines as trajectories for example [VW90]. In our example in Figure 3.2.4 (a), we applied Douglas-Peucker as a representative for geometry-based line simplification algorithms. Douglas-Peucker achieves only very limited simplification and focuses mostly on long straight movement parts. In particular, small movements in dense regions are not simplified at all and self-intersections and overplotting are severe problems. Since dense regions are quite common in trajectory data, we need techniques dealing with them properly.

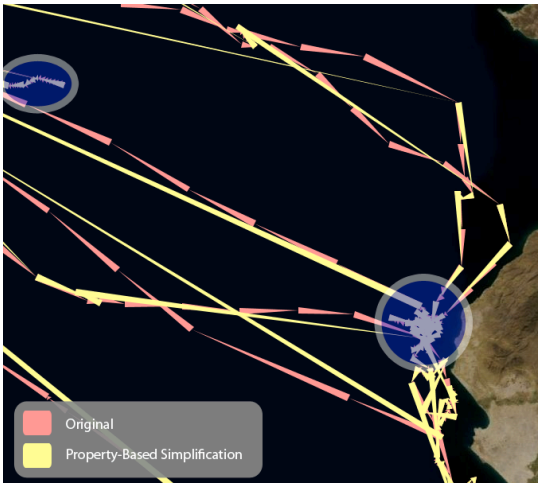
We applied our density-based simplification method to the very same albatross data set in Figure 3.2.4 (b). Density-based methods will simplify dense regions according to resolution and zoom level. By simplifying dense regions, these techniques will focus on where a moving object has been and avoid visually cluttered dense regions. Consequently, screen and perceptual



(a) Geometry-Based Simplification



(b) Density-Based Simplification



(c) Property-Based Simplification

Figure 3.2.4: Galapagos albatross trajectory simplified by three simplification approaches and represented using tapered segments [HvW09]. (a) Geometry-Based Simplification (using the Douglas-Peucker algorithm) achieves a good overall fit to the original trajectory. (b) Density-Based Simplification precisely models the long migration segments and simplifies dense regions so that the path of the albatross can be nicely followed after entering the marked dense region on the right. (c) Property-Based Simplification merges segments having similar attribute properties (e.g., heading and sinuosity) and preserves segments with attribute changes. Reprinted from [VJN+15], © 2015 IEEE.

limitations in terms of how much information is distinguishable are taken into account and overplotting will be reduced.

The property-based simplification reveals how a moving object behaves with respect to potentially both geospatial topology and properties depending on the analysts choice. In Figure 3.2.4 (c), the albatross data set is simplified by the property-based simplification. The approach reduces the cognitive load by showing only important changes of properties. Perceptual limitations are tackled but not completely solved by aggregating segments with minor variations in the selected properties. Screen and resolution limitations are per se not covered but the approach could to some extent be enhanced with respect to these limitations.

Comparing Figures 3.2.4 (a) and (b) in detail, we can spot two major differences resulting from the different underlying approaches. The first difference is related to straight movement patterns consisting of several segments being simplified to one single line by the property-based simplification (c). The property-based simplification, however, does not take heading and sinuosity into account and straight movement patterns will be only simplified if the resolution is not high enough. The second more severe difference corresponds to simplification of line segments in dense regions being highlighted by circles in Figure 3.2.4. The density-based simplification method (b) is the only technique focusing on the density distribution and especially tries to reduce overplotting in dense regions. Consequently, the overall movement in the encircled regions is better visible in the density-based simplification.

Although it seems that the density-based simplification outperforms the property-based approach, both techniques have their merits depending on the analysis tasks. We want to emphasize and illustrate the merits of the property-based simplification in another comparison. In Figure 3.2.5, we investigate the simplification of a 3D flight trajectory. The density-base approach replaces spatially close points by a representative point, whereas the property-based simplification method simplifies only trajectory segments if their properties are quite diverse. This behavior can be seen in Figure 3.2.5 highlighted in yellow. The left highlight shows an example where a single segment of the property-based simplified red trajectory corresponds to several of the density-based simplification. The right highlight gives an example for a single segment of the density-based simplified blue trajectory corresponding to several of the property-based simplification. These differences are also highlighted in Figure 3.2.4.

Both techniques have their merits and drawbacks and the decision which to apply is highly analysis task dependent. With both techniques being designed in a way to support the respec-

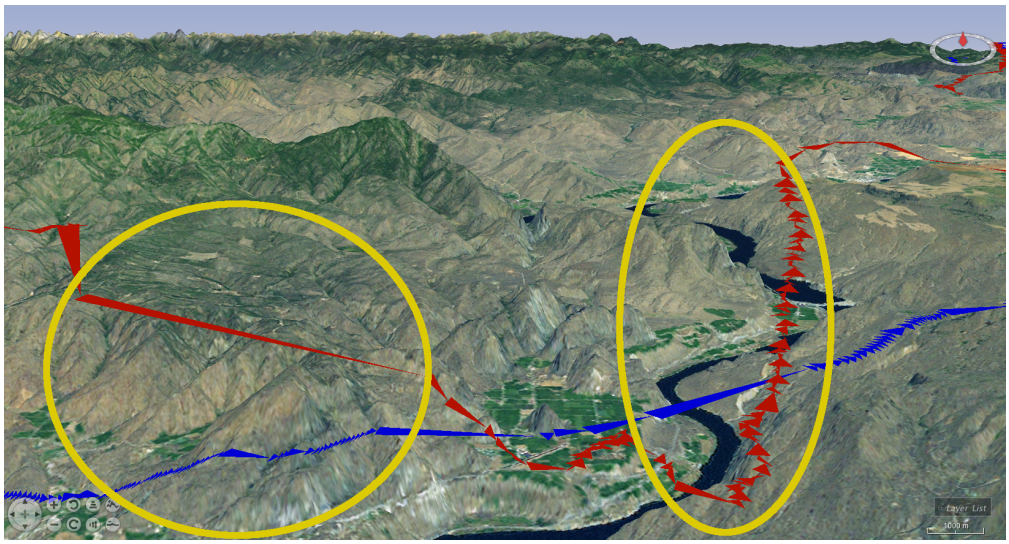


Figure 3.2.5: Results of the density-based (blue) and the property-based (red) simplification methods applied to the same trajectory and displayed in the same image using tapered segments. Differences between the methods are highlighted in yellow. Left: a single property-based simplified trajectory segment of constant descend corresponds to many density-based simplified segments of slightly varying course. Right: a single blue density-based simplified segment corresponds to many property-based simplified segments of an upwards spiral. Reprinted from [VJN+15], © 2015 IEEE.

tive analysis task by parametrization, the analyst is able to tailor the simplification to his needs. Nevertheless, the simplification process is always a trade off between the number of data points shown and the accuracy of the simplified trajectory. Although we lose movement details, we preserve the temporal aspect of the movement patterns as the representative segment covers the very same time span as the cluster of points being represented. The simplification will only change the temporal sampling rate of the underlying movement pattern.

3.2.4 Semantic Trajectory Abstraction

We describe in this section an approach for trajectory abstraction with focus on both the temporal and geospatial aspects of movement. As motivated by our example in Figure 3.2.1 movement patterns may vary over time. Consequently, we first aim to detect movement patterns and afterwards show their development over time. In detail, we automatically detect dense regions and use them for visual abstraction of complex motion patterns. We partition the trajectory into regions with high density and transitions between them. Additionally we analyze and de-

tect temporal shifts in movement patterns and use them for what we call temporal abstraction. We furthermore analyze the underlying land-use distribution and display the land-use by word clouds.

In this section, we will first introduce our approach for the visual abstraction with focus on the geospatial dimension. The temporal dimension is tackled in the second part of this chapter and deals with the automatic detection of coherent time windows.

Geospatial Abstraction

The basic idea of our technique is to visually abstract the trajectory when certain properties are fulfilled (e.g., low speed or bad weather conditions) and to show only simplified transitions. More specifically, we reduce the amount of over-plotting by a process of filtering, clustering, and finally visual abstraction. The overall abstraction process is depicted in the schematic Figure 3.2.6. In the filtering step (I), the analyst can at first interactively select points of the trajectory with specific attribute properties. We therefore provide filter functionality for attributes like speed, heading, or duration and also for additional attributes like weather information as precipitation.

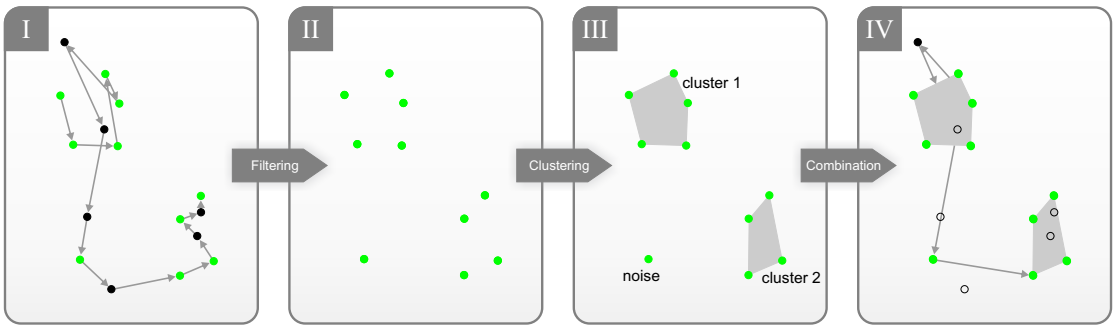
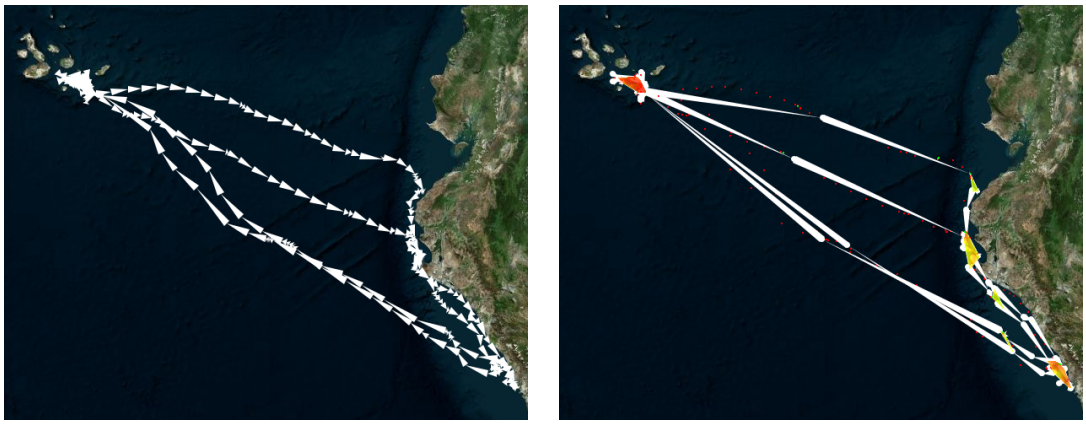


Figure 3.2.6: Process pipeline of the visual abstraction beginning with the raw data in (I) applying filtering (II), clustering (III), and finally combining and visualizing the results in (IV). Only the green points fulfill the user selected filter criteria. Reprinted from [JJDK14], © 2014 SPIE.

The user interface for filtering is capable of a concatenation of several filters tailored to the analyst's needs. For a more effective filtering we support the analyst by providing a histogram of the attribute's value distribution. All data points passing the filters, shown in Figure 3.2.6 (II) are then clustered by DBSCAN [EKSX96], a density based clustering technique, providing the

result in Figure 3.2.6 (III). The user can control the clustering granularity as he can influence the *epsilon* and *MinPoints* parameter interactively. For each cluster we then compute the convex hull and use it as a visual abstraction for this part of the trajectory. In addition to drawing the convex hull polygon, we fill each convex hull with the density distribution visualized by a heatmap. The colormap depicted in Figure 3.2.1 encoding the density distribution goes from green (low density) over yellow (mid density) to red (high density).



(a) original trajectory **(b)** geospatial abstraction

Figure 3.2.7: Our geospatial abstraction method is applied to an albatross trajectory depicted on the left hand side. The final result of the proposed method is shown on the right side. We used for filtering a low speed in order to find resting and foraging places. Reprinted from [JJDK14], © 2014 SPIE.

The last step is to visualize the trajectory on a geographic map, while using the previously computed convex hulls and using simplified trajectories in between. For simplification purposes we use only one intermediate point between the clusters. Simplifying the transition trajectories can be performed by applying the Douglas-Peucker simplification for example. We iterate over all points of the trajectory and look only for points being the transition from one convex hull to another. Note that we also handle cases where these two convex hulls are identical and still show the transition. These transitions are then visualized in a simplified version to only give a rough overview of the transition course, depicted in Figure 3.2.6 (IV). A final result of the transformation process can be seen in Figure 3.2.7, where we applied our method to a trajectory of an albatross.

We do also provide another visual representation of the processed trajectory. Basically, the

convex hulls and the transitions in between can be seen as a graph network. We therefore implemented a graph visualization of the trajectory with the convex hulls being the nodes and the transitions being the directed edges. The graph layout is implemented in a way that it reflects the geographic relations while providing a very high-level and abstract overview of the data. The graph-based and the map-based representation can be shown side by side supporting Brushing and Linking. The interaction between both techniques gives the analyst an overview graph, showing the major moving patterns, and enables further detailed analyses in the map-based visualization.

Temporal Abstraction

The result of the geospatial abstraction described in the previous section is used as an input to our temporal abstraction algorithm. We visualize the identified convex hulls and transitions as a graph network. The graph network is an abstract representation of the processed trajectory, but does per se not include additional information compared to the map visualization. Therefore, we add supplemental information to the graph network to enable further analysis steps. Each identified cluster as well as all transitions between these clusters contain several points holding spatio-temporal information. We are using both, the time and the spatial location of these points, in order to enhance the visual representation for deeper understanding of the movement data.

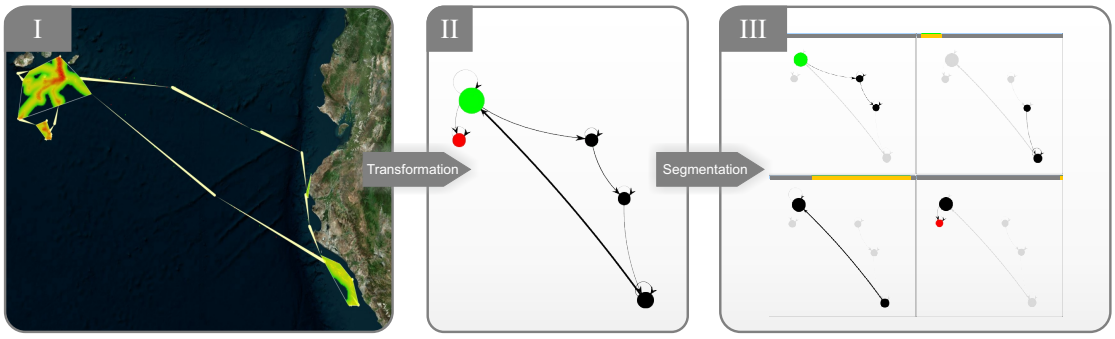


Figure 3.2.8: Exemplified abstraction of the map visualization (I) into a graph network (III). All clusters are represented as nodes and transitions as directed edges. The position of the nodes of the graph (II) corresponds to the relative geospatial positions of the clusters preserving a visual alignment of map and graph. The resulting graph is partitioned into several graphs in (III), each representing a certain timespan and a certain number of transitions. Reprinted from [JJDK14], © 2014 SPIE.

Figure 3.2.8 gives an overview of the overall transformation process. At first, we need to map the position of a cluster, visualized by the convex hull, to the node position in the graph network. We want to keep the displays as similar as possible and retain the relative positioning as best as possible. The convex hull consists of multiple points and we therefore calculate the centroid of the convex hull defining the position of the node. This step visually aligns the representation on the map with the abstract representation as graph. Once the position has been calculated, several attributes can be used to affect the node size: The amount of points enclosed by the convex hull can give a global overview in which cluster most of the time was spent. But the size can also be determined by mapping the size of the enclosed area to the node size. Another possibility is to map the time span a cluster was visited to the node size. The application of these different parameter mappings depends on the analysis.

The transition from one cluster to another is visualized by a directed edge. In order to reduce visual clutter, multiple transitions between the same two clusters are aggregated and handled as one. If a moving object, for example, often leaves the convex hull and returns immediately, we represent this as one single reflexive directed edge. Here also the possibility is given to use different attributes to affect the stroke width. The user has two possibilities: Either using the amount of intermediate points or the amount of transition revisits.

Crucial for the temporal abstraction is handling the temporal dimension in a meaningful and expressive way. Our goal is to retrieve further information like motion patterns and therefore a segmentation of the motions according to different criteria is needed. One possibility is, to divide the motion data into several equal-time intervals and to visualize the visited nodes and edges. In this case, at least two severe problems occur: First, selecting a fixed time span may lead to patterns not being visible as they might cross the border between the time spans. Second, by drawing the affected nodes and edges only the context of the motion pattern is missing. It may be impossible to put the affected nodes and edges into context with respect to the entire graph network. The fewer nodes a motion pattern covers and the bigger the entire graph is, it is more and more difficult to visually map the nodes. Hence, we propose our technique combining the overview and detail methodology [PCS95] with a semi-automatic algorithm for defining the time intervals dynamically. Furthermore, we integrate the Small Multiples technique [Tuf90] to visualize all time intervals side-by-side. Small Multiples are the combination of several snapshots of the same visualization in a dashboard-like representation; in our case, every snapshot represents a different time window. Furthermore, we enable the analyst to influence the time segmentation process according to his needs.

As stated above, the geospatial context of the motion pattern may not be retrievable when visualized alone. As a context visualization we draw the entire, fully connected graph, but gray out all nodes and edges not being contained in the calculated time span. In addition, the very first node of the trajectory is drawn green and the very last node is shown in red.

As already mentioned, the time intervals are calculated automatically and do not cover equal long time spans. Algorithm 3.2.2 shows the computation process in more detail. We try to avoid to partition the graph at points in time where the motion pattern would be interrupted, for example in the middle of an intermediate edge between two clusters, we propose the usage of a threshold. This threshold decides after how many hops the motion will be interrupted to create a new Small Multiple or rather to end the time interval and to start a new one. The user can change the threshold interactively and directly influence the partitioning process. Figure 3.2.8 (III) shows the result for the threshold zero. Every transition from one cluster to another is visualized in a separate Small Multiple.

Algorithm 3.2.2: Determining the Small Multiples inclusive the covered time span

```
multiples ⟵ ∅
multiple ⟵ createMultiple()
hops ⟵ 0
for iTraj ∈ intermediateTrajectories do
    if iTraj.fromCluster equals iTraj.toCluster then
        multiple.addIntermediateTraj(iTraj)
    else
        if hops < threshold then
            multiple.addIntermediateTraj(iTraj)
            hops ⟵ hops + 1
        else
            multiples ⟵ multiples ∪ {multiple}
            multiple ⟵ createMultiple()
            multiple.addIntermediateTraj(iTraj)
            hops ⟵ 1
        end
    end
end
if multiple is not empty then
    multiples ⟵ multiples ∪ {multiple}
end
```

The segmented graph, resulting from the user defined threshold, covers a certain time span. We determine the exact time span using the temporal information of the aggregated trajectory points. We consequently calculate the temporal distance between the first point leaving a cluster and the last point before entering the last cluster of the segmented graph (see Figure 3.2.9). The computed time span specifies the covered travel distance. A time bar at the top of each Small Multiple visualizes the covered time span. The orange bar specifies the covered time span in relation to the full span (dark gray). The transition between one Small Multiple to another results in a small gap between the covered time spans. This gap represents the time a mover remains in a cluster before it starts a new journey.

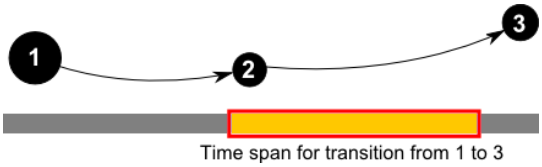


Figure 3.2.9: This figure shows the determination of the time span. The time span starts when leaving cluster 1 and ends when entering cluster 3. Notice, that the points contained by the clusters 1 and 3 do not contribute to the identified time span. Reprinted from [JJDK14], © 2014 SPIE.

Analysis and Visualization of Land-Use

The temporal abstraction shows per se only spatial and topological relations and omits some spatial context information being available in maps. From our discussions with subject matter experts we noticed that some additional context information in the temporal abstraction view would be beneficial. Consequently, we integrated land-use information into our prototype. For each cluster, we determine the most prominent land-use categories and their frequencies. This information is used to create a Word Cloud like visualization on top of the temporal abstraction view. We will describe further details about the land-use integration in the next paragraphs.

As input for our land-use determination we used the GLOBCOVER data set provided by the European Space Agency. The GLOBCOVER data set differentiates between 22 different land-use categories with a spatial resolution of 300×300 square meters. For each of the clusters, we determine the land-use categories for all the points belonging to the current cluster and store this information. Lastly, we compute the relative frequencies of land-use categories for each cluster.

The distribution of land-use categories is used to draw a word cloud of the most prominent categories. We implemented the Word Cloud algorithm described in [VWF09] following an Archimedes spiral in order to detect the next free position. Basically, we compute for each cluster a Word Cloud separately and add them to the temporal abstraction view. We assign the cluster's position to the corresponding Word Cloud. Furthermore, we use a global list of already occupied positions to avoid overplotting of land-use labels of different clusters. For better legibility of the combination of temporal abstraction and land-use Word Clouds, we color the outline of the black labels by white coloring and set the transparency of the Word Cloud layer to fifty percent.

3.2.5 Application

For all techniques it is challenging to be applicable for a wide variety of different motion patterns. There are on the one hand very condensed, territorial trajectories and on the other hand large-scale movements with highly varying velocities. We will show that our methods are applicable to every type of movement data. We will apply our techniques to different kinds of animal movement with varying density and velocity properties. More in detail, we will show and analyze movement data of albatrosses and lions with our methods. An albatross travels over long distances between regions with slow movement, whereas lions stays at a certain region and moves inside his territory. In both applications, the amount of points enclosed by the convex hulls is mapped to the corresponding nodes and the amount of intermediate points is mapped to the stroke width. The land-use categories are not varying at all for the first two application examples. The albatrosses have basically only measurements on water areas and the lions are tracked in savanna areas. We will lastly investigate the migration movements of white storks tracked on their journey from Germany to Southern Africa. The land-use categories differ strongly along their movement trajectory.

Motion analysis of albatrosses

The movement data of albatrosses is characterized by varying distances, densities, and velocities. To apply our proposed methods, we analyze an albatross whose movement is between the Galapagos islands and the coast of Ecuador. The motion data of the albatross consists of 1113 points. The covered time span lasts from May 31^{st}, 2008 until August 9^{th}, 2008. First of all, we apply a filter of the speed determining the computed clusters. Filtering the low speed reflects

resting and foraging places. Second, we adjust the parameters for the density based clustering algorithm (DBSCAN). We choose a high amount of minimum points and a low distance rate to determine the clusters. For the graph segmentation we apply a threshold of seven transitions. Figure 3.2.10 shows the resulting graphs.

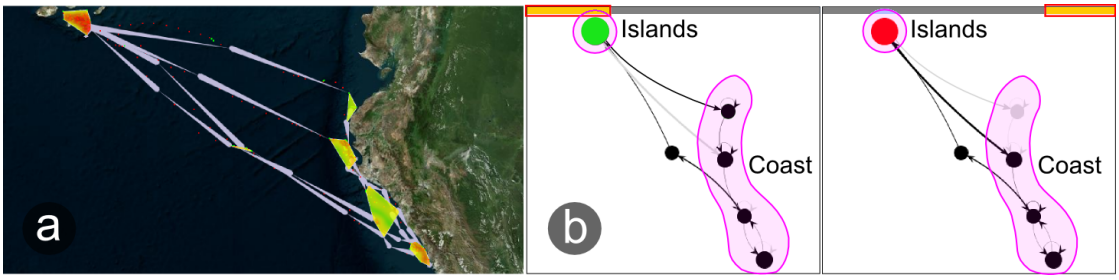


Figure 3.2.10: This figure shows the (a) geospatial as well as the (b) temporal abstraction of the movement data of an albatross. The (b) temporal abstraction shows two Small Multiples with the threshold of seven transitions in order to segment the graph in the biggest cluster (the Galapagos islands) and to unfold patterns. The orange, red framed time bar on top of each Small Multiple visualizes the covered time span. Reprinted from [JJDK14], © 2014 SPIE.

The threshold of seven transitions has been chosen in order to separate the Small Multiples at the biggest cluster node. By applying the temporal abstraction, two abnormalities are made visible. The first one is the stopover, represented by the cluster for the Galapagos islands. The chosen segmentation for that cluster unfolds a wide time gap between the two Small Multiples. This time gap is revealed by the time bar. The time bar for the Small Multiple on the left hand side visualizes the time span for the transition from the starting cluster to the cluster representing the Galapagos islands, which is also the largest (Figure 3.2.10 (b), the orange, red framed time span on the left). The time bar for the Small Multiple on the right hand side visualizes the time span from the largest cluster to the cluster where the movement ends (Figure 3.2.10 (b), the orange, red framed time span on the right). By comparing both time spans with each other, there is a wide span missing in between. This gap represents the time spent by the albatross on the Galapagos islands. From this inspection we can suspect the Galapagos islands to be his roost.

Another observation is related to the movement along the Ecuadorian coast. The motion patterns and clusters show, that the albatross remains in the bay areas and furthermore all clusters representing the bay areas all have reflexive transitions indicating the circling of the albatross.

For these areas it is likely that the albatross was hunting, indicated by the time spent and the circling behavior. After consulting ornithologists, they approved, that the biggest visible cluster – which represents the Galapagos islands – is the bird's roost and the remaining clusters along the Ecuadorian coast represent the bird's hunting grounds.

Motion analysis of lions

Compared to albatrosses, lions usually do not cover wide distances. Hence, the motion data is very dense being challenging for visual analyses. In this section we will compare the motion data of two different lions, living in the savanna. For the first one, Figure 3.2.11 shows (b) the geospatial as well as (c) the temporal abstraction of (a) the raw motion data. The motion data consists of 396 points and the covered time span lasts from April 28^{th}, 2002 until September 13^{th}, 2002.

We choose to filter the speed, so that only the motion with a low speed is being considered. Furthermore, to adapt the number of Small Multiples to the movement we choose a threshold of three transitions. Figure 3.2.11 (b) shows the geospatial abstraction and thus a very dense and large cluster. Via this visualization we can see where the lion probably remained most of the time, but we cannot clearly identify the movements in between the clusters; (c) gives us that information and also reveals a moving behavior of the lion: The purple highlighted node in (c) is drawn black in every Small Multiple. This means, the lion always returns to this place after visiting at most two other places (indicated by the threshold of three transitions). Moreover, the graph shows a second node with the same size of the purple marked node, which means, that in that area the lion did spend the same amount of time, indicated by the amount of spatial points included in that cluster. In the geospatial abstraction the corresponding cluster has not the same size, but contains a very dense movement.

Figure 3.2.12 shows the second lion and the corresponding movement analysis. This motion data covers the time span from April 28^{th}, 2002 to July 21^{st}, 2007. Furthermore, it contains 1465 recorded points and is very dense (see Figure 3.2.12 (a)). Therefore, we filter for a very high speed rate to refine the motion data in order to identify hunting scenarios and then apply the geospatial abstraction (b) . This visualization contains several small-sized clusters being barely visible due to the very small clusters. To visualize them in a more suitable way, we apply (c) the temporal abstraction with a threshold of two transitions.

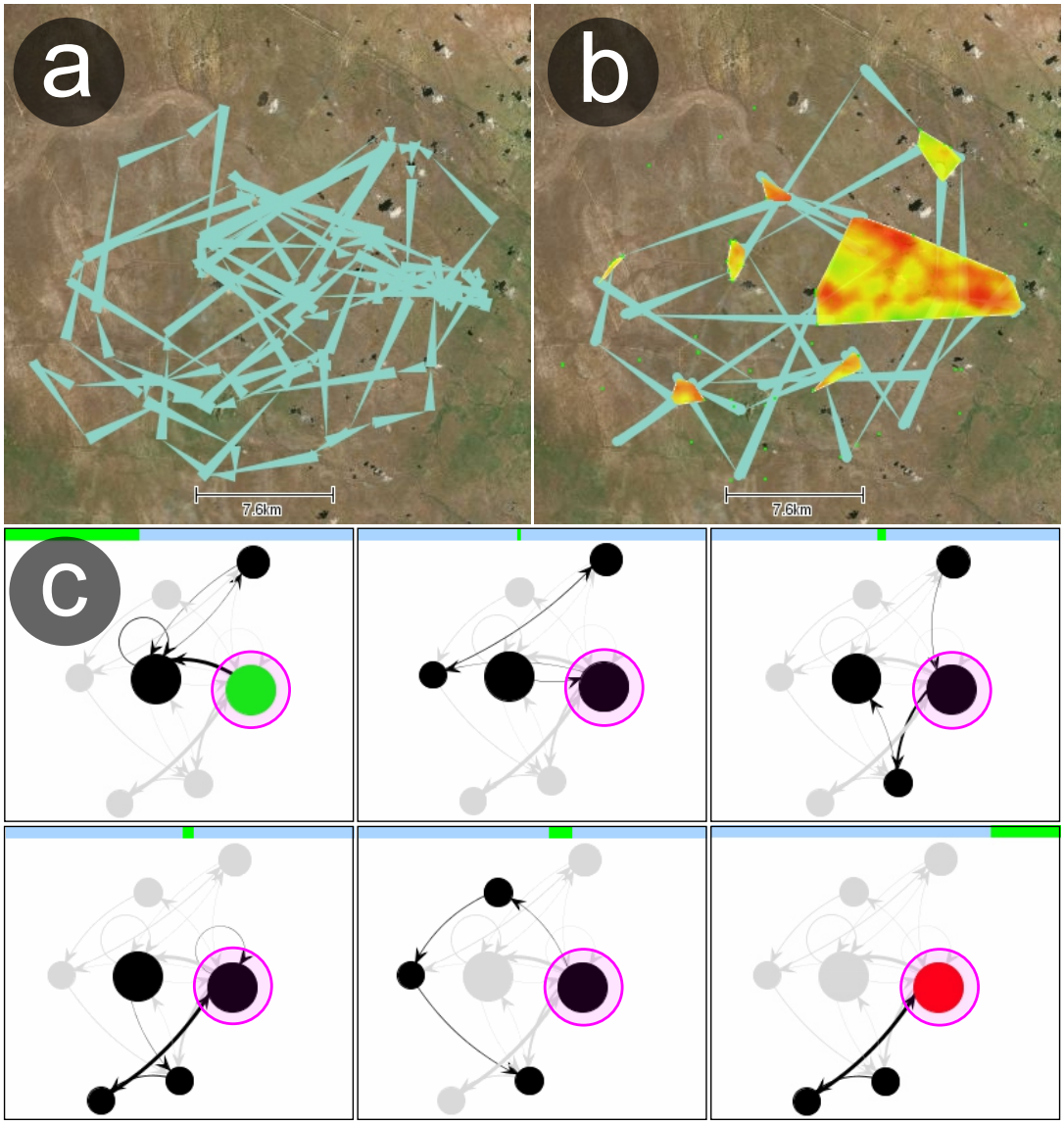


Figure 3.2.11: This figure shows the (a) raw movement data, (b) the geospatial, and (c) the temporal abstraction of a lion living in the savanna. Figure (c) reveals the pattern that the lion has one favorite spot. Reprinted from [JJDK14], © 2014 SPIE.

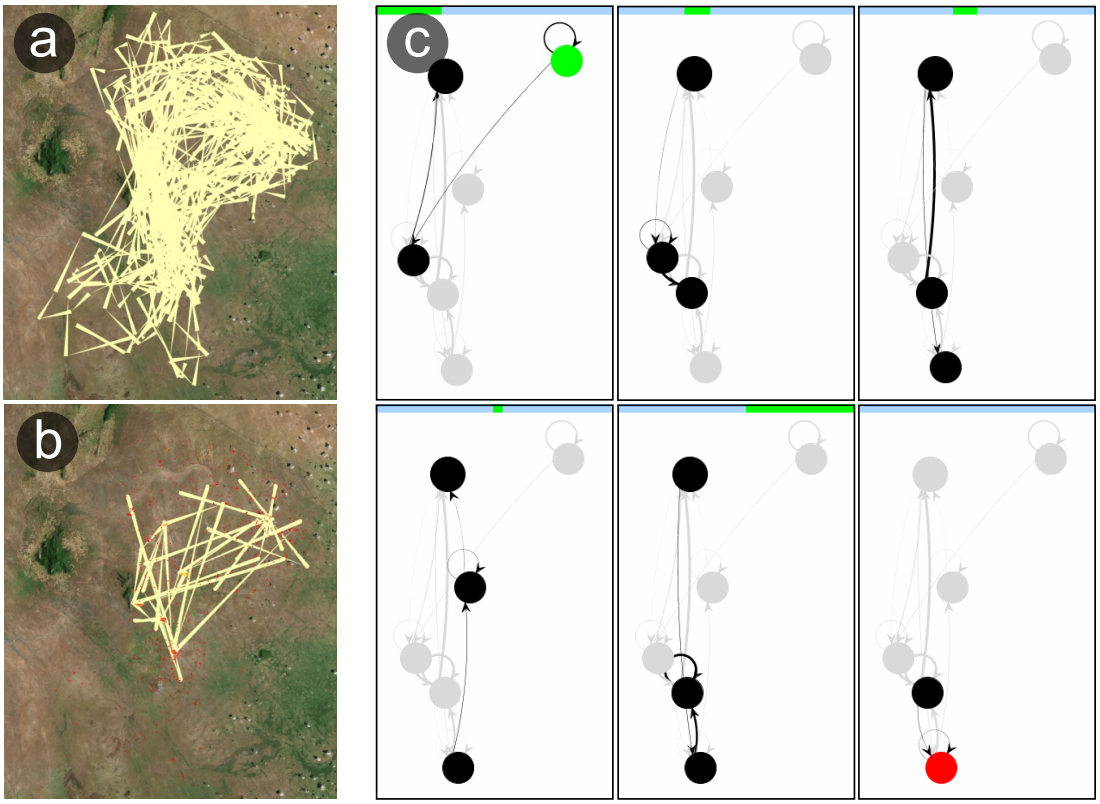


Figure 3.2.12: This figure shows the (a) raw movement data, (b) the geospatial, and (c) the temporal abstraction of the second lion living in the savanna as well. Figure (c) shows the abstraction of the hunting grounds. Reprinted from [JJDK14], © 2014 SPIE.

The resulting Small Multiples make these clusters visible to the user. As this motion data has been filtered for movements with a high speed rate, the clusters do stick out, because they highlight a high speed in a dense area. High speed of lions is typically related to hunting and the Small Multiples reflect the temporal sequence of hunting grounds.

Motion analysis of white storks

In this section, we investigate the migration pattern of a white stork. The data was collected between August 1998 and May 1999 and contains 887 points. White storks migrate every fall from Europe (in our case Germany) to Southern Africa and spend the winter there. In spring, white storks will travel all the way back to spend the summer in Europe. Consequently, the land-use categories will vary along the trajectory path.

For our studies, we were interested in the stop-over areas, where the storks rest and forage regaining their power. We filtered for data points with very slow speed and performed the above described clustering. The resulting stop-over areas can be seen in Figure 3.2.13. We partitioned the overall movement pattern into two sections, basically the way forth and back. Furthermore, we added the most prominent land-use categories as a semi-transparent layer on top of the temporal abstraction visualization.

Obviously, the storks travels through very sparse vegetation areas, being visible on the traditional map in Northern Africa. Nevertheless, all the stop-over points share the same kind of land-use categories, namely croplands, forests, and bare areas. According to ornithologists this is reasonable as the animals need a certain kind of environment to rest and forage. Our visualization of land-use categories added more context to the stop-over areas, which allows the analyst to get more insights to the bird's movements. It is important that these stop-over areas are protected in order to ensure storks being able to rest.

3.2.6 Expert feedback

Though, we showed the applicability of our proposed technique in the previous application section, it is very important that real users rate our approach effective and helpful. We therefore conducted an expert study with two biologists, both highly experienced in the domain of movement and trajectory analysis. They usually perform their data analyses in command-processing analysis environments, like the statistical toolkit R. Interactive parameter setting and immedi-

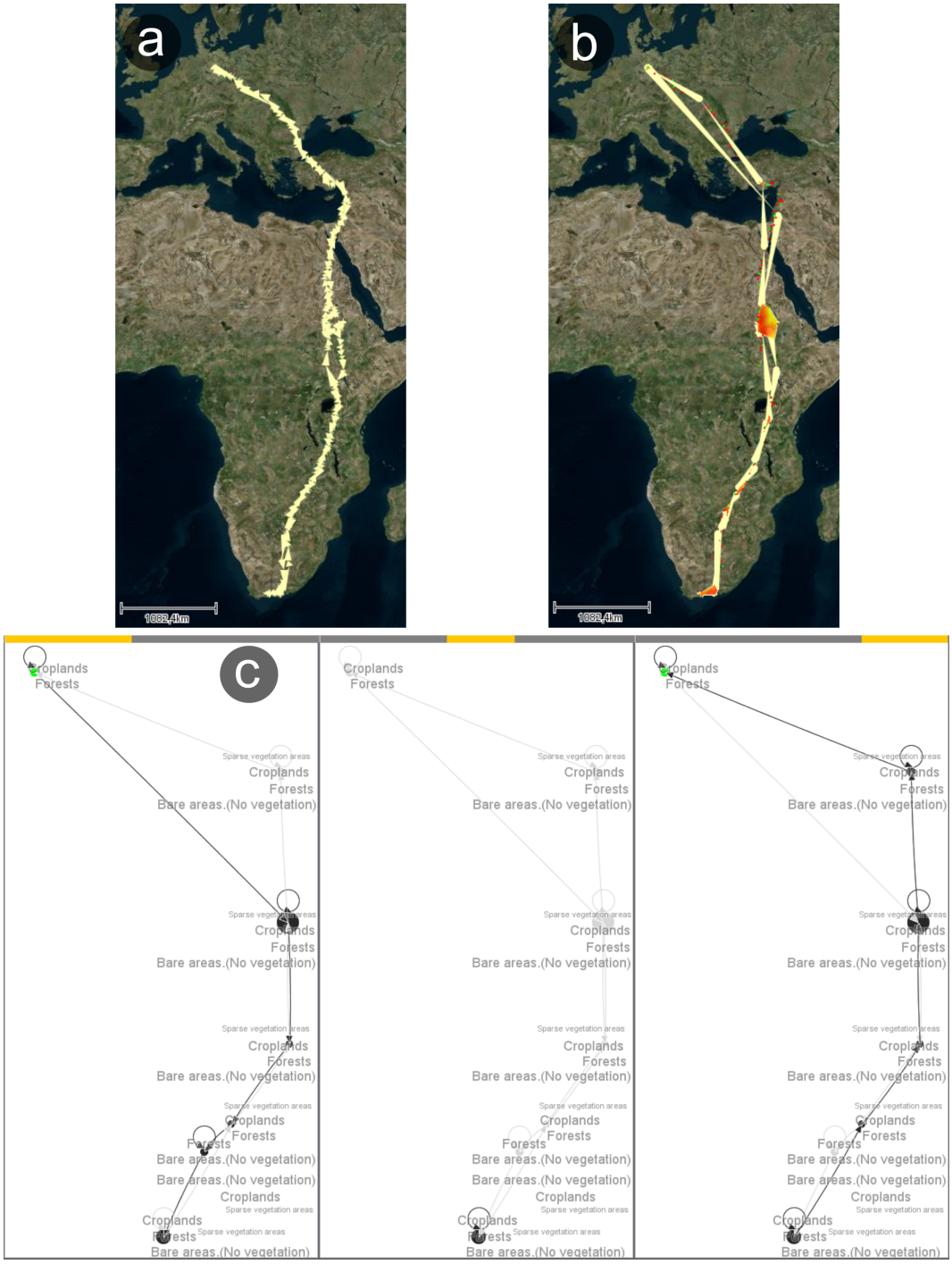


Figure 3.2.13: The analysis results for a white stork is depicted in this figure. (a) shows the original trajectory, while (b) presents the geospatial, and (c) the temporal abstraction of the white stork's movement. We added the contextual land-use information to the temporal abstraction in (c).

ate visual feedback of our system was therefore highly appreciated. We got very valuable feedback resulting in aspects for future work (see next section) and the following discussions of the merits and limitations of our technique. We first explained our approach to the biologists and afterwards let them interact with our system and look at the different results for the applications described above. We asked them to describe their typical way of analyzing data and furthermore to comment on our proposed technique.

The first feedback we received was related to the two ways of visualizing the same data in a map and in a graph. The biologists stated they would first look on the map for interesting clusters based on the surrounding geographic topology and afterwards investigate the corresponding node in the graph-based representation. They were quite interested in the previous visited node, which was easier to find in the abstract graph than in the map. Furthermore, the directly next visited cluster is also important to them as they want to understand the context of the observed cluster. The identical spatial layout of the graph corresponding to the topology of the map was considered very helpful. The biologists were quite enthusiastic about the fact that they do not have to deal with the high degree of over-plotting anymore. They usually visually inspect only small parts of trajectories due to high degrees of over-plotting or apply aggregation techniques and visualize kernel-density estimations.

Another very interesting result of the feedback session was the usage scenario the biologists described. As described above, our expert users had a high background in statistical analysis, which is quite common for biologists. They typically import their trajectories into R and perform analysis tasks like statistics or data mining. What they are really lacking is an interactive system enabling them to investigate the parameter space of the applied algorithms. As we showed them our system and let them interactively change the clustering parameters, the biologists were discussing the different clustering results intensively. One outcome was that they would like to have the resulting clusters and polygons in order to import it to R and perform further statistical analysis. Additionally, they would like to have the possibility to annotate the resulting clusters in order to materialize their findings and again export these annotations.

One more serious point arising along our study several times was the importance of the proper filter settings. Only if the analyst knows what he is looking for and is capable of specifying the properties accordingly the analysis process will result in semantically meaningful results. Without any knowledge of the mover's foraging behavior, for example, it will be very difficult to detect these motion patterns. One biologist discussed the difference between migration and foraging patterns for sea birds concerning speed and variation of headings. For migration pat-

terns the heading stays typically the same while traveling quite fast. In the case of foraging, speed is typically lower and variation in motion headings is higher compared to migration settings. In our case, it holds true that the experts know what they are looking for and are able to specify filter parameters accordingly. But there might be cases where this is not possible, e.g., looking for unknown behavior patterns like displaying of very rare and shy birds.

What the experts liked from a biologist's point of view is the combination of the home-range philosophy with occurrences of behavioral patterns. The home-range is defined in biology as the region, where the animal spends fifty percents of its time. In our case the home-range can be seen defined by density, because our trajectories are equally sampled. The reason why the biologists liked the combination of both approaches can be seen in one of the higher goals of the biologists. By understanding the resource requirements for distinct motion patterns they can save areas with corresponding resources as nature protection areas and consequently stop the extinction of endangered species. Combining the notion of density with motion patterns has the advantage that areas where the animal showed a specific behavior only once are considered as outliers and not shown in the abstract visualizations. The data basis for further investigations of interesting behaviors is therefore stronger and more convincing when arguing for natural reserves.

3.2.7 Discussion

The land-use categories were one feature the biologists requested when discussing our first prototype with them. We added the land-use categories to the temporal abstraction view and can therefore support the biologists combining the graph view with the spatial context. We identified an important point during integrating land-use data. Depending on the season, the environmental changes are not reflected by the static land-use categories. Consequently, visualizing only land-use is too little to describe the environmental conditions an animal experiences. Especially, extreme landscapes as deserts for example are very dry regions throughout the year, although short rain periods provide large amounts of water in short time. Dynamical data sets reflecting seasonal changes might help to explain observed movement patterns.

Our presented process pipeline involves, first, a user-driven filter step, second, a density based clustering, and third, the combination and visualization of the results. We use the result of the geospatial abstraction as input to our second, temporal abstraction step. The identified convex hulls are being visualized as a graph network with active node being highlighted for each time

interval. Since each cluster contains both, spatial and temporal information, we use them to enhance the visual representation. A technique called Small Multiples is used to visualize the automatic temporal segmentation of the graph. The analyst can influence the segmentation process and adapt the result to his needs.

However, the analysis and visualization are highly dependent on the right parameter settings. First, the filtering step has influence to the semantics of the found clusters as only points fulfilling the filtering criteria are considered for clustering. Choosing the proper clustering parameters, e.g., *MinPoints* and *epsilon*, is highly application dependent and is not trivial. Furthermore, the analyst has to partition the whole time span into meaningful units by setting a threshold. This threshold depends on all the previous steps and not intuitive to control. Only by inspecting the visual results, the analyst can judge whether the parameters were set meaningfully. Consequently, a Visual Analytics expert is needed to assist the subject matter expert in his analysis.

3.2.8 Conclusion

We presented in this section two line simplification techniques tailored to geospatial movement data. In the first part, we described and compared pure line simplification methods with focus on expressiveness and effectiveness. Zoom level dependent simplification allows adopting the line granularity to the respective spatial resolution. We designed our algorithm in a way enabling interactive response to zoom level changes with linear runtime. Depending on the applied similarity measure during clustering the focus of the simplification will change. If only spatial distances are used, the simplification will only focus on density. If similarity measures for meta-attributes (e.g., sinuosity or speed) are applied, segments with same behavior will be grouped together and simplified by a straight line. The second part of this section described an approach to analyze often visited places with a certain property and the corresponding visiting sequences. The analyst can choose which property describes an interesting place by filtering and control clustering and partitioning the trajectory interactively. With all the proposed simplification and abstraction methods, we were able to derive some insights of the investigated animal movement data sets.

The techniques proposed in this section were quite general with respect to the application scenarios. We covered both long-distance and short-distance movements as well as territorial movement behavior. These techniques do not require the movement to fulfill certain criteria and are applicable to a wide set of movement patterns. Our approaches are quite general and

use only movement attributes as position or derived attributes. Consequently, our methods do not include domain knowledge and are less effective than highly tailored methods. In the next chapter, we will restrict the investigated movement data to soccer matches. As the application domain is restricted, we can tailor our methods and propose line simplification methods using domain knowledge and allow deeper insights.

I am seeking for the bridge which leans from the visible to the invisible through reality.

Max Beckmann

4

Application to Movement Data of Soccer Matches

Contents

We discussed in the previous chapters several methods to improve visualizations for both temporal and geospatial data. In the second chapter dealing with temporal data, we discussed visual boosting techniques, peak-aware prediction methods, and a visual analysis system for temporal power consumption data. We supported the analyst detecting and predicting interesting patterns and visually boosting interesting patterns appropriately. The latter chapter

dealing with geospatial data focused on an overlap-free representation of points and overlap-reduced, simplified trajectories represented by lines. In this chapter, we will extend the techniques discussed previously and discuss Visual Analytics techniques suitable for soccer data. Recorded soccer matches are spatio-temporal data with a high resolution containing complex movement patterns. We will apply some previously discussed techniques and propose methods specifically tailored to the soccer analysis scenario. The baseline for our presented methods is to support soccer experts and enable an efficient analysis process without having the user to dig for patterns by watching whole matches. We rather want to point out patterns and situations being possibly of interest to the analyst and additionally learn from user feedback. Consequently, we will discuss in this chapter different analysis and visualization techniques covering single- and multi-player analysis.

This chapter is taken with slight modifications and some additions from the following publication[1]:

Feature-Driven Visual Analytics of Soccer Data

H. Janetzko, D. Sacha, M. Stein, T. Schreck, D. A. Keim, and O. Deussen.

Proceedings Visual Analytics Science and Technology, 2014. [JSS+14]

4.1 Preface

The visual analysis of soccer data is interesting for two reasons: first of all it is scientifically interesting since it is an instance of a geo-spatial analysis problem with complex, interdependent trajectories and events. On the other hand, soccer is a very popular sport and actively played by

[1]In this work, Dominik Sacha took care of the single-player analysis and the Horizon Graph integration. Manuel Stein focused on the Visual Analytics multi-player analysis under my supervision. Tobias Schreck reorganized and rephrased the Related Work section. Tobias Schreck, Daniel Keim, and Oliver Deussen helped with fruitful discussions and advices. I did all the research and implementation work not mentioned above, basically implementing the analysis prototype, proposing Data Mining and visualization techniques, and performing a first expert study. The second, more exhaustive expert study was conducted by Manuel Stein for his Master thesis under my supervision. David Perlich did some basic implementations for the single player analysis during his Bachelor studies. Feeras Al-Masoudi implemented an interactive trajectory simplification for selected time intervals in soccer matches for his Master thesis. I had the idea to integrate Visual Analytics to the detection of interesting game events, to present the history of classification results, to visually compare several classifiers, and to develop an abstract line simplification method for arbitrary time intervals reflected in different abstraction levels. All sections that were not written or rephrased during the paper writing process by myself are quoted.

approximately 270 million people [FIF15]. Soccer plays a huge role in public media coverage and also, poses analytical needs by sport decision makers. Recently, GPS- and video-based tracking technology became available which allows recording spatio-temporal data of players at high frequency and accuracy. The arising data is interesting to analyze for two main purposes:

- Scouts are looking for high-performance players, where performance needs to be assessed by many measurable parameters or combinations thereof, in relation to other players and play situations, and over time. For example, these attributes may be the accuracy of shots, the quality of passes, or the willingness to run in the last minutes of a long and exhausting match. Depending on these high-level attributes of a player, the underlying analysis must focus on different sets of basic features. The willingness to run, for instance, depends on the time of the match, the speed of the player, and the current game situation. If the player's team is already three points ahead of the other team, it is not that important to run very fast in the last minutes of the match.

- Coaches are analyzing matches to improve the overall performance. The analysis can be performed either in real-time, in the halftime break, or after the match. Depending on when the analysis is performed the focus is different, which has to be reflected by the analysis process. In defending situations, coaches are interested in dangerous situations, how they occurred and how the team resolved those. For instance, an analysis of the back-four formation can help in assessing the quality of the defense.

Soccer data is a representative of spatio-temporal datasets and therefore already inherently challenging. Compared to standard movement data, the spatial restriction of the movement stands out. Movement data of soccer matches is located on an approximately 105 by 68 meters pitch. As 22 players are moving in a relatively small area, the resulting data is very dense and difficult to visualize by a single visualization. Furthermore, the observed movement patterns are very complex as the movements of each player depend on the movement of all the other players. Nearly every movement action causes a reaction, because of the high interdependencies between all players. Compared e.g., to the flock movement of birds, there are two opposing teams aiming for different targets and trying to hinder the other team. Simple leaderships rules, as e.g., in flock movement theory, are therefore not applicable. To make matters worse, soccer is a very dynamic game as tactics and strategics change over time. Depending on the current game situation a team might, for example, switch their overall game-play from a defensive to an

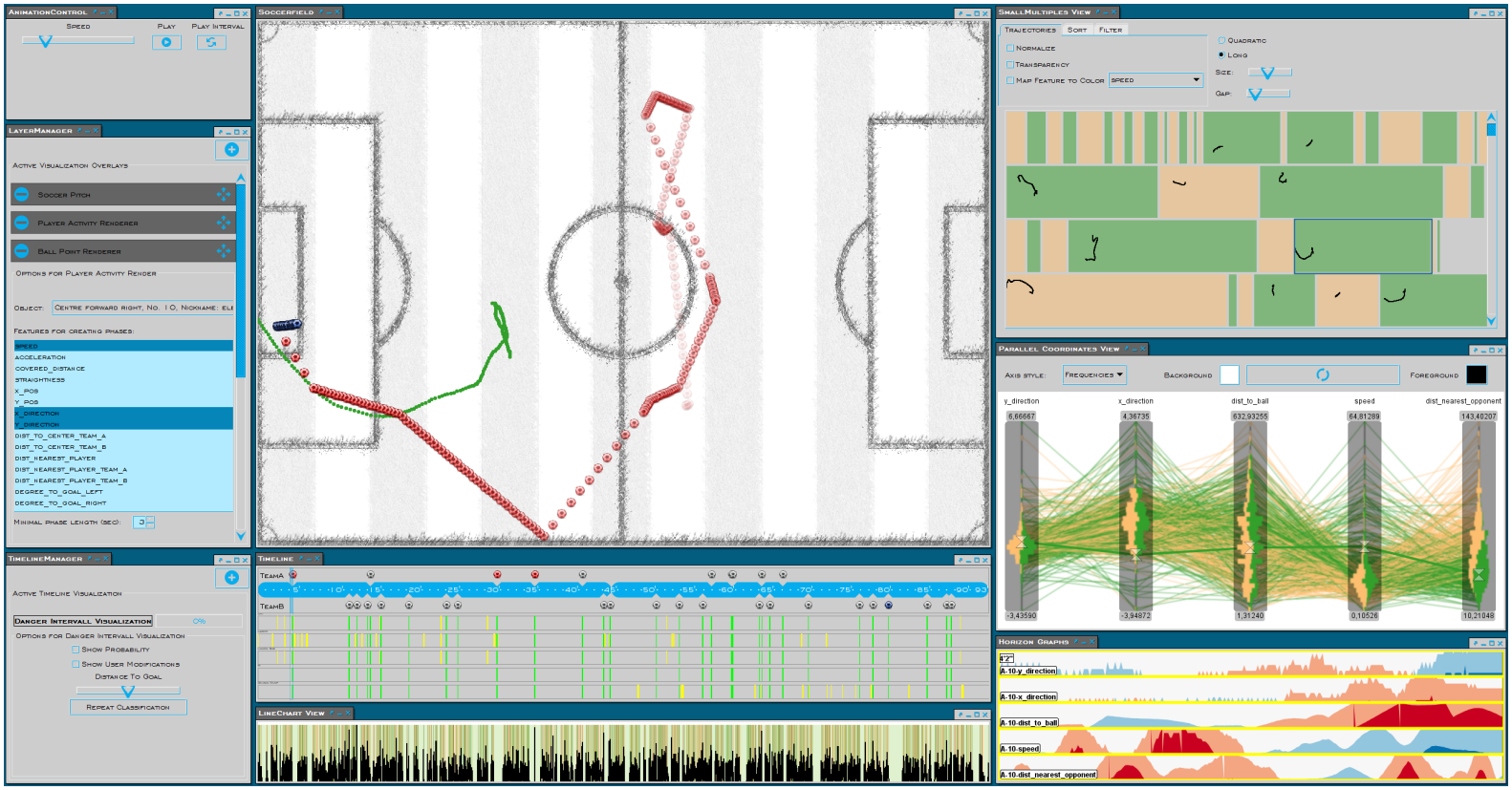


Figure 4.1.1: Interesting phases of a single player can be automatically found by applying the clustering approach presented in Section 4.3. In this case, we analyze a forward and are interested in the attacks in that the player was involved. Resulting phases can be inspected using the Small Multiples view (top-right panel) in combination with the other rendering layers and Horizon Graphs (left and bottom panels).

offensive one being directly reflected in the observed movement patterns. Even historically the formations changed from a sweeper up to the late nineties over the 4-4-2 formation to today's most often used 4-2-3-1 formation (see also Section 4.4.2). Though the outcome of the game, basically who wins and who loses, is not necessary reflecting superiority. Some goals or non-goals are lucky or due to incorrect referee decisions. But what is important when assessing games is why a team won or lost. In order to assess the quality of a team, it is important to take the respective context, e.g., strategy or movement patterns, into account. Switching from offensive to defensive gameplay when losing the ball for instance can be an important clue for coaches.

In this design study, we analyze soccer data with Visual Analytics methods using single-player, multi-player and event-based features. We apply feature analysis techniques to present the most important features to the analyst depending on the respective analysis task. We combine data mining techniques detecting interesting game events with interactive visualizations allowing immediate user feedback to the data mining process. Our focus is to help coaches in investigating interesting and dangerous game situations. There are two points to tackle when trying to support coaches. First of all, interesting game situations have to be identified and presented to the coach and, second, the coach should be able to analyze features of players with respect to these situations. Analyzing a certain game situation can be performed on different levels, such as taking only one single player into account or consider even several players. Our implemented prototype is depicted in Figure 4.1.1.

The remainder of this paper is structured as follows. We outline existing and related approaches and system in Section 4.2. The analysis of a single-player is described in Section 4.3, followed by a discussion of our multi-player analyses in Section 4.4. Furthermore, we perform event-based analyses in Section 4.5. We implemented a modular, layer-based system allowing an easy integration of existing data mining and visualization techniques, described briefly in Section 4.6 with further outlook to the Visual Analytics capabilities. Our design is validated by use cases and interesting findings in Section 4.7. We furthermore conducted an expert study to evaluate our design in more depth in Section 4.8. We finally conclude our paper and give an outlook to future work in Section 4.9.

4.2 Related Work

We first discuss related work in general visual analysis of sports data in Section 4.2.1, followed by specific works organized according to the considered analysis perspective in Sections 4.2.2 and 4.2.3. Section 4.2.4 positions our approach within the aforementioned works.

4.2.1 Visual Analysis of Sport Data in Research Interest

The visual analysis of data related to sports has recently come into focus of research and application [BCC+13]. The interest is seen driven by advances in acquisition of high-resolution sports data, and in advances in visualization and analysis of sensor and movement data. Sports analysis is expected to foster many new applications for end users, sports coaches, and sports managers alike [BCC+13]. Analytical goals in these applications include overview and comparison of player and team performance, prediction and correlation of behavior, and understanding changes over time on the short, medium and long term perspective. Commercial systems are very hard to compare to as the there are high financial interests behind the scenes. We had some discussions with a professional soccer analyst telling us that existing automatic approaches cover more or less only single-player statistics. In-depth team analyses are typically performed by manual inspection.

We just mention two of the most recent sport analysis systems here as examples, before surveying more in the following paragraphs. A recent work on visual analysis of sport data includes [LCP+13], where a visual search system for scenes in a Rugby match was introduced. The approach is based on the configuration of team players and their movement during a match, where this data is extracted by means of video analysis. The approach offers a sketch-based query processing for movement patterns extended by Visual Analytics methods. Instead of using movement sketches, we directly look at manually annotated important and dangerous situations and extract similar dangerous ones. We compute semantically meaningful features with respect to soccer and use them for our data mining process. Regarding soccer, by means of a design study, in [PVF+13] a tool was developed which combines different perspectives on soccer match data with the aim of creating play reports. The data set used included raw player positions and movement, as well as manually annotated match events like goals, fouls or ball contacts. Thereby, the match data was segmented into meaningful units, which can be visualized in different views. The matches were for instance partitioned by looking at shots and going back in time until the team gained the ball. We extend this work by detecting interesting event

and phases semi-automatically by integrating statistical features.

4.2.2 Movement and Constellation-based Analysis

In general, many approaches for sports analytics consider trajectories extracted for players and teams as a basic abstraction of the data to be analyzed. Consequently, methods of spatio-temporal data analysis are applicable [AAB+13b, AA06]. Important data analysis methods in this area include the segmentation, abstraction, correlation, clustering or classification of trajectories. Today, many applications for trajectory-based data analysis have been identified, including studying of traffic data [WLY+13], movements of pedestrians in office spaces [IWSK07], or analyzing eye tracking data in context of user studies [OAA+12]. Further applications of trajectory-based analysis include understanding of animal movements [SJM+11], or analysis of time-dependent measurements in a 2D diagram space [SBTK09, vLBSF13]. In general, key to successful trajectory-based analysis is finding a meaningful trajectory representation [AAB+13c].

The trajectory of even a *single* player can already be useful for sports analysis of a game, and it certainly is useful for measuring the performance of a given player. However, often also properties of *groups* of players are relevant. To this end, certain approaches first detect specific constellations among groups of players which may then again, be described by trajectories or other time-dependent group features. Examples for soccer analysis include [KKL11], where player formations are analyzed. Specifically, the spatial constellation between all defenders of one team are analyzed over time, which can reveal tactical maneuvers. In [FMT+13], the area on the field where a given player showed a particularly strong influence during the game, was identified. In [TH00], speed and direction were considered as features in such areas of interest. In [FS05], distances between player, puck and goal within hockey games were used as features of analysis. Further extensions of the approach of associated areas can be found in [KHL06, Kim04, NMMN10].

Other works detect specific scenes of interest during a match. In a work by Gudmundsson et al. [GW13], pass alternatives and their specific contextual difficulty are visualized. Furthermore, paths frequently taken by individual players are considered in that work.

4.2.3 Analysis Based on Temporal and Statistical Aspects

Besides trajectory-based analysis, also methods from time series and multivariate analysis are applicable to sports data analysis. In general, any relevant measure which is recorded over time

(including properties of trajectories) can give rise to time series analysis approaches [Ham94]. Examples include comparison and correlation of measurements among players, or analyzing for cyclic behaviors of measurements [AA06]. In addition, time-dependent measurements can also be aggregated by descriptive statistics such as mean, variance or other statistical moments of interest.

In [LPLBDG10], it was evaluated which statistic measures correlate with the outcome of a game. The temporal development of geometric statistics, like the convex hull, circumference, or center of a team were analyzed in [DAF+13]. Also, statistics were used in [DSBT+07] to differentiate between players of different positions. A number of commercial and academic software solutions for the analysis of statistical sports data exists. In [RSB11, RSB+10] an interactive statistical tool for coaches is introduced, enabling to analyze and compare players. Furthermore, domain-dependent tools exist e.g., Matchpad [LCP+12], CourtVision [Gol12] and SnapShot [PSBS12].

Statistical measures can, among other transformations, be defined based on a relational perspective on data: Passing networks can be seen as a rich source for investigating soccer matches even further. Then, statistics can be extracted from a network (or graph-based) representation of the data. E.g., in ball sports, the passing network indicates which player passes the ball to which other players over time. In [PT12], the performance of players is measured by aggregates of the ball passing network. In [DWA10], additional nodes for "shots to goal" and "shots wide" are added to the passing network description.

4.2.4 Summary and Positioning of our Work

We distinguish two classes of analysis of sports data. Approaches based on low-level features extract measurements from movement or other sensor data and perform statistical and correlation analyses on the (possibly, pre-processed) data. On the other hand, approaches being oriented toward higher-level representations, such as semantic annotations of data, exist. These can stem, e.g., from manual annotation by human experts or crowds; or by recognition of specific constellations of interest, based on heuristics or Machine Learning approaches.

The work most closely related to ours is [PVF+13]. Similarly, we present an interactive system for explorative analysis of soccer data. Our system is flexible in that it incorporates both low-level features (based on trajectory features, see Section 4.6) and semantic annotations (based on recognition of play configurations, see Section 4.5) for the analysis. Our system flexibly al-

lows to draw on either of these analysis perspectives, based on the user task. Our semi-automatic selection of features helps to cope with the otherwise difficult problem of feature selection by users. We achieve this by incorporating a user-configurable classifier which allows detecting further events in the movement data, based on a number of example events and input features. Thereby, our system is not limited to detect a certain number of pre-configured situations, but helps in configuring detectors for many events of interest.

4.3 Single Player Analysis

The single player analysis investigates the performance and features of one player at a time. We want for example to detect when a player is actively participating during a match. Certain player features, as speed or distance to the ball, will be of use for this kind of analysis. More abstract, we divide the different behavior and motion patterns of a player into different phases. The features being relevant for a single player analysis can be divided into three categories: *Individual Characteristics* (e.g., coordinates and speed), *Game Context* (e.g., distance to ball), and *Events* (e.g., shots, receptions and fouls) features. These features can be seen as numerical time series with changing values over time, with events transformed into a binary time series with singletons.

In order to segment the match into different phases, we apply clustering and hereby detect similar phases. Phases derived from the clustering results should be as homogeneous as possible with respect to the underlying numerical features. The overall analysis process in depicted in Figure 4.3.1. We first partition all time series into small, fixed-size intervals and aggregate the values into a numerical feature vector describing the respective time interval. The values are linearly normalized to avoid any biases during the distance calculation. Additionally, we can apply dimension reduction techniques such as PCA to remove noisy dimensions if necessary. The PCA is performed by WEKA [HFH+09] automatically reducing the number of dimensions with a threshold of 95 percent of the variance being still explained. The intervals are afterwards clustered resulting in a certain number of clusters (depicted by small letters in Figure 4.3.1). In our analyses, we apply k-Means (allowing us to control the number of resulting clusters) and DBSCAN (being a robust clustering technique with respect to noise and outliers). Finally, we merge similarly clustered and adjacent intervals to phases.

We visualize the analysis results using colored trajectories, line charts, parallel coordinates [ID91] and Small Multiples [TGM83] all linked via Brushing & Linking. In Figure 4.1.1, we

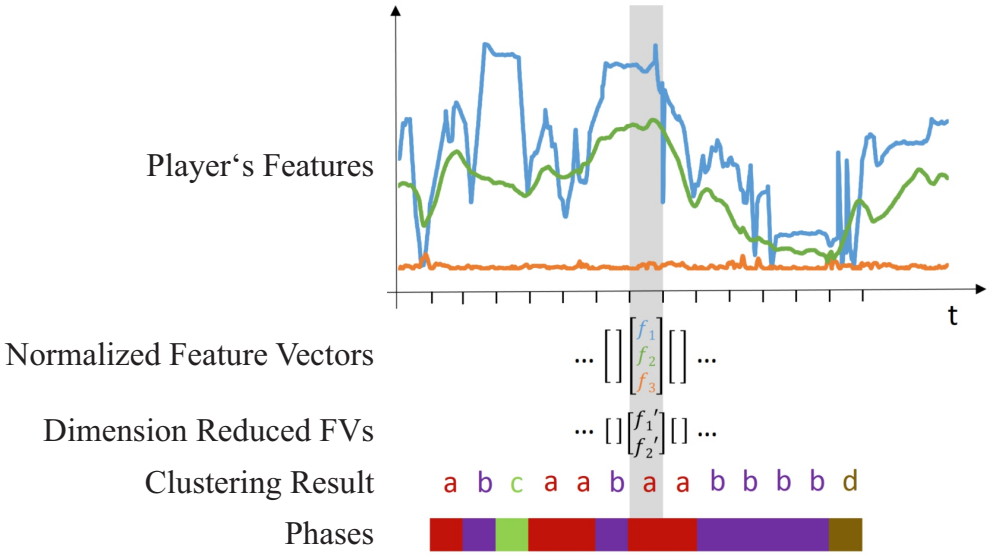


Figure 4.3.1: Feature-based approach to detect similar activity phases of a single player. Reprinted from [JSS+14].

present the visual interface showing the results analyzing a forward described in more detail in Section 4.7.1. It is very crucial in the analysis process to understanding the semantical meaning of found clusters or phases. We therefore integrated several views onto the segmentation results and the human analyst can bring in his expertise.

A first overview is provided by a line chart with a freely selectable feature and background coloring depicting the phases (bottom of Figure 4.1.1). Parallel coordinates help to understand the distributions of feature values in the respective clusters. Finally, Small Multiples offer several interaction possibilities like filtering, sorting (according to a feature, phase similarity or time), or visualization options (e.g., mapping feature values to the trajectory's color).

A typical workflow for this kind of analysis is shown in Figure 4.3.2. We start specifying parameters including the clustering parameters and features to regard during the segmentation process. Afterwards, the analyst uses the line chart and parallel coordinates in combination with the Small Multiples view allowing filtering, highlighting and inspecting phases. Furthermore, all other implemented visualization layers can be applied to analyze the selected situations of the soccer match in more detail. As our system is interactively reflecting changes to the clustering settings, all steps of the workflow may be revisited several times.

The resulting clusters and phases have to be semantically interpretable for a successful anal-

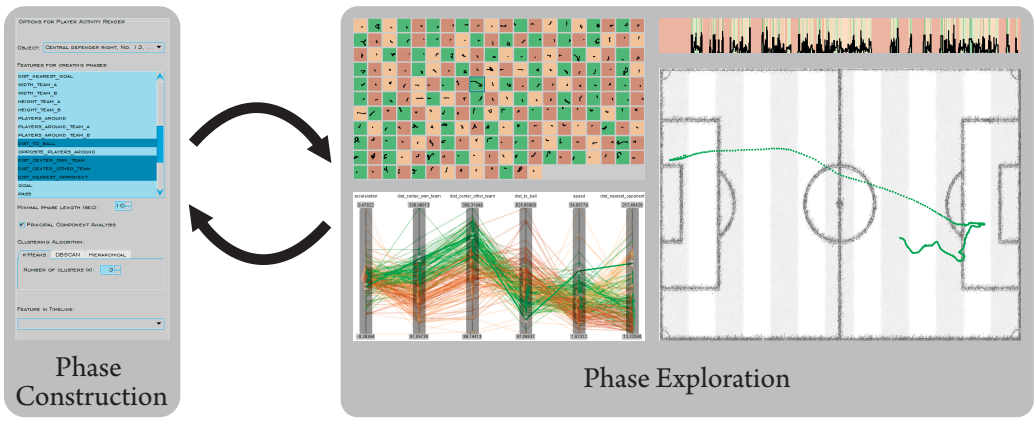


Figure 4.3.2: Schematic workflow for the analysis of a single player.

ysis. We support the analyst by visualizing each phase and the corresponding averaged feature values in a parallel coordinate visualization. We integrated several interaction and visual boosting techniques. In Figure4.3.3, we show a schematic depiction of our parallel coordinates implementation for two dimensions with focus on the filtering capabilities. We added to each dimension interactive range selectors providing the following interaction possibilities:

(I) The upper boundary of the respective selection can be either moved individually or all upper bounds can be moved simultaneously.

(II) Instead of increasing or shrinking the selected range, the analyst can just drag the selection range on the axis up and down.

(III) The lower selection limit can be either dragged individually or all lower limits can be moved simultaneously.

Allowing the user to simultaneously change all upper and lower limits helps in performing manual nearest-neighbor queries. In this case, the filter intervals would be initialized by the system to fit one single, user-selected parallel coordinates line. The analyst is then able to adjust all upper and lower filter boundaries simultaneously. The filtering results are presented applying blurring techniques based on our previous discussion on visual boosting in Section 2.1. All lines in the parallel coordinates plot fulfilling the filter criteria (denoted by the blue hatched area in Figure 4.3.3) are drawn unblurred. All other lines are blurred to guide the analyst's awareness to the filtered ones.

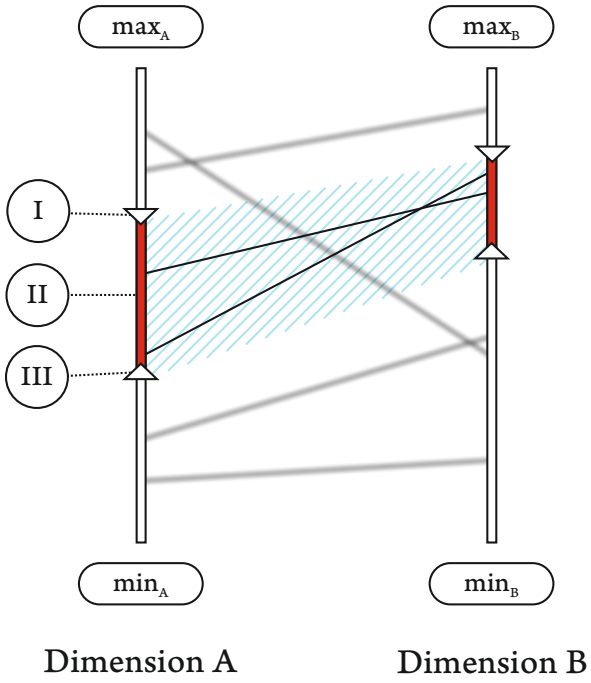


Figure 4.3.3: Schematic parallel coordinates implementation showing the filtering capabilities and the visual presentation of the filtering results.

Overplotting of lines is a common problem occurring in parallel coordinates visualizing medium amounts of data. We employ parallel coordinates to investigate segments of trajectories being clustered. In order to semantically interpret the clustering results, it is crucial to be able to explore the value distributions for the different segments and clusters. We consequently support the user in investigating the parallel coordinates plot by integrating a stacked bar chart visualization. We visualize the frequency distribution of clusters along a dimension axis as exemplified in Figure 4.3.4. Although the implemented visualization technique have drawbacks concerning scalability and readability, the analyst get a feeling for the feature distribution of clusters and interdependencies of dimensions. This technique is similar to the work presented by Hauser et al. in [HLD02]. We will investigate one application scenario in the subsequent use cases Section 4.7.2.

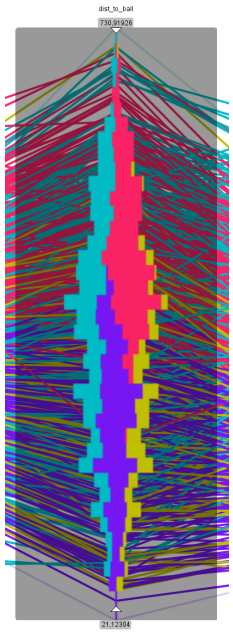


Figure 4.3.4: Example frequency visualization for parallel coordinates plot. The cluster frequencies are computed and visualized along the dimension axis.

4.4 Multi Player Analysis

Regarding more than only one player in the analysis process is very important as soccer is a team sport. This section introduces our methods for the analysis of soccer matches with respect to

the movement patterns of multiple players.

4.4.1 Player Comparison

We enable the analyst to compare several players visually by providing Horizon Graphs [HKA09] for selected players and features. In Figure 4.4.1, we show the speed of all field players of one team in the first three minutes of a soccer match. The correlation and the similarity of the speed feature is clearly visible. There are phases with high speed (blue) and also phases with almost no speed (red) showing that the players act as a team. The bottom player can be seen as an outlier to the coherent movement behavior. The bottom Horizon Graph represents a forward who does usually not participate in all defense actions explaining the observed pattern.

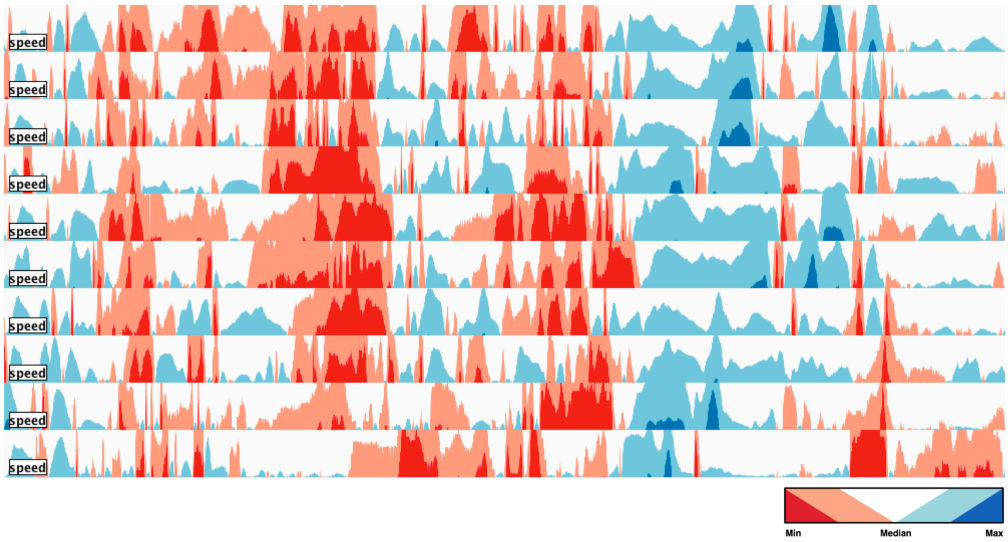


Figure 4.4.1: Speed feature of all field players of one team in the first three minutes of a soccer match. Reprinted from [JSS+14].

We furthermore extend the single-player segmentation process described in the previous section towards a multi-player analysis. The combination of phases together with the possibility to inspect selected features visually can reveal interesting patterns. In Figure 4.4.2, we analyze for example two central defense players. The trajectories are colored by detected phase and the speed features are visualized by Horizon Graphs for the selected time interval (blue rectangle in the timeline). Interestingly, both defense players act very similar, which is reflected in both, the movement features and the phase coloring.

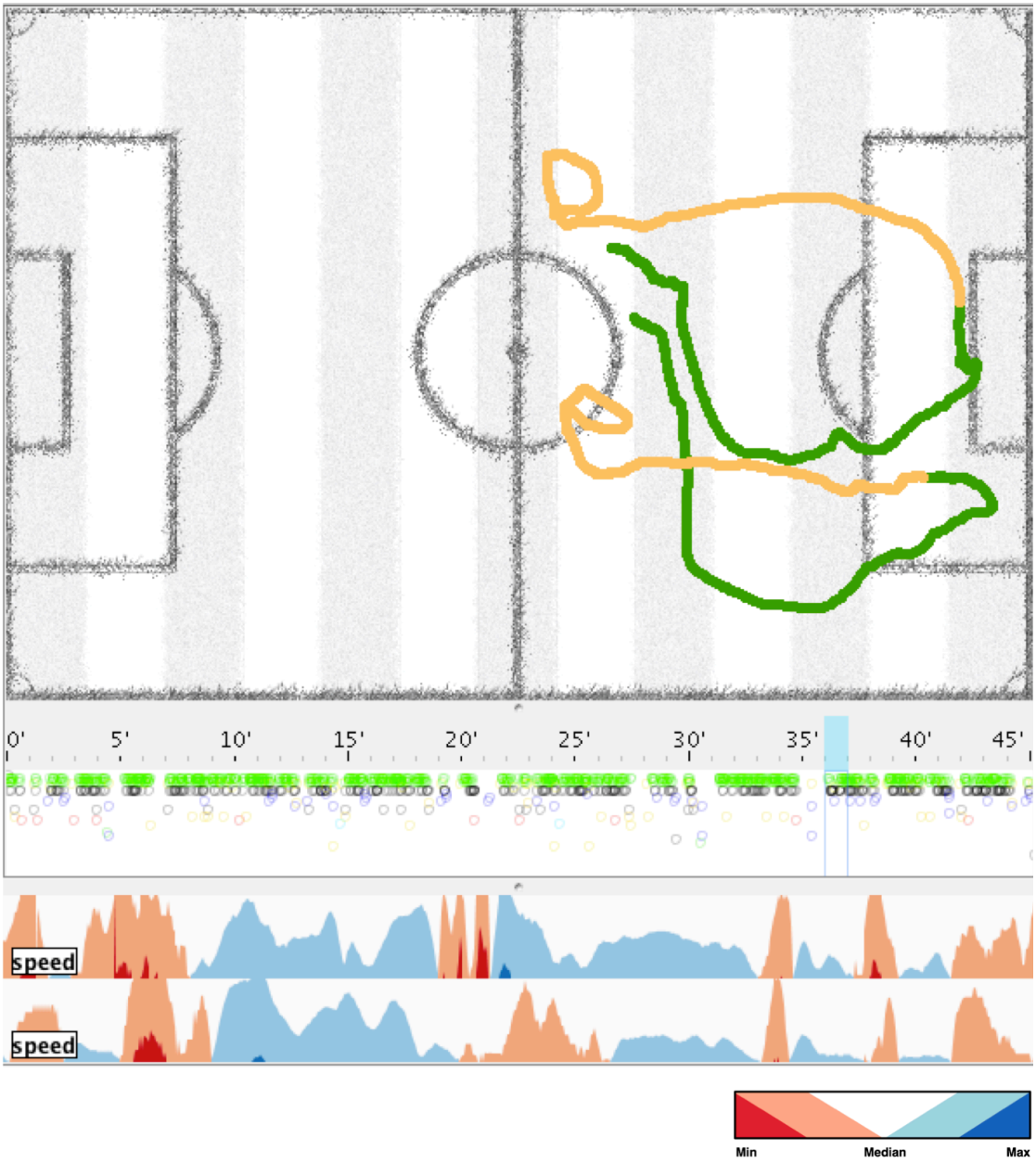


Figure 4.4.2: Activity phases and trajectories for two defense players. Reprinted from [JSS+14].

We help the analyst in selecting interesting features to visually inspect by predefined sets of features for the analysis of certain event types. Further details can be found in the use case Section 4.7.

4.4.2 Constellations and Formations

In addition to low-level and statistical trajectory features, the analysis of spatio-temporal team formations in soccer games is very important. This is because formations reveal semantically meaningful patterns and may relate to tactics or strategies of the teams. Formations tell us more about tactics than single player analysis. There exists a variety of formations in modern soccer, like the nowadays very widely known and used 4-2-3-1, the 4-4-2 (a.k.a. "Diamond"), or the 4-3-2-1 (a.k.a. "Christmas Tree") formation. In this section, we focus on the analysis of the defensive lines and more specifically on the back-four formation. Other defensive structures, such as the defensive triangle, could be also easily automatically assessed. Further descriptions of different formations can be found in [Wik15].

The crucial point when analyzing the back-four formation is to assess the quality with means to the defensive effectiveness. We therefor need a definition for a good and a bad back-four formation. The main task of the back-four formation is to defend their own goal. Nowadays, zonal marking is the widely used defense strategy. Consulting soccer literature and training handbooks, we found some criteria how the back-four formation should react to attacks [CDH12]. There exists an *ideal line* parallel to the ground lines of the pitch, where all back-four players should be. Basically all players should be on the same height, which is also very relevant for the offside trap. Scoring the defensive formation is then simply computing the average distance from the ideal line. However, there exist different kinds of attacks that have to be dealt with differently, resulting in a more complicated assessment. Incoming attacks can be differentiated by the following criteria: As long as the distance between ball and goal is larger than 24 yards, the back-four formation shall use the ideal line described above. If the ball is closer to the goal, we will have to distinguish between an attack from the middle and one from the side. Attacks from the side should be answered by a sickle-like formation. Further details can be seen in this Youtube video [You15]. From the computational perspective, we need to check the curvature between outside and central defender of the respective side. Furthermore, the distance between central and outside defender must not be too big, because the outside defender might need help. The defenders of the side which is not attacked should then build an ideal line reflecting

the positions of the other defenders. Defense triangles are the correct reaction to attacks from the middle. The computational assessment is performed by angle computations between the affected defenders. We score the defensive triangle by computing the angles and also include the distances of the involved players. Figure 4.4.3 shows two examples for a bad and a good back-four formation.

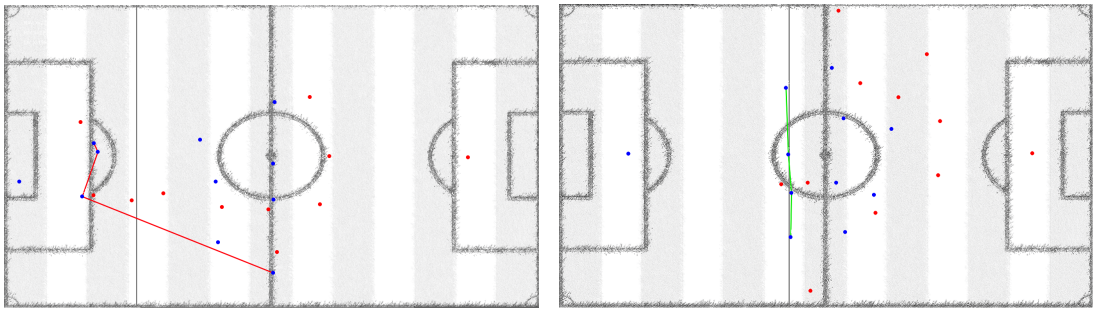


Figure 4.4.3: An example for a bad (left) and good (right) back-four formation evaluated based on the ideal line. Reprinted from [JSS+14].

4.5 Event-Based Analysis

Soccer matches are not only continuous movements of players, but there are also incisive events. Besides goals and fouls there are also events like passes or crosses. These events are manually annotated and added to our datasets. We use these events as a basis for event-specific feature pattern exploration. We support two modes of analysis within our system. The first visualizes the development of selected features around user-chosen event types. The second analysis applies a classification technique to support discovery of previously unnoticed candidate events of interest.

4.5.1 Interactive Feature Analysis

If the analyst wants to analyze a certain kind of events, we visualize features in a time frame around the events with Horizon Graphs. Player specific features are derived from the involved players and additionally game context features as ball specific features are available. We render for each feature and event a single Horizon Graph, and lay them out in a tabular way. A line within each visualization indicates the time point when the event occurs. We included also

Brushing & Linking to enable the selection of single events being reflected in all other shown visualizations.

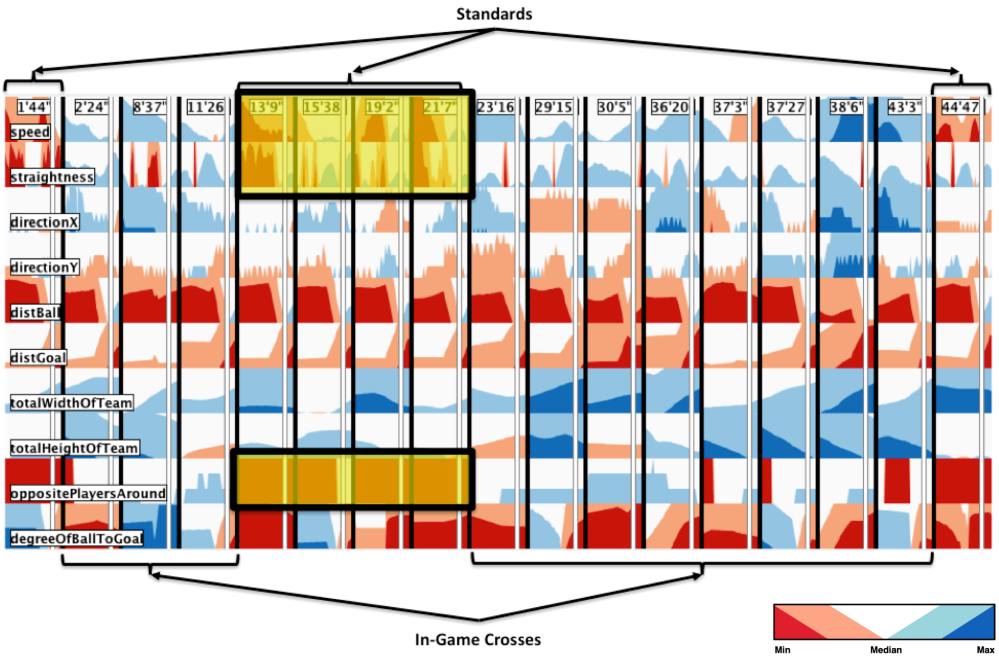


Figure 4.5.1: Features for all crosses occurring in one half of a match. Standard crosses like corners or free kicks can be clearly distinguished from crosses that happen within the match. Before a standard cross speed and straightness are similar lower than other crosses and there are almost no opposite players around the ball. Reprinted from [JSS+14].

Figure 4.5.1 illustrates as an example Horizon Graphs for all crosses occurring in one half of a soccer match. Feature patterns for standard crosses like corners or free kicks are visible as opposite players are typically not near of the executing player. Furthermore, the speed of the executing player is very low at the beginning of the interval, as the player is waiting until he is allowed to perform the free kick. This visualization serves also as a verification for the similar phase analysis presented in the next section.

4.5.2 Similar Phase Analysis

Using manually annotated data comes with the advantage of human knowledge being added to the data. Though at the same time, there is no guarantee that all events have been detected. Manually annotating data is a tedious and expensive task but very common in the soccer do-

main. Video analysts are employed to analyze soccer games and find interesting phases. We try to learn algorithmically from the annotated data to propose a set of similar and therefore potentially interesting phases. Our approach follows the subsequent points:

1. Investigate, which features are important for the classification process.

2. Explore and evaluate state-of-the-art classifiers to apply only the most promising classifiers.

3. Employ the top five classifiers and integrate a Visual Analytics feedback loop to steer the classification process.

We focus in this work on important events as shots on goal, fouls, crosses, and assists. We analyze how specific features, some related to only the involved player and some related to the team, develop right before these events over certain time intervals (2, 5 and 10 seconds). We use classifiers to detect similar phases in our data and validate the new found events in our tool as described further in Section 4.6.4. We use KNIME [BCD+07] as a state-of-the-art data mining framework for first experiments. Decision trees were used to get a hint, which features are important for the classification process. We applied all widely used classifiers as Neural Networks, Decision Trees, Probabilistic Models, and Support Vector Machines. Evaluating the classifiers by n-cross-fold validation, we came up with five classifiers performing best.

4.6 System

In this section, we describe the developed components of our system more technically. Our developed Java prototype for the analysis of soccer data is depicted in Figure 4.1.1. We implemented a layer-based soccer-pitch visualization, with several visualization techniques available (e.g., player position renderer, phase renderer, and heat map). The visualization layers can be added interactively and the order and further parameter settings can be controlled by a control panel. Furthermore, we integrated a timeline visualization and additional panels related to the analyses described in the previous sections. We designed the system in a modular and expandable way in order to enable an easy development of new layers or visualizations being connected to all the other components.

Single Player Analysis	
Speed	Acceleration
Position	Direction of movement
Distance covered	Straightness
Distance to next opposite	Distance to ball
Distance to own team center	Distance to opposite team center
Multi Player Analysis	
Width of team shape	Height of team shape
Opposite players around player	Back-four formation
Event Based Analysis	
Shots on goal	Passes
Fouls	Off-site
Cards	Reception
Goal	Clearance
Running with ball	Assists
Game Specific Analysis	
Ball-goal distance	Ball position
Angle of ball to goal	

Table 4.6.1: Features implemented in our system.

4.6.1 Features

Most of our visualizations and analyses rely on different kinds of features (see previous sections). These features are extracted, derived, and finally delivered to all other components. Player-specific features are computed and available for each player. Furthermore, team- and ball-related features are calculated as well. In Table 4.6.1, we list all features that are already implemented and available in our system. The extension of this list is an ongoing process triggered by new use cases and analysis needs emerging by prototype usage and expert interviews.

4.6.2 Visualization Components

Our prototype offers several panels where visualization can be plugged into and also provides synchronization functionality between the components. The analyst can control the currently visualized time windows by using the timeline component showing the selected time interval and event occurrences. We furthermore developed a layer manager where several layers can be registered and rendered on a soccer pitch area simultaneously. For each layer it is possible

to integrate an option panel handling the layer's configuration (e.g., clustering parameters). Finally, we offer a feature export component allowing to export features based on selected players, events, or time intervals. We make use of the export capabilities integrating external software components, described in more detail in Section 4.6.4.

4.6.3 Visualizations

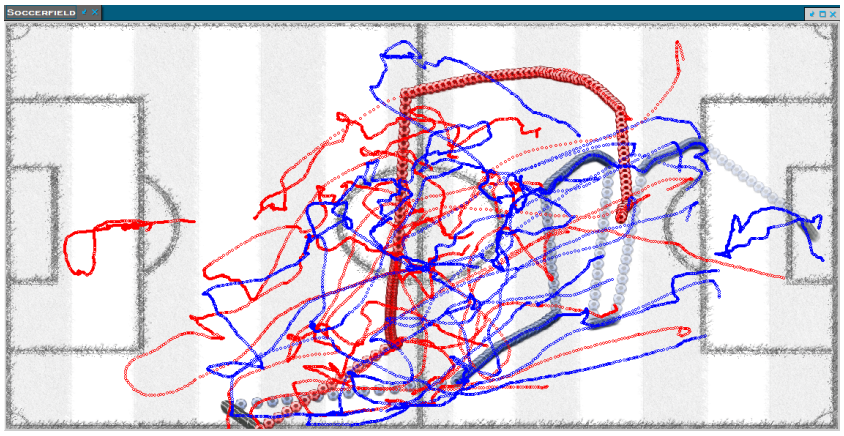
Depending on the analysis task, we provide different visualizations. Most of the visualizations are realized as layers that can be drawn on a soccer pitch. In order to get details of a soccer scene, we offer a player and ball renderer visualizing a selected scene. For larger time windows, we provide a heat map that can be computed for every spatio-temporal object (e.g., player, ball, event position). Selected features may be analyzed through line charts or horizon graphs. We provide specific views being useful in combination with each other. For example, the single player analysis view consists of the colored trajectory on the soccer pitch, the Small Multiples view, a colored line chart and the parallel coordinates plot. Another example is the back-four formation layer that renders formation dependent lines and colors on top other layers and also adds information to the timeline component.

We described in the previous chapter techniques simplifying lines reducing the amount of overplotting. We implemented line simplification also in our Visual Analytics system enabling the analyst to better investigate the ball movement. The raw movement is simplified by only showing the players directly involved in ball interactions and furthermore reduce the details. In Figure 4.6.1, we show our implemented line simplification approach. We omit all player movement not being directly interacting with the ball. Furthermore, we differentiate between passes (lines consisting of small triangles) and dribbling (wavy lines). For further details, the analyst can hover over players (circles with numbers) and see their movement of the selected time window. The time window visualized in Figure 4.6.1 starts with a pass from the blue goalkeeper (rightmost player) and ends with a pass to a red attacker.

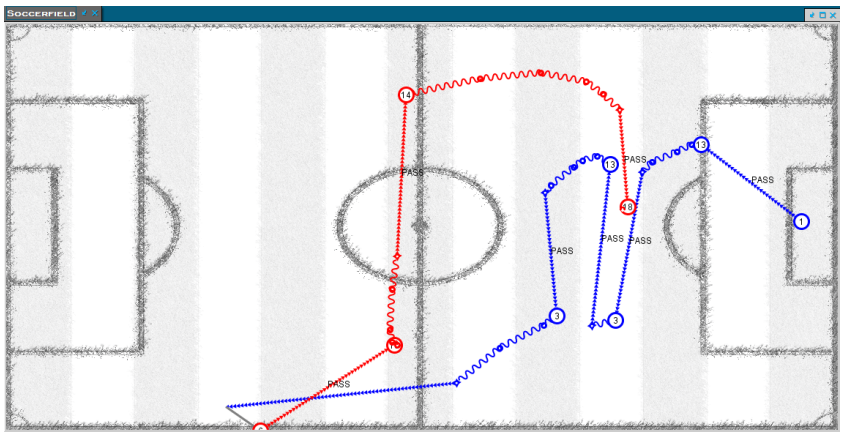
4.6.4 Similar Phase Analysis Facilities

This section briefly describes how our system integrates analysis functionality detecting similar events.

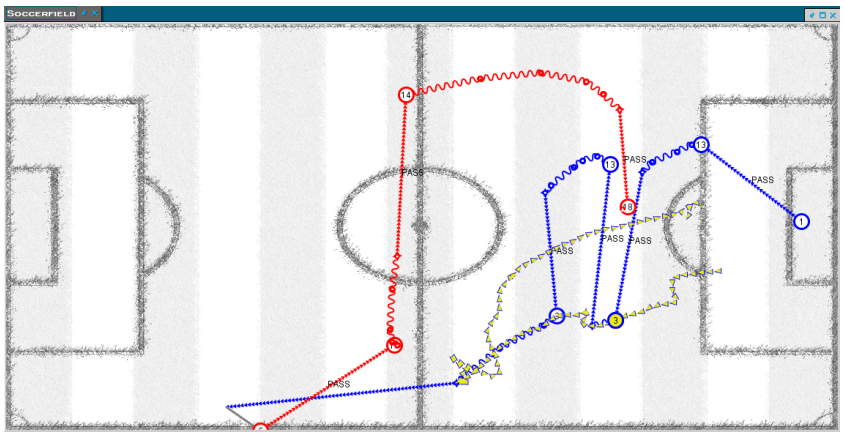
WEKA Clustering. We used the WEKA-library [HFH+09] in order to support state-of-the-



(a) original trajectories of players and ball



(b) simplified movement



(c) details on demand for hovered player (yellow)

Figure 4.6.1: Line simplification applied to soccer movement (top). We focus on the ball movement and represent only ball interacting players (middle). Details on demand are enabled by mouse hovering with the movement of the selected time span of the respective player being shown (bottom).

art analysis techniques. WEKA takes care of the cluster analysis described in section 4.3. We integrated the clustering components K-Means, DBSCAN, and hierarchical clustering for the single player analysis. The classification capabilities of WEKA are used in KNIME for the data mining part of our Visual Analytics pipeline.

First Visual Analytics Integration and Machine Learning with KNIME. As stated above, we are interested in gaining knowledge from investigating features of annotated events. We want to study which features and values are significant for different kinds of events. Furthermore, we want to use this knowledge for finding new events that were not annotated but can fulfill the found criteria. We set up a KNIME workflow and integrated the workflow into the analysis process depicted in Figure 4.6.2. We export all extracted and computed features into the KNIME workflow and partition the time series data into fixed-length intervals. Intervals including an event are marked as class A, while all others are marked as class B. After preprocessing, we train all available KNIME and WEKA classifiers with a 33% data sample and evaluate with the remaining data. We take the best five classifiers (LMT, LibSVM, Logistic Base, FT, and Decision Stump) according to their accuracy measured by their confusion matrix. The accuracy of the best classifiers ranges from 72 to 90 percent. We consider for our decision also the amount of false positives, which should be reasonable. False positives indicate new potential interesting intervals not yet annotated in our data. The classification results are then imported back into our prototype allowing the analyst to investigate time points labeled as class A. Furthermore, we integrate a feedback loop enabling the analyst to confirm found, previously untagged events and use them as additional training data for the classifier. This feedback loop may be repeated as often as the analyst wishes to.

Integration into our Visual Analytics System. Our next step is to integrate the Data Mining part tightly into our Visual Analytics system for several reasons. We used the knowledge gained from our experiments with KNIME and implemented the resulting, final workflow in our system. We still use WEKA, but preprocess the data and invoke the classification directly in our system. The first advantage of the integration is that there is no need for export and import steps anymore. The second and more severe benefit is a strongly increased performance. By self-implementing the analysis process, we could speed up the classification from 20 minutes to less than thirty seconds. The speed up was achieved by temporary data sets suiting the needs of WEKA and by threading adjusted to the respective number of processor cores. The tight integration allows us furthermore to reuse the trained classifiers for new matches not seen before.

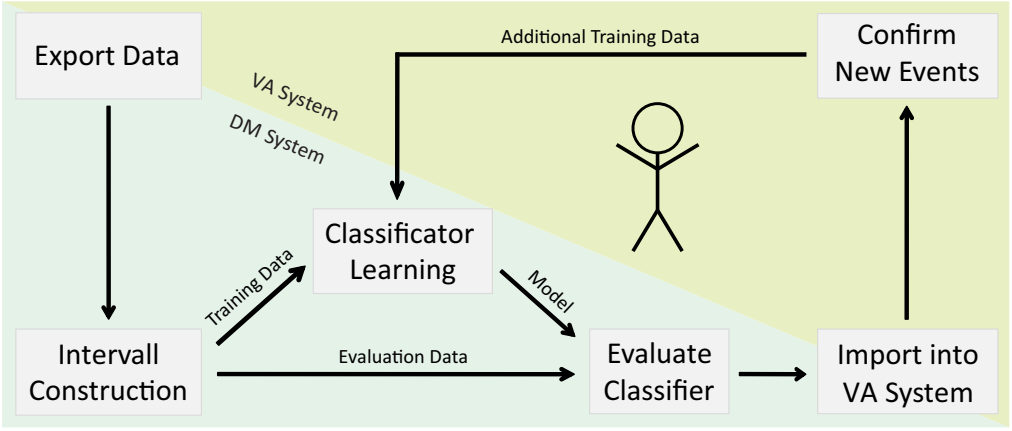


Figure 4.6.2: Analysis process for the detection of similar events and feedback loop to the classifier. Import and Export to KNIME needed for first experiments. Reprinted from [JSS+14].

In Figure 4.6.3, we present the process pipeline after the integration. Note that the resulting pipeline is basically the Visual Analytics pipeline reflecting our Visual Analytics claim.

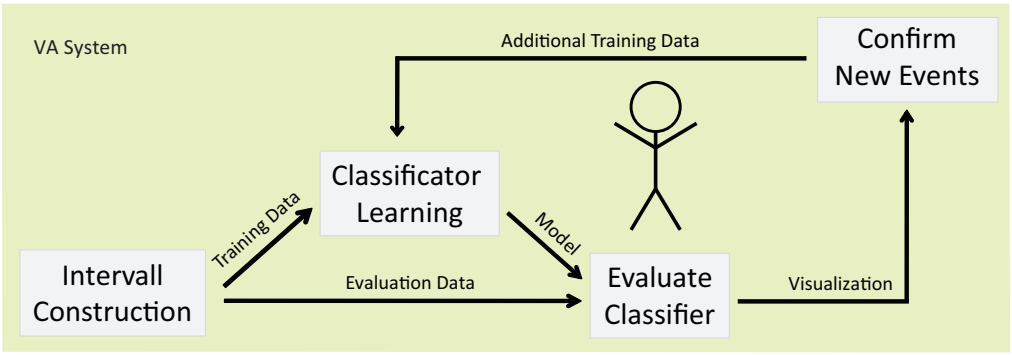


Figure 4.6.3: Analysis process for the detection of similar events and feedback loop to the classifier. Integration of the classification process into our prototype.

4.6.5 Interaction and Animation

Every developed component offers several interaction possibilities allowing the analyst to steer his analysis. Linking & Brushing is supported among all visualizations enabling multi-view data exploration. Besides mouse interactions and parameter setting controls, we provide common keyboard shortcuts in order to facilitate power user operations (e.g., animation control). Ad-

ditionally, animation of selected soccer scenes turned out to be useful in order to verify results or to understand and investigate longer phases avoiding overplotting issues.

4.7 Use Cases

In this section, we demonstrate in applicability of our prototype in different analytical use cases. We present several analyses and findings in which a domain expert could gain further knowledge about his team. We analyze a single player, detect similar situations in the soccer game, and investigate team formations as the back-four formation.

The data analyzed in our use cases is provided by prozone/mastercoach. The data set is not publicly available and was anonymized as it was a professional game. For each of the 22 players timestamped, two-dimensional position data are available with a temporal resolution of 100 milliseconds. Furthermore, the data includes manually annotated events containing information about position, time, and event-specific information as the involved player. These events are less frequent and lack in accuracy as they are manually tagged.

The use-cases were designed to show how our prototype can help coaches in analyzing the offensive and defensive qualities of their team. We will first analyze a single player and focus on his active phases. Afterwards, we investigate the offensive gameplay and the defensive back-four formation. These use-cases reflect some of the most important training aspects for a successful training, basically the attacking and defending skills. The last paragraph will cover some expert feedback we received when showing the tool to a subject matter expert.

4.7.1 Analysis of a Forward

Grouping and clustering interesting phases of a single player can be performed automatically by applying the clustering approach presented in section 4.3. In this use case, we analyze a forward and are interested in the attacks where he was involved. Therefore, we select the features *Speed, Direction of Movement (x and y-dimension), Distance to nearest opposite Player, Distance to Ball* and apply a k-Means clustering with two desired clusters in order to divide interesting from non-interesting phases. The resulting phases can be inspected using the Small Multiples view in combination with the other rendering layers and the Horizon Graphs. In Figure 4.1.1, we show the analysis results of the forward's attacks.

If we want to investigate the two clusters, we will use the parallel coordinates plot showing the feature values for all phases. Though the labels of the parallel coordinates plot show the

original data space, we used normalization before applying clustering. Obviously, green phases are defined by large distances to the ball. These green phases are uninteresting phases which we can ignore in our analysis. The uninteresting green phases can be hidden from the Small Multiples view to focus only on the interesting phases. As a next step, we take a closer look at the interesting (orange) phases where the player was very active and near to the ball. We sort the Small Multiples according to his x-position in order to see the phases where the player was closest to the opposite goal first. Selecting one Small Multiple will make all other components showing the selected phase. Figure 4.1.1 shows the third phase the system found, in which the forward receives the ball after he started to sprint and scores his first goal. The player is rendered by an orange trajectory and the phase can be animated as well. As a next step, the coach could inspect the other phases or arrange the Small Multiples by similarity in order to find similar patterns. Another option would be to explore the player's features using horizon graphs as described previously.

4.7.2 Feature Analysis for Defender Movement

We introduced in a previous section our parallel coordinates implementation allowing interactive filtering and additionally visualizing the cluster distribution on each axis. In this section, we will investigate the clustering and segmentation results for a defender. We clustered the movement data using the following four dimensions: speed, acceleration, distance to ball, and distance to the nearest opponent. We applied k-Means clustering with a desired cluster number of four. The resulting phases are depicted in Figure 4.7.1 with color representing the four clusters.

Without any further visualizations, the analyst is unfortunately not able to interpret the clusters completely. Nevertheless, there are some patterns visible by coloring the trajectory according to cluster membership as shown in Figure 4.7.1. From a spatial perspective, the defender stays always on his assigned right side. More interesting and insightful is that the purple phases seem to be the only ones occurring around the own goal. All other clusters are mostly located outside the penalty area. We will further discuss this finding when analyzing the corresponding parallel coordinates visualization.

There seems to be no clear spatial explanation for the other three clusters (red, yellow, and turquois). For this purpose, we integrated parallel coordinates visualization and enhanced them by a distribution visualization introduced previously. We visualize all phases of the defender's

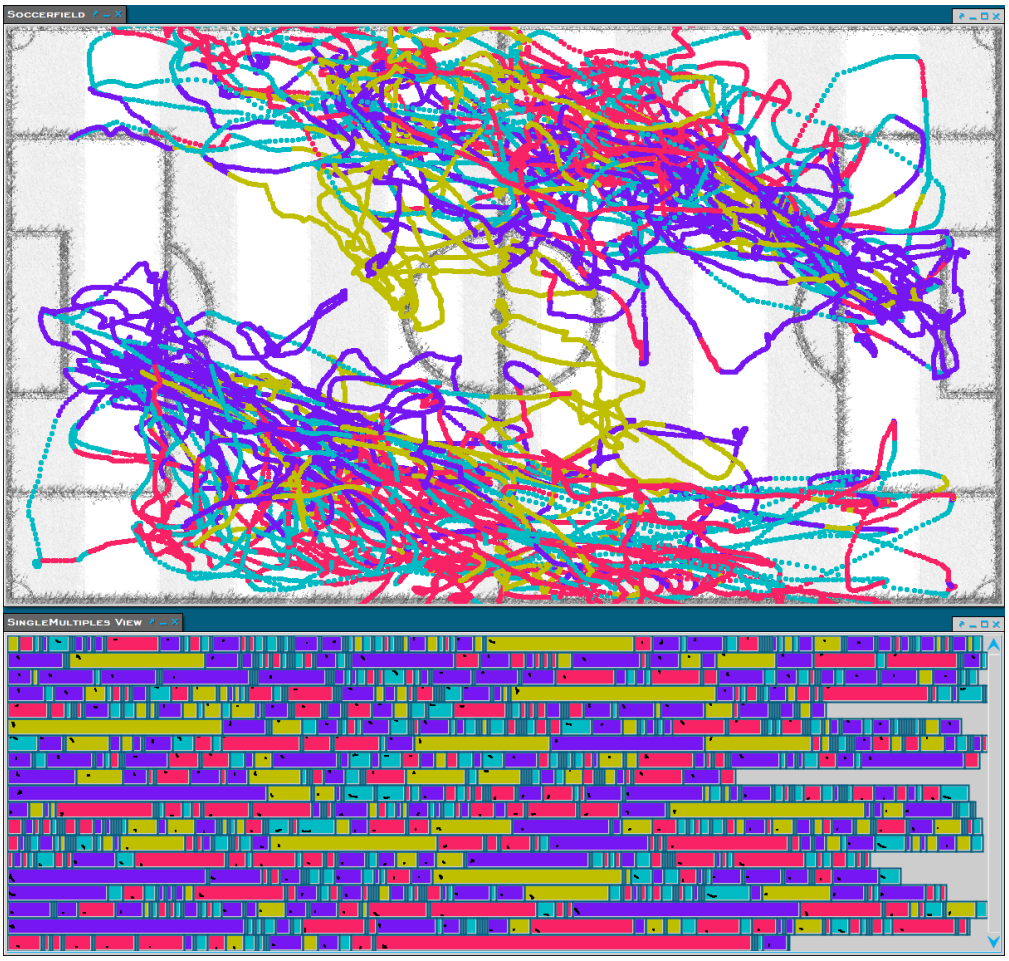
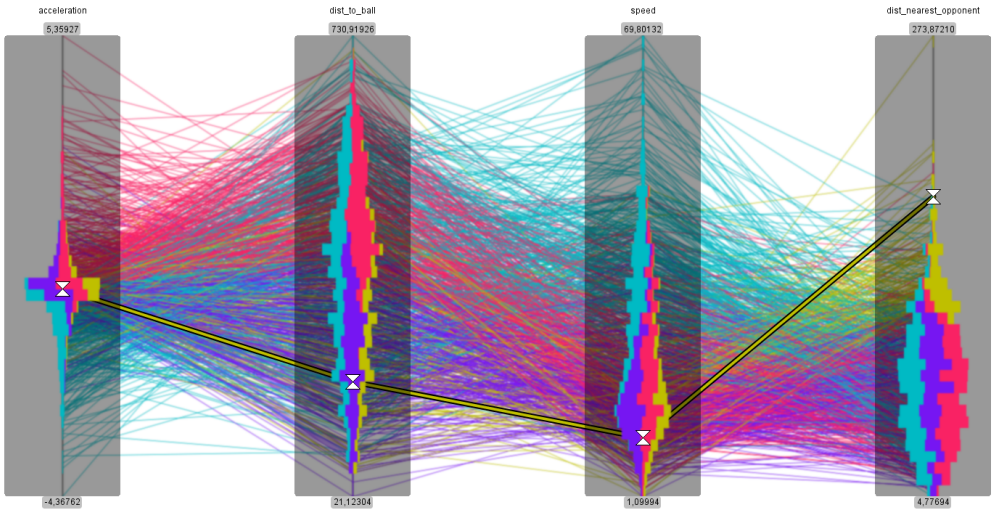


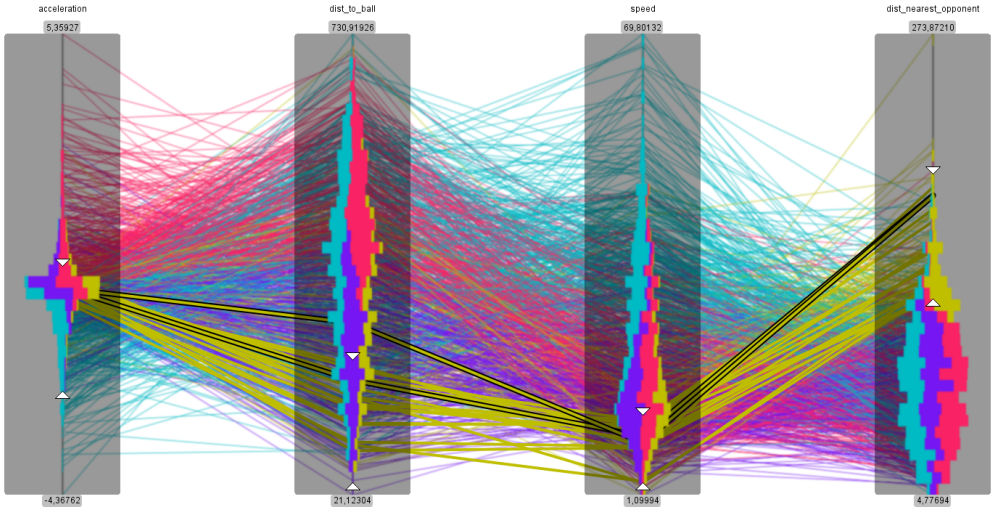
Figure 4.7.1: Clustering and segmentation results of a defender. The movement is colored according to cluster (top) and the temporal changes is depicted by colored bars with the width representing the length of the respective phase (bottom).

movement in a parallel coordinates plot and represent each single phase as one data item (one line in the parallel coordinate plot). We compute average values of each phase and use them in the parallel coordinates plot. The corresponding visualization are depicted in Figure 4.7.2.

We show in Figure 4.7.2 two different filtering steps during the analysis process. In the upper figure, the analyst selected one single phase to investigate the corresponding parallel coordinate line (highlighted by black borders). The filtering intervals will be automatically adjusted to fit the selected phase. As the analyst wants to understand the properties of yellow phases, he moves all range sliders simultaneously starting from the single selected yellow phase (lower



(a) single phase selected



(b) nearby phases filtered

Figure 4.7.2: Parallel coordinate plots for segmentation results with interactive filtering. A phase of interest is selected (top) and interactively the filtering range is increased resulting in similar phases being selected (bottom). The data items emphasized by black borders are highlighted either by phase selection (top) or mouse hovering (bottom).

figure). The analyst hovers over the previously selected line on the axis labeled distance to the nearest opponent, in order to similar phases with the same distance to the nearest opponent. All parallel coordinate lines at the mouse position will be highlighted by black borders and will be rendered unblurred independent of filtering criteria (lower figure). Analyzing the phases visualized in Figure 4.7.2 we were able to derive the following findings:

- Yellow phases correspond to movement with high distances to the nearest opponent, low speed, and low to medium distances to the ball.
- Red phases describe movement with a high distance to the ball. Red phases have a positive acceleration and by trend lower speed compared to turquois phases.
- Turquois phases are independent of the distance to the ball and describe movement with negative acceleration. Negative acceleration values will only occur if the speed is sufficiently high.
- The purple phases being very visual salient in the geospatial representation are described by below-average values of distance to the ball, speed, and distance to the nearest opponent. Furthermore, the acceleration values are around zero.
- The difference between purple and yellow phases is only dependent on the distance to the nearest opponent. This is reflected in the spatial visualization as opponents are mostly near to defenders when opponents attack and defenders should to be near their own goals during opposite attacks.

From these observations, we see that we need several views to the data. For instance, the difference between yellow and purple phases could be only fully understood when combining the spatial and the multi-dimensional feature visualization. We believe that combining several views and connecting them interactively by Brushing & Linking is an effective way to support the analyst.

4.7.3 Shot-Event Feature Pattern Analysis

As described above, we try to gain knowledge from the manually annotated events. We focus in this section on the most important event of a soccer event, namely the shot on goal. We applied and investigated the Decision Trees mentioned above in Section 4.6.4 to classify the events.

We found that the most relevant features are *x-Position* (near to left or right goal), *Total Width of Team* (in dangerous situations the team width in x-dimension is greater that usual), and *Opposite Players around* (more opposite players are around trying to prevent shots). Furthermore, the *speed* feature turned out to be useful for all kind of events. Crosses and shots are events easily detectable by classifiers, whereas fouls and assists are difficult to detect.

The main target of highlighting interesting situations to the analyst is to avoid him watching the whole game over and over again. Our systems proposes situations that might be of interest to the user depending on his selections and helps to skip uninteresting parts of a soccer match.

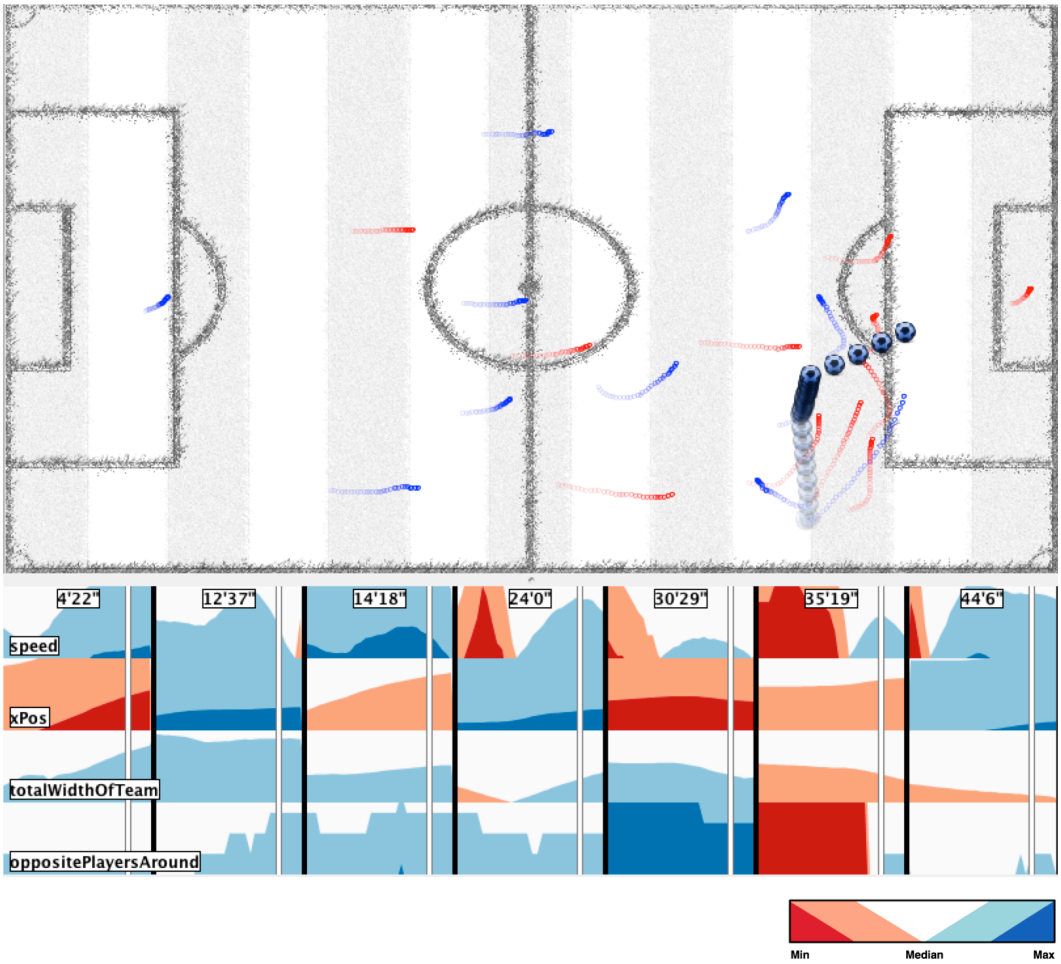


Figure 4.7.3: Horizon Graphs for the relevant features of all shots on goal events in first half of the game. The second shot event is shown on the soccer pitch above. The time point of the event is represented by a vertical white line. Reprinted from [JSS+14].

Annotated Shot Events. We visually inspect all pre-annotated shots on goal by plotting them next to each other using Horizon Graphs for the most relevant features. In Figure 4.7.3 we investigate all relevant features of the first half of the game by Horizon Graphs in combination with the soccer pitch, players, and ball rendered for the second shot event. Similar to crosses (analyzed in Figure 4.5.1), we can detect one direct free kick (6th event) as there are no opposite players around and there is no speed before the shot (the player is waiting until he is allowed to perform the free kick). During all other shot events there are many of opposite players around and the x-position is near to the relevant goal. In most of the events the team width is higher than usual indicating that there is a fast movement of the offensive players towards the goal.

Shot Events Found by Classification. The analyst may also be interested in similar, dangerous, and interesting situations not yet being marked in the data. We therefore exported the transformed soccer data into the KNIME workflow as described in Section 4.6.4. We trained and evaluated our classifiers and imported the results back into our prototype. Several new shot on goal events could be detected by our classifier but were not yet marked in the original game data. Figure 4.7.4 illustrates the classification results. Where green bars depict correctly found events, red represent not found events, and yellow bars stand for potentially interesting events.

The analyst is able to validate new found shot on goal events and mark correct found as new shot on goal events. Following the Visual Analytics pipeline it is possible to add the new events to our KNIME workflow and to update the classifiers. It is therefore feasible to extend, update and improve the classifiers to gain more insights.

For our example, we inspected all found shots on goal events not annotated before and marked the correct ones. We retrained our classifiers with the additional training data and imported the classification results into our tool. By this single iteration we discovered eight new events of which five were relevant. An excerpt of the newly found events can be seen in Figure 4.7.5.

It seems that the extension of our classifiers with additional interesting events helped to move away from pure shot on goal events to overall dangerous events. The upper image in Figure 4.7.5 shows a new not yet marked shot on goal event, whereas the middle and lower image show dangerous situations. In the bottom row for example a striker tried to enter the penalty area, but was stopped in the very last moment. We see the discovery of overall dangerous situations as a prove that the Visual Analytics pipeline helps in improving the classification results.

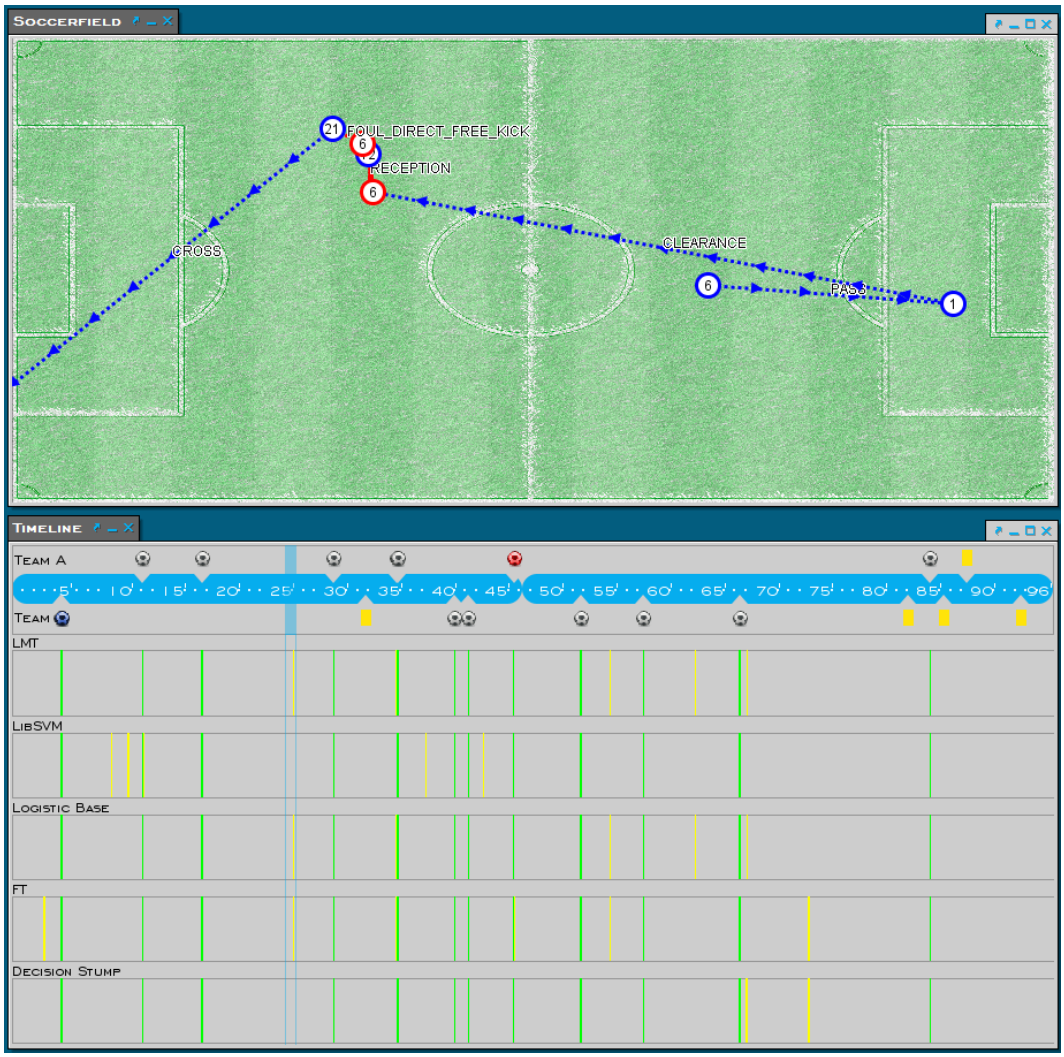


Figure 4.7.4: Analysis of detected new shot on goal events. Green colored bars indicate correct classified events, red represent not found events, and yellow bars show events found by the classifier but not tagged in the original input data. Reprinted from [JSS+14].

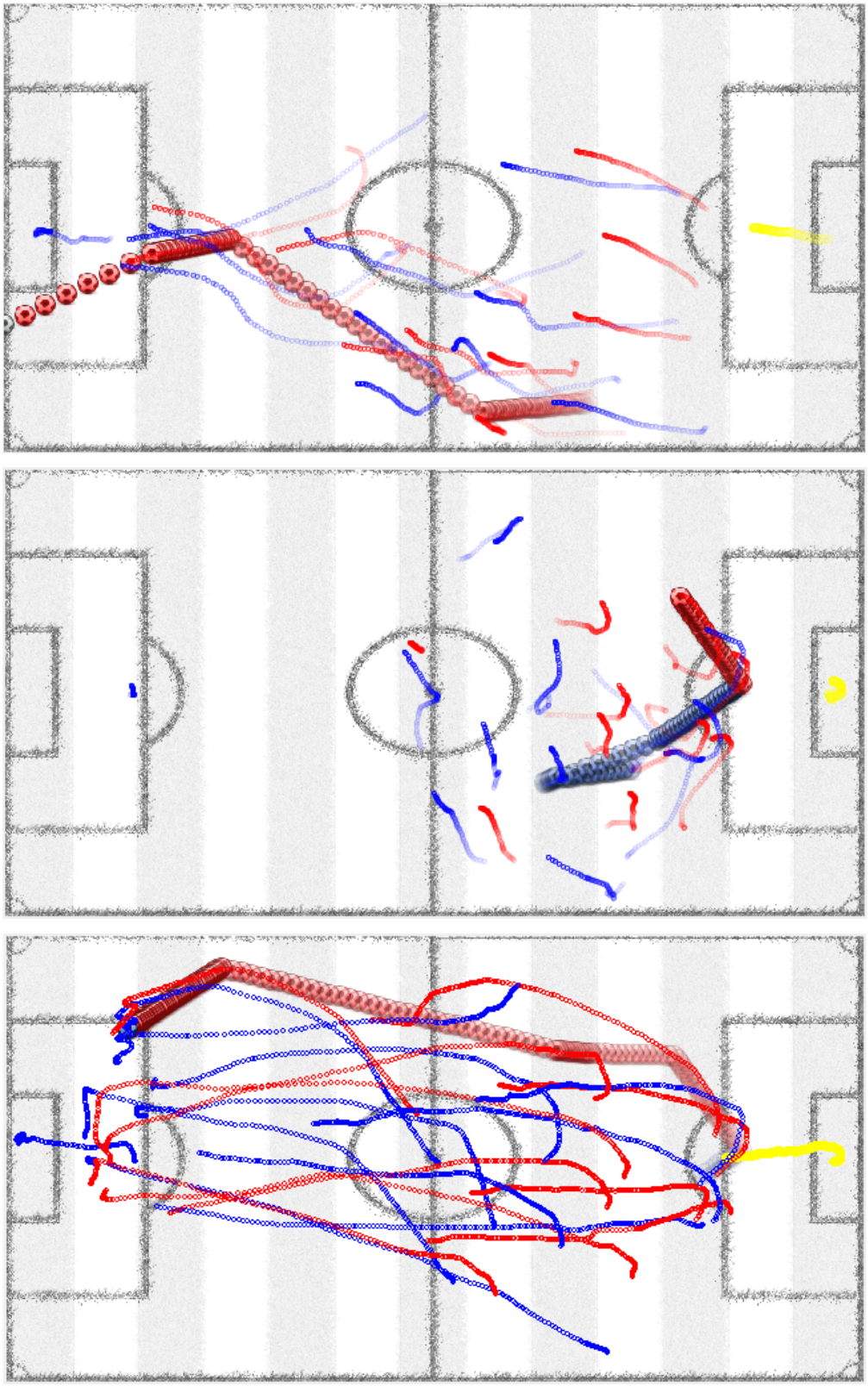


Figure 4.7.5: New events found after adding confirmed events to the classifier's training data. The classifier returns not only shots on goal (top) anymore but also semantically dangerous situations (middle and bottom). Reprinted from [JSS+14].

4.7.4 Back-Four Formation

In this use case, we want to evaluate how the back-four formation performed right before a goal was scored. We investigate a short period before the goal of the first use case is scored (Section 4.7.1).

The key scene of the failure is shown in upper Figure 4.7.6. Our previously described assessment of the back-four formation detects that there seems to be something wrong with the back-four formation resulting in a red coloring. Investigating this time frame we can see why: the back-four formation seems to have problems with their coordination. The nearest midfield player to the right-back is not fulfilling any correct defensive tasks. Unfortunately, the central right defender decides wrong and moves out to the sideline in order to cover another opposite player. Instead, he should have stayed near his usual position to cover the central areas in front of the goal. Although, a free opposite player at the sideline is not good, it is much more dangerous to have large distances between defenders and uncovered opposite players near the middle. A simple pass through the resulting free space leads to a situation with again too much free space for the opposite striker. Three own defending players are consequently outplayed and not involved in the defense anymore.

In lower Figure 4.7.6, the back-four formation has improved their positions and tries hard to recover from their previous mistake. As the central right defender moved back, the overall formation is better than before resulting in greenish coloring. Though the mistake was too severe to recover from and the opposite players is already on his way to score a goal.

The coach of this team can learn from the analysis and teaches his central-back players to stay near the center area and avoid any free spaces in the center. Furthermore, the coach should improve the collaboration and coordination of defensive midfield players and the back-four formation as well. If the midfield player at the sideline had covered his opposite number, the central right defender would not have needed to assist at all.

4.8 Evaluation

During the development of our Visual Analytics prototype, we had some contacts to two domain experts. Expert A is involved into playing soccer since 23 years and into coaching since nine years. Currently, he is working for FC Bayern München being an international successful German soccer club. Expert B plays soccer since 18 years and is referee for matches in the local

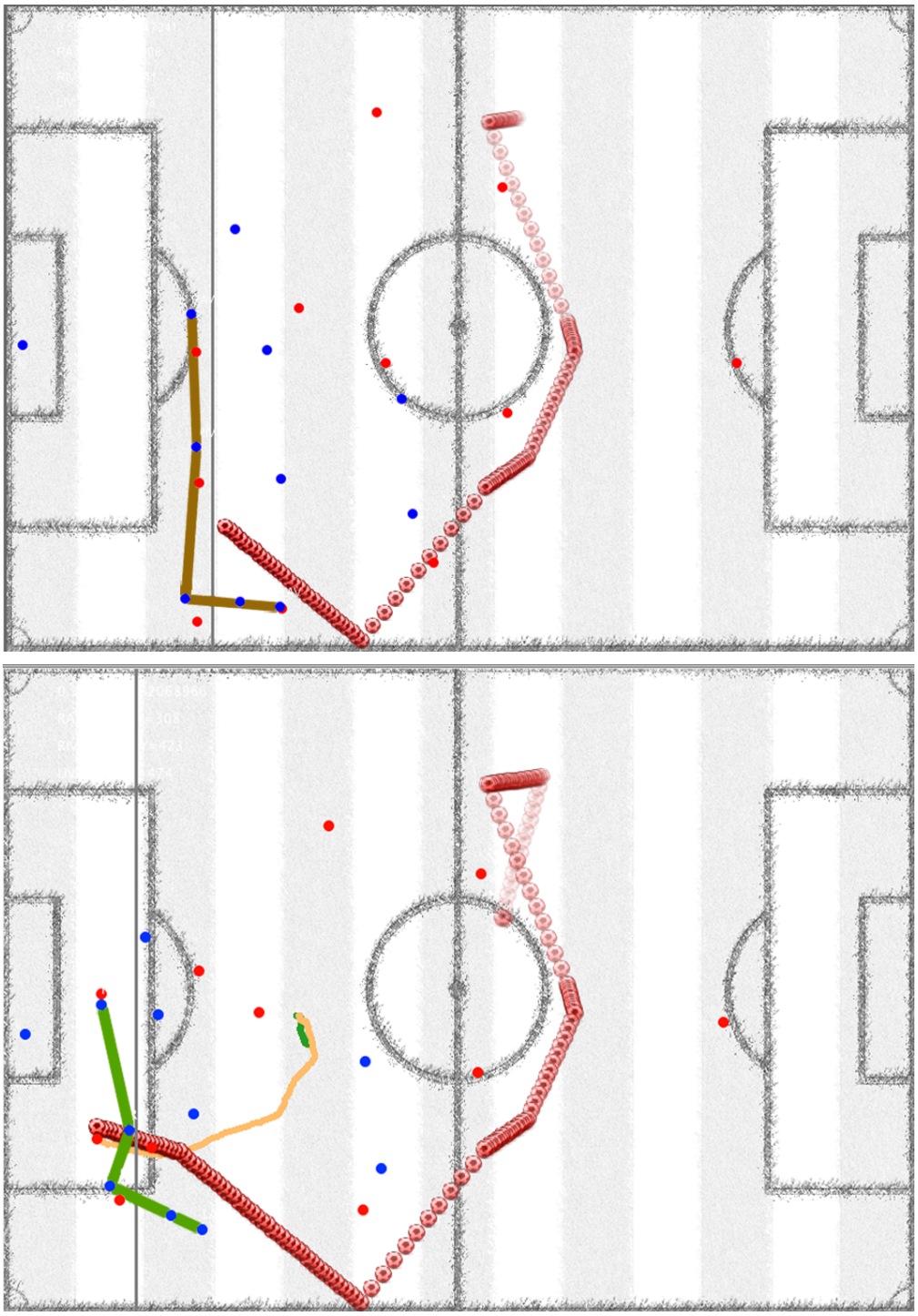


Figure 4.7.6: Back-four formations immediately before the goal occurs. The connecting line is colored from red to green representing the computed quality of the back-four formation. Reprinted from [JSS+14].

area around the lake of Konstanz. We first gathered informal expert feedback in order to evaluate our overall approach and to get some hints for future development. In our second, more formal user study, we decided to focus on the semi-automatic detection of interesting and dangerous events.

4.8.1 First Informal Expert Feedback

We held our first informal feedback session with soccer expert A. He is certain about the benefits a semi-automatic tool has and that such tools can be implemented in professional soccer sports. The semi-automatic analysis will help coaches in cases where there is not enough time for a manual analysis and it allows analyzing more games in the same amount of time compared to a pure manual analysis. Current developments in soccer show that coaches want to decide less by intuition but more by hard facts and figures. We showed the capabilities of our current prototype and the use-cases to the soccer expert and asked for his feedback and opinions. The overall feedback was quite good, but he came immediately with suggestions for improvements that will be included in future versions of this tool.

We were especially interested in the effectiveness of the implemented Horizon Graphs. Horizon Graphs were not intuitive to the soccer expert and were explained to him by showing the visual process of transforming a line chart into a Horizon Graph. After the explanations, he was not only able to read the visualizations but was also convinced that this visualization technique supports him better than traditional line charts. He was amazed by the possibility to see the team's coherence for certain features as speed or acceleration supported by the color changes around the quartiles. In his opinion, Horizon Graphs are most beneficial when comparing the same attribute across several players. Comparing several players reflects the spirit of soccer being a team sport.

Detecting dangerous situations and potential shots semi-automatically was regarded positively, especially with respect to fast analysis tasks. During half-time breaks, the detection of potentially dangerous situations can be very helpful. He mentioned that it would be also interesting to get hints about, why a certain attack did not succeed and lead to a goal.

With respect to future improvements and capabilities of our tool he sees the following potential: Coaches could validate their – maybe intuitive or experience-based – hypotheses in our tool, by looking for a certain kind of situation specified by the coach. Thereafter, the system should automatically derive the corresponding features and detect similar situations and

display them.

4.8.2 Expert Study

In our more formal expert study we invited both, expert A and B, and focused on the semi-automatic detection of dangerous events. We first gave specific tasks to the expert, in order to allow first insights and gain some understanding of our prototype. In a second step, we interviewed the experts and asked them about some of our design decisions. We concluded our study by some general questions about the usefulness of the Visual Analytics approach to soccer and asked for missing features.

Detailed Study Description

We followed in our study a set of tasks and questions depicted in Figure 4.8.1. The tasks were developed, in order to guide the experts and let them explore our prototype. At first, we focused on the detection of dangerous situations using the annotated shot events. The participant should inspect the proposed situations and rate whether these situations are dangerous. We were present during the study and could answer questions and write down interesting quotes and results. We conducted an interview after the tasks and asked for the expert's opinion regarding certain aspects of our prototype.

Study Results

Both experts spent several hours with analyzing the anonymized data sets with our prototype. We were quite astonished that though they didn't know anything about the matches, they were really interested in insights and enjoyed working with our Visual Analytics tool. Furthermore, we got very valuable feedback and suggestions for future improvements.

Concerning the visual classification representation both experts agreed that the history of classification results and the corresponding user interactions are comprehensible and interesting though not presentable to a coach. Coaches are not interested in the course of the analysis but in the results.

The possibility to manually add and remove dangerous situations was meant to be important by both experts. According to expert A, a Visual Analytics system should be flexible to enable coaches and analysts to steer the analysis process in the desired direction. Especially, as every

Expert Study

Analysis Tasks

1. Investigate dangerous situations based on "shot" events. Use the *Single Classifier* mode and *LMT* as classifier. Judge the quality of the found situations and iterate the classification process until you think there are no more good results.
2. Repeat the first task using the *All Classifier* mode.

Interview

1. How useful is the history of old classification results and the corresponding user interaction?
2. How important is it to remove and add new dangerous situations?
3. Do you like to see the classification probability?
4. Do you consider the tooltips helpful at all and/or what information is missing?
5. Do you want to compare the results of different classifiers?

Overall Usefulness

- Why is the application useful and in which situations can it be used?
- What can we improve?
- Do you have any miscellaneous remarks?

Figure 4.8.1: Sheet of tasks and questions guiding through our expert study.

coach has his own opinion about good soccer. Expert B remarked that manually adding situations is beneficial when the coach knows about an important situation not being reflected in the annotated data set before.

Both experts liked to see the classification probability especially with respect to the half time break focusing only on the most important situations. Showing only the classification probability is not enough as the distance to the goal or other influencing factors should be reflected as well.

Basically, tooltips were seen enriching the visualization. Nevertheless, we should integrate more specific information about the kind of situation. For instance, which team is attacking and from which side, who are the involved players, and how did the situation evolve.

Very interesting from our design perspective was to ask the experts whether they wanted to compare different classifiers as shown in Figure 4.7.4. Both experts liked to see whether the majority of classifiers found the same event. Seeing a visual comparison of different classifier results in Small Multiples lead to a very good idea of expert B. He proposes to use the Small Multiple comparison for visualizing the occurrence of different kind of situations. One row could represent all free kicks and another one could represent all counterattacks. Figure 4.8.2 depicts a visual draft for visualizing different kind of situations.

Figure 4.8.2: Visual Draft for Small Multiples visualizing the occurrence of different kinds of events.

Both experts are convinced that our Visual Analytics approach is beneficial and useful for soccer analysts and coaches. They both see the usage scenario during a halftime break and for the preparation and the debriefing of a match. Additionally, our system could support coaches for the individual training of a player and therefore help to win.

During our expert study, we collected a number of improvement wishes and feature requests ranging from user interface to new visualizations and analysis facilities. Some of these will issue

new research work in our prototype. We are for instant currently working on visualizing arbitrary time windows of a match, adjusting the visualization technique to the length and gameplay of the selected time window. As the experts performed several iterations during the detection of interesting situations, we could see how the classification results evolve over time. Summarizing the results for three iterations, we detected that the results of the first iteration nearly all situations detected were dangerous, unannotated shot events. The second iteration extends this set of situations by mostly interesting ones though there are some less important situations. The third iteration results in situations being not critical for the gameplay but can be used for soccer training purposes. Based on these study results and our own experiments, we believe that analysts do not need to perform more than three classification iterations, while two iterations may be sufficient in most cases. Nevertheless, we will further investigate the number of iterations and the corresponding results.

4.9 Conclusion

We presented a Visual Analytics approach to investigate soccer data and gain new insights. Based on the analysis of single-player, multi-player, and event-based we were able to easily detect standard situations as crosses for example. The integration of state-of-the-art data mining techniques helps to find and understand interesting events. Additionally, even not previously annotated interesting events could be found by Visual Analytics methods. Currently, our prototype is set up as an expert tool. We followed a data-driven tool design, namely, we aimed to combine Visual Analytics techniques deemed useful to answer analytical questions in context of high-resolution soccer sensor data. These techniques include interactive and automatic data filtering, visual representation of trajectories on a soccer field, and compact time series visualization using Horizon Graphs. As we expect the types of required movement features to vary between different analytical questions, we decided to compute a large number of features from which the expert can chose. In addition, inspired by similar recent work [BTH+13], we incorporated an interactive classifier which can help to discover events of potential interest, based on example event annotation, relying on a broad basis of features. Given our system is set up as an expert system, we recommend it being used in Pair Analytic scenarios [AHKGF11]. Domain experts were using our Visual Analytics framework and were able to detect dangerous game situations semi-automatically. The domain experts were enjoying performing analyses and exploring soccer matches in our tool.

Our future work includes to provide our prototype to coaches and to support the most often used analyses by predefined configuration settings and the definition of task-driven views. The analysis of soccer features is at an early stage, but this pre-study showed already some information available in the data. We want to extend our approach to a semi-automatic detection of mistakes of a team to help the coach in finding critical situations. Furthermore, we want to integrate video material into our system whenever available and implement more assessment criteria for formations. Furthermore, we want to integrate a better visualization for the movement of players and soccer-specific artifacts as free spaces or running paths. Especially when visualizing longer time windows, a more abstract visualization technique adapted to soccer is necessary. Additionally, we will integrate the experts' feedback in order to support the coach in validating hypotheses and present the findings to his team.

He who would learn to fly one day must first learn to stand and walk and run and climb and dance; one cannot fly into flying.

Friedrich Nietzsche

5 Conclusions and Future Perspectives

As Nietzsche phrased very accurately, it is impossible to achieve a high-level goal consisting of several sub-accomplishments without fulfilling those sub-tasks. Transferred to the domain of Visual Analytics, we want to enable the user to generate and validate hypotheses, derive insights, and gain new knowledge. Designing such Visual Analytics systems asks for many decisions being wisely made. Beginning from the very first data preprocessing decisions to proper visualizations and data mining techniques, all steps require not only skills in Visual Analytics but also domain expertise. We can only achieve our ultimate goal of gaining new knowledge when all techniques applied complement each other perfectly. In this thesis, we discussed and enhanced visualization and analysis techniques in the domain of temporal and geospatial data.

5.1 Summary

We introduced state-of-the-art boosting methods enhancing the visual saliency of data items and discussed their applicability and prerequisites. We believe that guiding the analyst's atten-

tion to data items of interest supports an effective and efficient analysis. The boosting techniques were employed throughout this thesis and for instance highlighted anomalies in time series of power consumption data. A main contribution of this section is a table comparing all boosting techniques. The techniques are not equally effective in the different boosting tasks and the design process of a visual analysis systems should reflect this knowledge.

Our peak-preserving prediction technique was developed to enhance the prediction result, as state-of-the-art methods usually interpret peaks as outliers and disregard them. However, to our domain experts these peaks were crucial as they may hint to severe problems. We decided to design a prediction method being easily inspectable and interpretable based on weighted averaging. Besides a higher accuracy, the main advantage of our prediction technique is that the analyst can control the influence of peakiness to the prediction. Another interesting aspect of this section is the determination of peaks using the inverse of the recursion level of an applied Douglas-Peucker algorithm.

We combined both boosting and prediction in a visual analysis system for power consumption data. We focused in this work on the detection of anomalies and the visual exploration of the time series data. In close collaboration with our subject matter experts, we implemented the major state-of-the-art visualization techniques for time series and allowed to easily switch between the visualization techniques. In order to guide the analyst's attention to exceptional power consumptions, we integrated two anomaly scores with different properties and adjusted the visual layout according to the anomaly scores. For detailed analyses, we implemented similarity-based queries to detect for instance root causes for a selected power consumption pattern.

Scatter plots are widely applied and are a basic technique visualizing two-dimensional data. We enhanced scatter plots by an ellipsoid pixel placement representing local correlations. Our pixel placement algorithm removes the overplotting of data points in dense regions by moving overplotting points to a nearby free position based on the local correlation. We furthermore added lighting to the scatter plot, in order to visualize the original position of a data point. The advantage of an overplotting-free scatter plot is that points can be colored expressively representing a third dimension.

We discussed several methods to simplify line-based spatial visualizations and reduce the amount of overplotting. However, line-based representations are more complicated than point-based visualizations. Lines overlap not only in dense regions but also because of intersections and crossings. We presented approaches to simplify line-based movement representations by reducing the number of segments based on feature distributions. Our technique allows a real-

time adaption of the simplification to the current zoom level and enables both a simplified overview and details on demand by zooming. Complementing the simplification, we proposed an abstraction technique representing rather the concept of the movement patterns.

Visual Analytics for soccer matches combines aspects from the temporal and the spatial domain. We support the analyst in exploring and analyzing soccer matches without replaying the whole recorded game by animation. We implemented several visualization and analysis techniques based on movement related features. We provided a toolbox of methods enabling the analyst investigating the behavior of single or multiple players in highly interactive and interrelated views. Furthermore, we integrated Visual Analytics to guide the analyst's attention to important game situations based on his interests. In collaboration with our domain experts, we could get insights into previously unknown games and could highlight different facets. From an analytics perspective, using false positives resulting from classification for the proposal of important situations is an interesting approach.

5.2 Future Perspectives

When we compare the different methods and techniques being either discussed by related work or presented in this thesis, we can observe some similarities and challenges for future work. During our research enhancing visualization for temporal and spatial data, we discovered several open issues being too large to be covered in this thesis but being essential for a successful integration of the analyst into the Visual Analytics process.

In our work, we enhanced visualization and data mining techniques, in order to support the analyst and guiding his attention to the outlying and interesting aspects in the data. Especially in the application-driven sections for the analysis of power consumption and soccer matches, a tight integration of the analyst into the design process was crucial. During our research, we recognized that materializing domain knowledge in the Visual Analytics process is quite challenging. There are several ways to integrate domain knowledge in form of actions, such as relevance feedback or tuning of parameters. But these actions are results of the domain knowledge and do not provide direct access to the domain expertise. This detour from analysts to the Visual Analytics system via actions can be quite error-prone, as intentions and reasonings of the analyst are not known to the system. Furthermore, from the system's perspective it is not obvious which information helps the analyst best in solving his tasks. For the anomaly detection in power consumption data for example, we thought of integrating additional maintenance

events or weather information to help the analyst explaining detected unusual patterns. These additional information will help, but selecting from a potentially infinite number of additionally available data sources the proper ones is already based on domain knowledge. Consequently, the only way materializing domain knowledge is to tightly collaborate with domain experts and to design the Visual Analytics system accordingly. It is unrealistic to ask the subject matter expert to externalize all his domain knowledge, but his domain knowledge is crucial for the design process. There are already approaches like User-Centered Design in software development, however Visual Analytics tries to generate findings not known to the user beforehand with complex analysis techniques based on domain knowledge. It would be very desirable to have Visual Analytics techniques supporting arbitrary combinations of facts, rules and fuzzy intuitions tightly integrated in the analysis process.

Bridging the gap between animations and still images is research-wise both interesting and challenging. As we discussed in the introduction, both types of visualizations can convey different kinds of information. Depending on the task, the designer of a Visual Analytics system can either choose animations or static images. But there is nothing in between, besides Small Multiples or adjusting the animation speed according to the information load in a scene. A novel technique bringing the best of both worlds together would be a huge contribution to our field. It is obviously not clear, if at all there exists a single method better than the state-of-the-art techniques. The novel technique should be able to increase the situational awareness, visualize correlations, gradual and abrupt changes, and enable the analyst to detect and investigate single, interesting situations.

The simultaneous visualization of geospatial and temporal aspects in data is very challenging and there are not many convincing examples. Some techniques try to encode the temporal dimensions as a third dimension on top of a two-dimensional map resulting in occlusions and perception issues. Others apply techniques like Small Multiples, animation, or glyph representation. Inventing an innovative method depicting temporal changes in a geospatial domain without animation is something worth to pursue. Again, there might be no technique better than the existing ones but any improvement for visualizing temporal and spatial data simultaneously is definitely worth researching.

Acknowledgments

Sorting my thanks in chronological order, I would first like to thank my parents for teaching me scientific curiosity and the fun of knowledge transfer. As a child, you let me assist you during your lectures, wiping the blackboard, and wandering around helping students to understand error messages in Mathematica. During this time, I experienced how inspiring and encouraging a proper learning environment can be by allowing positive emotions during a lecture. I deeply appreciate how you supported and guided me!

I wish to thank Dr. Florian Mansmann for hiring me as a student assistant after my first semester. I was able to get in touch with Information Visualization before I had sufficient implementation skills. It was the time, when I was struck by the power of visualization: Letting the non-visible become visible still fascinates me. Thank you for giving me always the freedom to reach the research goals in a way I found worth pursuing.

Prof. Daniel Keim gave me the great opportunity to work and research in his group already as an undergraduate. I believe that the spirit of inspiring collaboration existing in the Data Analysis and Visualization group is just putting your values into practice. I like to thank you for seeing the researcher behind his research and furthermore for all the encouraging discussions making this thesis possible.

Prof. Oliver Deussen taught me the beauty of Computer Graphics and made me think outside of my Visual Analytics box. You inspired me to combine visualization and Computer Graphics techniques for both more powerful and more aesthetic methods. During our discussions, you guided me to research interesting and challenging problems and to show the connections to other domains.

Visiting Ming Hao at the Hewlett Packard Laboratories in Palo Alto, was a great luck for me. I could enjoy a new culture and solve real application problems having you advising me. Prof. Tobias Schreck was always available to restructure any unstructured contents of my papers and

helped in uncountable fruitful discussions. Violating the chronological order, I like to thank all my companions during my studies, all my collaborators, and co-authors: you are awesome! I love to collaborate with you and we still have so many ideas waiting in our desks for future papers. Special thanks go to Dominik Jäckle, Sebastian Mittelstädt, Dr. Christian Rohrdantz, Dominik Sacha, David Spretke, Manuel Stein, and Florian Stoffel.

Last but not least, I want to express my deepest thanks to my friends and my family for all your additional support. I could always rely on you, when there were any problems and I needed someone to talk to. You taught me more skills and lessons than I could write into one Ph.D. thesis. You made this thesis fly.

Bibliography

[AA06] Natalia V. Andrienko and Gennady L. Andrienko. *Exploratory analysis of spatial and temporal data - a systematic approach*. Springer, 2006.

[AA08] Gennady Andrienko and Natalia Andrienko. Spatio-temporal aggregation for visual analysis of movements. In *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on*, pages 51–58. IEEE, 2008.

[AA10] G. Andrienko and N. Andrienko. A general framework for using aggregation in visual exploration of movement data. *The Cartographic Journal*, 47(1):22–40, 2010.

[AAB+13a] Gennady Andrienko, Natalia Andrienko, Peter Bak, Daniel Keim, and Stefan Wrobel. *Visual Analytics of Movement*. Springer Verlag, 2013.

[AAB+13b] Gennady L. Andrienko, Natalia V. Andrienko, Peter Bak, Daniel A. Keim, and Stefan Wrobel. *Visual Analytics of Movement*. Springer, 2013.

[AAB+13c] Natalia V. Andrienko, Gennady L. Andrienko, Louise Barrett, Marcus Dostie, and S. Peter Henzi. Space transformation for understanding group movement. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2169–2178, 2013.

[AABW12] Gennady Andrienko, Natalia Andrienko, Michael Burch, and Daniel Weiskopf. Visual analytics methodology for eye movement studies. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2889–2898, 2012.

[ABKS99] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, pages 49–60, New York, NY, USA, 1999. ACM.

[ADO15] Adobe kuler, 2015. Adobe Systems Incorporated.

[AHKGF11] Richard Arias-Hernández, Linda T. Kaastra, Tera Marie Green, and Brian D. Fisher. Pair analytics: Capturing reasoning processes in collaborative visual analytics. In *HICSS*, pages 1–10, 2011.

[AHPMW05] Pankaj K. Agarwal, Sariel Har-Peled, Nabil H. Mustafa, and Yusu Wang. Near-linear time approximation algorithms for curve simplification. *Algorithmica*, 42:203–219, 2005.

[Ake84] Jerry Van Aken. An efficient ellipse-drawing algorithm. *IEEE Computer Graphics and Applications*, 4(9):24–35, 1984.

[AMST11] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer London, 2011.

[AS07] Aleks Aris and Ben Shneiderman. Designing semantic substrates for visual network exploration. *Information Visualization*, 6(4):281–300, 2007.

[AVH+06] A. Anagnostopoulos, M. Vlachos, M. Hadjieleftheriou, E. Keogh, and P.S. Yu. Global distance-based segmentation of trajectories. In *Proceedings of 12th ACM SIGKDD*, pages 34–43, 2006.

[AWG09] Yuvraj Agarwal, Thomas Weng, and Rajesh K Gupta. The energy dashboard: improving the visibility of energy consumption at a campus-wide scale. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 55–60. ACM, 2009.

[BA03] Adrian W Bowman and Adelchi Azzalini. Computational aspects of nonparametric smoothing with illustrations from the sm library. *Computational statistics & data analysis*, 42(4):545–560, 2003.

[BA04] Adrian W Bowman and Adelchi Azzalini. *Applied smoothing techniques for data analysis*. Clarendon Press, 2004.

[BAP+05] Paolo Buono, Aleks Aris, Catherine Plaisant, Amir Khella, and Ben Shneiderman. Interactive pattern search in time series. In *Electronic Imaging 2005*, pages 175–186. International Society for Optics and Photonics, 2005.

[BBBL11] Ilya Boyandin, Enrico Bertini, Peter Bak, and Denis Lalanne. Flowstrates: An approach for visual exploration of temporal origin-destination data. *Computer Graphics Forum*, 30(3):971–980, 2011.

[BCC+13] R. Basole, E. Clarkson, A. Cox, C. Healey, J. Stasko, and C. Stolper (Organizers). First IEEE visworkshop on sports data visualization, Oct. 14, 2013.

[BCD+07] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. KNIME: The Konstanz Information Miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer, 2007.

[BDvKS10] Maike Buchin, Anne Driemel, Marc J. van Kreveld, and Vera Sacristan. An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In *GIS*, pages 202–211, 2010.

[BE95] Marla J Baker and Stephen G Eick. Space-filling software visualization. *Journal of Visual Languages and Computing*, 6(2):119–133, 1995.

[BGR06] Thorsten Buering, Jens Gerken, and Harald Reiterer. User interaction with scatterplots on small screens-a comparative evaluation of geometric-semantic zoom and fisheye distortion. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):829–836, 2006.

[BHvW00] Mark Bruls, Kees Huizing, and Jarke van Wijk. Squarified treemaps. In *Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization (VisSym 00)*, pages 33–42. Eurographics Association, 2000.

[BLG99] Magnus Broberg, Lars Lundberg, and Håkan Grahn. Visualization and performance prediction of multithreaded solaris programs by tracing kernel threads. In *13th International Parallel Processing Symposium & 10th Symposium on Parallel and Distributed Processing (IPPS/SPDP 1999)*, pages 407–413, 1999.

[Bli78] James F Blinn. Simulation of wrinkled surfaces. *ACM SIGGRAPH Computer Graphics*, 12(3):286–292, 1978.

[BMA+11] Gowtham Bellala, Manish Marwah, Martin Arlitt, Geoff Lyon, and Cullen E Bash. Towards an understanding of campus-scale power consumption. In *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 73–78. ACM, 2011.

[BMA+12] Gowtham Bellala, Manish Marwah, Martin Arlitt, Geoff Lyon, and Cullen Bash. Following the electrons: methods for power management in commercial buildings. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 994–1002. ACM, 2012.

[BPS06] Cullen E Bash, Chandrakant D Patel, and Ratnesh K Sharma. Dynamic thermal management of air cooled data centers. In *Thermal and Thermomechanical Phenomena in Electronics Systems, 2006. ITHERM'06. The Tenth Intersociety Conference on*, pages 445–452. IEEE, 2006.

[BRR11] Maxim Buevich, Anthony Rowe, and Raj Rajkumar. SAGA: Tracking and Visualization of Building Energy. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2011 IEEE 17th International Conference on*, volume 2, pages 31–36. IEEE, 2011.

[BTH+13] Harald Bosch, Dennis Thom, Florian Heimerl, Edwin Puttmann, Steffen Koch, Robert Krüger, Michael Wörner, and Thomas Ertl. Scatterblogs2: Real-time monitoring of microblog messages through user-guided filtering. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2022–2031, 2013.

[Bur05] D. Burghardt. Controlled line smoothing by snakes. *GeoInformatica*, 9(3):237–252, 2005.

[BW08] Sven Bachthaler and Daniel Weiskopf. Continuous scatterplots. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1428–1435, 2008.

[BW11] Michael Burch and Daniel Weiskopf. Visualizing dynamic quantitative data in hierarchies. In *Proceedings of International Conference on Information Visualization Theory and Applications*, pages 177–186, 2011.

[CCM10] Yu-Hsuan Chan, C Correa, and Kwan-Liu Ma. Flow-based scatterplots for sensitivity analysis. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 43–50. IEEE, 2010.

[CDH12] Stacey Chapman, Edward Derse, and Jacqueline Hansen, editors. *LA84 Foundation Soccer Coaching Manual*. LA84 Foundation, 2012.

[Cha03] Chris Chatfield. *The analysis of time series: an introduction*. CRC press, 2003.

[CIE78] *Recommendations on Uniform Color Spaces, Color-difference Equations, Psychometric Color Terms*. CIE publication. International Commission on Illumination, 1978.

[CLNL87] Daniel B Carr, Richard J Littlefield, WL Nicholson, and JS Littlefield. Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82(398):424–436, 1987.

[CM84] William S Cleveland and Robert McGill. The many faces of a scatterplot. *Journal of the American Statistical Association*, 79(388):807–822, 1984.

[CMM10] Victoria M Catterson, Stephen DJ McArthur, and Graham Moss. Online conditional anomaly detection in multivariate data for transformer monitoring. *Power Delivery, IEEE Transactions on*, 25(4):2556–2564, 2010.

[Cro07] Samuel T Croker. Effective forecast visualization with sas/graph. In *SAS Global Forum*, 2007.

[CSC+05] Malu Castellanos, Norman Salazar, Fabio Casati, Umesh Dayal, and Ming-Chien Shan. Predictive business operations management. In Subhash Bhalla, editor, *Databases in Networked Information Systems*, volume 3433 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2005.

[DAF+13] Ricardo Duarte, Duarte Araújo, Hugo Folgado, Pedro Esteves, Pedro Marques, and Keith Davids. Capturing complex, non-linear team behaviours during competitive football performance. *Journal of Systems Science and Complexity*, 26(1):62–72, 2013.

[DM03] J. Dykes and D. Mountain. Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications. *Computational Statistics & Data Analysis*, 43(4):581–603, 2003.

[DP73] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

[DSBT+07] V. Di Salvo, R. Baron, H. Tschan, F. J. Calderon Montero, N. Bachl, and F. Pigozzi. Performance characteristics according to playing position in elite soccer. *International journal of sports medicine*, 28(3):222, 2007.

[DWA10] Jordi Duch, Joshua S. Waitzman, and Luís A. Nunes Amaral. Quantifying the performance of individual players in a team activity. *PloS one*, 5(6):e10937, 2010.

[DWF09] S. Dodge, R. Weibel, and E. Forootan. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems*, 33(6):419–434, 2009.

[ED02] Geoffrey Ellis and Alan Dix. Density control through random sampling: an architectural perspective. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 82–90. IEEE, 2002.

[EKSX96] Martin Ester, H.P. Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, volume 1996, pages 226–231. Portland: AAAI Press, 1996.

[FH00] P. Forer and O. Huisman. Space, Time and Sequencing: Substitution at the Physical/Virtual Interface. *Information, Place, and Cyberspace: Issues in Accessibility*, page 73, 2000.

[FIF15] Fédération Internationale de Football Association. Big Count, March 2015. http://www.fifa.com/worldfootball/bigcount/index.html, accessed April 2015.

[FLF+11] Nivan Ferreira, Lauro Lins, Daniel Fink, Steve Kelling, Christopher Wood, Juliana Freire, and Claudio Silva. Birdvis: Visualizing and understanding bird populations. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2374–2383, 2011.

[FMT+13] Sofia Fonseca, João Milho, Bruno Travassos, Duarte Araújo, and António Lopes. Measuring spatial interaction behavior in team sports using superimposed voronoi diagrams. *International Journal of Performance Analysis in Sport*, 13(1):179–189, 2013.

[FN05] Glenn A. Fink and Chris North. Root Polar Layout of Internet Address Data for Security Administration. In *Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSEC)*, pages 55–64. IEEE Computer Society, 2005.

[FS05] Akira Fujimura and Kokichi Sugihara. Geometric analysis and quantitative evaluation of sport teamwork. *Systems and Computers in Japan*, 36(6):49–58, 2005.

[GGM10] J.L. Guerrero, J. Garcia, and J.M. Molina. Air traffic trajectories segmentation based on time-series sensor data. *Sensor Fusion and its Applications*, 2010.

[GJL+09] Edward Grundy, Mark W. Jones, Robert S. Laramee, Rory P. Wilson, and Emily L.C. Shepard. Visualisation of sensor data from animal movement. *Computer Graphics Forum*, 28(3):815–822, 2009.

[Gol12] Kirk Goldsberry. Courtvision: New visual and spatial analytics for the nba. In *MIT Sloan Sports Analytics Conference*, 2012.

[Goo13] Google. Google PowerMeter, viewed 6/17/13. http://www.google.com/powermeter/about/.

[GP08] F. Giannotti and D. Pedreschi. *Mobility, data mining, and privacy: geographic knowledge discovery*. Springer, 2008.

[GPGP09] Jessica Granderson, Mary Ann Piette, Girish Ghatikar, and Phillip Price. Building energy information systems: State of the technology and user case studies. *Handbook of Web Based Energy Information and Control Systems*, 2009.

[GS05] R.H. Güting and M. Schneider. *Moving objects databases*. Morgan Kaufmann Publisher, 2005.

[GTC+11] Alessandro Giusti, Pierluigi Taddei, Giorgio Corani, Luca Gambardella, Cristina Magli, and Luca Gianaroli. Artificial defocus for displaying markers in microscopy z-stacks. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):1757–1764, 2011.

[Guo09] Diansheng Guo. Flow Mapping and Multivariate Visualization of Large Spatial Interaction Data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1041–1048, October 2009.

[GW13] Joachim Gudmundsson and Thomas Wolle. Football analysis using spatio-temporal tools. In *Computers, Environment and Urban Systems*. Elsevier, 2013.

[Ham94] J.D. Hamilton. *Time series analysis*, volume 2. Cambridge Univ Press, 1994.

[HBK+07] J.A. Helmuth, C.J. Burckhardt, P. Koumoutsakos, U.F. Greber, and I.F. Sbalzarini. A novel supervised trajectory segmentation algorithm identifies distinct types of human adenovirus motion in host cells. *Journal of structural biology*, 159(3):347–358, 2007.

[HDKS05] Ming C Hao, Umeshwar Dayal, Daniel A Keim, and Tobias Schreck. Importance-driven visualization layouts for large time series data. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 203–210. IEEE, 2005.

[Hea96] Christopher G Healey. Choosing effective colours for data visualization. In *Visualization'96. Proceedings.*, pages 263–270. IEEE, 1996.

[Hex15] Bivariate Binning into Hexagon Cells, 2015. http://rpackages.ianhowson.com/cran/hexbin/man/hexbin.html, accessed April 2015.

[HFH+09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[HJM+11] Ming C. Hao, Halldor Janetzko, Sebastian Mittelstädt, Walter Hill, Umeshwar Dayal, Daniel A. Keim, Manish Marwah, and Ratnesh K. Sharma. A Visual Analytics Approach for Peak-Preserving Prediction of Large Seasonal Time Series. *Computer Graphics Forum*, 30(3):691–700, 2011.

[HJS+09] Ming C. Hao, Halldor Janetzko, Ratnesh K. Sharma, Umeshwar Dayal, Daniel A. Keim, and Malu Castellanos. Poster: Visual prediction of time series. *IEEE VAST*, pages 229–230, 2009.

[HK06] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[HKA09] Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1303–1312. ACM, 2009.

[HLD02] Helwig Hauser, Florian Ledermann, and Helmut Doleisch. Angular brushing of extended parallel coordinates. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 127–130. IEEE, 2002.

[Holo4] Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10, 2004.

[HTC09] C. Hurter, B. Tissoires, and S. Conversy. FromDaDy: Spreading aircraft trajectories across views to support iterative queries. *IEEE T Vis Comput Gr*, 15(6):1017–1024, 2009.

[HvW09] Danny Holten and Jarke J. van Wijk. A user study on visualizing directed edges in graphs. In *SIGCHI Conf on Human Factors in Computing Systems*, pages 2299 – 2308, 2009.

[IBM13] IBM. IBM TRIRIGA Energy Optimization: Integrated software solution for improving buildings management and facilities operation, viewed 6/17/13. http://pic.dhe.ibm.com/infocenter/tivihelp/v57r1/topic/com.ibm.iteo.doc/infocenter.pdf.

[IBM13] IBM Research and the IBM Cognos software group. Many eyes: Public building energy consumptions, 2013.

[ID91] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates. In *Human-Machine Interactive Systems*, pages 199–233. Springer, 1991.

[ITFY02] Yoshihiko Ichikawa, Tomoko Tsunawaki, Issei Fujishiro, and Hiwon Yoon. A visualization environment for multiple daytime stock price predictions. *IEIC Technical Report (Institute of Electronics, Information and Communication Engineers)*, 102(208):181–186, 2002.

[IWSK07] Yuri Ivanov, Christopher Wren, Alexander Sorokin, and Ishwinder Kaur. Visualizing the history of living spaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1153–1160, 2007.

[JC10] Vikramaditya Jakkula and Diane Cook. Outlier detection in smart environment structured power datasets. In *Intelligent Environments (IE), 2010 Sixth International Conference on*, pages 29–33. IEEE, 2010.

[JHM+13] Halldor Janetzko, Ming C. Hao, Sebastian Mittelstädt, Umeshwar Dayal, and Daniel A. Keim. Enhancing Scatter Plots Using Ellipsoid Pixel Placement and Shading. In Jr. Ralph H. Sprague, editor, *Proceedings of the 46th Annual Hawaii International Conference on System Sciences*, pages 1522–1531. IEEE Computer Society, January 2013.

[JJDK14] Halldor Janetzko, Dominik Jäckle, Oliver Deussen, and Daniel A. Keim. Visual Abstraction of Complex Motion Patterns. In *SPIE 2014 Conference on Visualization and Data Analysis (VDA 2014), Best Paper Award*, volume 9017, pages 90170J–0–90170J–12. IS&T/SPIE, 2014.

[JS91] Brian Johnson and Ben Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on*, pages 284–291. IEEE, 1991.

[JS98] Dean F Jerding and John T Stasko. The information mural: A technique for displaying and navigating large information spaces. *Visualization and Computer Graphics, IEEE Transactions on*, 4(3):257–271, 1998.

[JSMK14] Halldór Janetzko, Florian Stoffel, Sebastian Mittelstädt, and Daniel A. Keim. Anomaly detection for visual analytics of power consumption data. *Computers & Graphics*, 38(0):27 – 37, 2014.

[JSS+14] Halldor Janetzko, Dominik Sacha, Manuel Stein, Tobias Schreck, Daniel A. Keim, and Oliver Deussen. Feature-Driven Visual Analytics of Soccer Data. *Proceedings of the 2014 IEEE Symposium on Visual Analytics Science and Technology (VAST '14)*, 20(12), December 2014.

[KAK95] Daniel A Keim, Mihael Ankerst, and Hans-Peter Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. In *Proceedings of the 6th conference on Visualization'95*, pages 279–286. IEEE Computer Society, 1995.

[Kal60] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[Kei00] Daniel A Keim. Designing pixel-oriented visualization techniques: Theory and applications. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):59–78, 2000.

[KHD+09] Daniel A. Keim, Ming C. Hao, Umeshwar Dayal, Halldor Janetzko, and Peter Bak. Generalized Scatter Plots. *Information Visualization Journal (IVS)*, 2009. 2009/12/24/online.

[KHL06] Chan-Hyun Kang, Jung-Rae Hwang, and Ki-Joune Li. Trajectory analysis for soccer players. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 377–381. IEEE, 2006.

[Kim04] Sangrak Kim. Voronoi analysis of a soccer game. *Nonlinear Analysis: Modelling and Control*, 9(3):233–240, 2004.

[KK95] Daniel A Keim and Hans-Peter Kriegel. Issues in visualizing large databases. In *Proc. Conf. on Visual Database Systems (VDB'95), Lausanne, Schweiz*, pages 203–214, 1995.

[KKEE11] KyungTae Kim, Sungahn Ko, Niklas Elmqvist, and David S Ebert. Wordbridge: Using composite tag clouds in node-link diagrams for visualizing content and relations in text corpora. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–8. IEEE, 2011.

[KKL11] Ho-Chul Kim, Oje Kwon, and Ki-Joune Li. Spatial and spatiotemporal analysis of soccer. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 385–388. ACM, 2011.

[KMH01] Robert Kosara, Silvia Miksch, and Helwig Hauser. Semantic depth of field. In *Proceedings of: IEEE Symposium on Information Visualization 2001 (INFOVIS 2001)*, pages 97–104. IEEE Computer Society Press, 2001.

[KMH+02] Robert Kosara, Silvia Miksch, Helwig Hauser, Johann Schrammel, Verena Giller, and Manfred Tscheligi. Useful properties of Semantic Depth of Field for Better F+C Visualization. In *Proceedings of the Symposium on Data Visualisation 2002*, VISSYM '02, pages 205–210. Eurographics Association, 2002.

[KNPS02] Daniel A Keim, Stephen C North, Christian Panse, and Jörn Schneidewind. Efficient cartogram generation: A comparison. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 33–36. IEEE, 2002.

[KO07] Daniel A. Keim and Daniela Oelke. Literature Fingerprinting: A New Method for Visual Literary Analysis. In *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology (VAST '07)*, pages 115–122. IEEE Computer Society, 2007.

[KPS+03] Daniel A Keim, Christian Panse, Matthias Schafer, Mike Sips, and Stephen C North. Histoscale: An efficient approach for computing pseudo-cartograms. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 93. IEEE Computer Society, 2003.

[KR96] T.A. Keahey and E.L. Robertson. Techniques for non-linear magnification transformations. In *Proceedings of the IEEE Symposium on Information Visualization, IEEE Visualization*, volume 10, pages 38–45, 1996.

[Kra03] M. J. Kraak. The Space-Time Cube Revisited from a Geovisualization Perspective. In *Int'l Cartographic Conf*, volume 1988–1995, Durban, South-Africa, 2003.

[KSS07] Daniel A. Keim, Jörn Schneidewind, and Mike Sips. *Scalable Pixel Based Visual Data Exploration*, pages 12–14. Springer, 2007.

[Kwa00] M.-P. Kwan. Interactive geovisualization of activity-travel patterns using three-dimensional geographical information systems: a methodological exploration with a large data set. *Transport Res C-Emer*, 8(1-6):185 – 203, 2000.

[LAB+09] Tim Lammarsch, Wolfgang Aigner, Alessio Bertone, Johannes Gartner, Eva Mayr, Silvia Miksch, and Michael Smuc. Hierarchical temporal patterns and interactive aggregated views for pixel-based visualizations. In *Information Visualisation, 2009 13th International Conference*, pages 44–50. IEEE, 2009.

[LÇK10] Xia Li, Arzu Çöltekin, and Menno-Jan Kraak. Visual exploration of eye movement data using the space-time-cube. In *Geographic Information Science*, pages 295–309. Springer, 2010.

[LCP+12] Philip A. Legg, David H. S. Chung, Matthew L. Parry, Mark W. Jones, Rhys Long, Iwan W. Griffiths, and Min Chen. Matchpad: Interactive glyph-based visualization for real-time sports performance analysis. *Computer Graphics Forum*, 31(3pt4):1255–1264, 2012.

[LCP+13] Philip A. Legg, David H. S. Chung, Matthew L. Parry, Rhodri Bown, Mark W. Jones, Iwan W. Griffiths, and Min Chen. Transformation of an uncertain video search pipeline to a sketch-based visual analytics loop. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2109–2118, 2013.

[LGP+07] P.P. Lévy, B. Grand, F. Poulet, M. Soto, L. Darago, L. Toubiana, and J.F. Vibert. *Pixelization Paradigm: First Visual Information Expert Workshop, VIEW 2006, Paris, France, April 24-25, 2006: Revised Selected Papers*. Springer, 2007.

[LHL08] J.G. Lee, J. Han, and X. Li. Trajectory Outlier Detection: A Partition-and-Detect Framework. In *IEEE Int'l Conf on Data Engineering*, pages 140–149, 2008.

[LHW07] J.G. Lee, J. Han, and K.Y. Whang. Trajectory clustering: a partition-and-group framework. In *Int'l Conference on Management of Data*, pages 593–604. ACM, 2007.

[LIS12] Guangwen Liu, Masayuki Iwai, and Kaoru Sezaki. A method for online trajectory simplification by enclosed area metric. In *Proc. of the Sixth International Conference on Mobile Computing and Ubiquitous Networking*, pages 40–47, 2012.

[LIW05] P. Laube, S. Imfeld, and R. Weibel. Discovering relative motion patterns in groups of moving point objects. *International Journal Geographic Information Science*, 19(6):639–668, 2005.

[LKL+04] Jessica Lin, Eamonn Keogh, Stefano Lonardi, Jeffrey P Lankford, and Donna M Nystrom. Visually mining and monitoring massive time series. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 460–469. ACM, 2004.

[LLLH10] Z. Li, J.G. Lee, X. Li, and J. Han. Incremental clustering for trajectories. In *Database Systems for Advanced Applications*, pages 32–46, 2010.

[LP10] P. Laube and R. Purves. Cross-scale movement trajectory analysis. In *GIS Research UK 18th Annual Conference GISRUK*, pages 103–107, 2010.

[LPLBDG10] Carlos Lago-Peñas, Joaquín Lago-Ballesteros, Alexandre Dellal, and Maite Gómez. Game-related statistics that discriminated winning, drawing and losing teams from the spanish soccer league. *Journal of sports science & medicine*, 9(2):288, 2010.

[MBMM05] Stephen DJ McArthur, Campbell D Booth, JR McDonald, and Ian T McFadyen. An agent-based anomaly detection architecture for condition monitoring. *Power Systems, IEEE Transactions on*, 20(4):1675–1682, 2005.

[MG13] Adrian Mayorga and Michael Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1526–1538, September 2013.

[Mou05] D.M. Mountain. Visualizing, Querying and Summarizing Individual Spatio-Temporal Behaviour. *Exploring Geovisualization*, pages 181–200, 2005.

[MPKP11] Johanna L Mathieu, Phillip N Price, Sila Kiliccote, and Mary Ann Piette. Quantifying changes in building electricity use, with application to demand response. *Smart Grid, IEEE Transactions on*, 2(3):507–518, 2011.

[NMMN10] Ryota Nakanishi, Junya Maeno, Kazuhito Murakami, and Tadashi Naruse. An approximate computation of the dominant region diagram for the real-time analysis of group behaviors. In *RoboCup 2009: Robot Soccer World Cup XIII*, pages 228–239. Springer, 2010.

[NS10] Dinh-Quyen Nguyen and Heidrun Schumann. Taggram: Exploring geo-data on maps through a tag cloud-based visualization. In *Information Visualisation (IV), 2010 14th International Conference*, pages 322–328. IEEE, 2010.

[OAA+12] Kristien Ooms, Gennady L. Andrienko, Natalia V. Andrienko, Philippe De Maeyer, and Veerle Fack. Analysing the spatial dimension of eye movement data using a visual analytic approach. *Expert Syst. Appl.*, 39(1):1324–1332, 2012.

[OJS+11] Daniela Oelke, Halldor Janetzko, Svenja Simon, Klaus Neuhaus, and Daniel A. Keim. Visual boosting in pixel-based visualizations. *Computer Graphics Forum*, 30(3):871–880, 2011.

[Par62] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, pages 1065–1076, 1962.

[PCS95] Catherine Plaisant, David Carr, and Ben Shneiderman. Image-Browser Taxonomy and Guidelines for Designers. *IEEE Softw.*, 12(2):21–32, March 1995.

[Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[PMSR09] Debprakash Patnaik, Manish Marwah, Ratnesh Sharma, and Naren Ramakrishnan. Sustainable operation and management of data center chillers using temporal data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1305–1314. ACM, 2009.

[PPK+11] C. Panagiotakis, N. Pelekis, I. Kopanakis, E. Ramasso, and Y. Theodoridis. Segmentation and Sampling of Moving Object Trajectories based on Representativeness. *IEEE Transactions on Knowledge and Data Engineering*, 2011.

[PSBS12] Hannah Pileggi, Charles D. Stolper, J. Michael Boyle, and John T. Stasko. Snapshot: Visualization to propel ice hockey analytics. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2819–2828, 2012.

[PSKN06] Christian Panse, Mike Sips, Daniel A. Keim, and Stephen C. North. Visualization of Geo-spatial Point Sets via Global Shape Transformation and Local Pixel Placement. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 12(5):749–756, 2006.

[PT12] Javier López Peña and Hugo Touchette. A network theory analysis of football strategies. *arXiv preprint arXiv:1206.6904*, 2012.

[PVF+13] Charles Perin, Romain Vuillemot, Jean-Daniel Fekete, et al. Soccerstories: A kick-off for visual soccer analysis. *IEEE transactions on visualization and computer graphics*, 2013.

[PXY+05] Doantam Phan, Ling Xiao, Ron Yeh, Pat Hanrahan, and Terry Winograd. Flow Map Layout. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 219–224, 2005.

[RFF+08] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of animation in trend visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1325–1332, 2008.

[RG10] Pedro Rodrigues and João Gama. A Simple Dense Pixel Visualization for Mobile Sensor Data Mining. In Mohamed Gaber, Ranga Vatsavai, Olufemi Omitaomu, João Gama, Nitesh Chawla, and Auroop Ganguly, editors, *Knowledge Discovery from Sensor Data*, volume 5840 of *Lecture Notes in Computer Science*, pages 175–189. Springer Berlin / Heidelberg, 2010.

[Roc71] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall Series in Automatic Computation, chapter 14, pages 313–323. Prentice-Hall, Englewood Cliffs NJ, 1971.

[RSB+10] Adrian Rusu, Doru Stoica, Edward Burns, Benjamin Hample, Kevin McGarry, and Robert Russell. Dynamic visualizations for soccer statistical analysis. In *Information Visualisation (IV), 2010 14th International Conference*, pages 207–212. IEEE, 2010.

[RSB11] Adrian Rusu, Doru Stoica, and Edward Burns. Analyzing soccer goalkeeper performance using a metaphor-based visualization. In *Information Visualisation (IV), 2011 15th International Conference on*, pages 194–199. IEEE, 2011.

[SAS13] SAS Institute Inc. Time series forecasting system offered by SAS software. Website, 2013. Available online at http://www.sas.com/technologies/analytics/forecasting/; visited on November 13, 2013.

[SBTK09] Tobias Schreck, Jürgen Bernard, Tatiana Tekušová, and Jörn Kohlhammer. Visual cluster analysis of trajectory data with interactive Kohonen maps. *Palgrave Macmillan Information Visualization*, 8:14–29, 2009.

[See07] John E Seem. Using intelligent data analysis to detect abnormal energy consumption in buildings. *Energy and buildings*, 39(1):52–58, 2007.

[Shn92] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on graphics (TOG)*, 11(1):92–99, 1992.

[Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[SJM+11] David Spretke, Halldor Janetzko, Florian Mansmann, Peter Bak, Bart Kranstauber, Sarah Davidson, and Manuel Mueller. Exploration through Enrichment: A Visual Analytics Approach for Animal Movement. In Divyakant Agrawal, Isabel Cruz, Christian S. Jensen, Eyal Ofek, and Egemen Tanin, editors, *Proceedings of the 19th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 421–424, New York, NY, USA, 2011. ACM.

[SKC03] Nayera Sadek, Alireza Khotanzad, and Thomas Chen. Atm dynamic bandwidth allocation using f-arima prediction model. In *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on*, pages 359–363. IEEE, 2003.

[SKM06] Tobias Schreck, Daniel A. Keim, and Florian Mansmann. Regular TreeMap Layouts for Visual Analysis of Hierarchical Data. In *Proceedings of the Spring Conference on Computer Graphics (SCCG'2006)*, pages 184–191, Casta Papiernicka, Slovak Republic, April 2006. ACM Siggraph.

[SLH+11] Lei Shi, Qi Liao, Yuan He, Rui Li, Aaron Striegel, and Zhong Su. SAVE: Sensor anomaly visualization engine. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 201–210. IEEE, 2011.

[SSB+08] Ratnesh K. Sharma, Rocky Shih, Cullen Bash, Chandrakant Patel, Philip Varghese, Mohandas Mekanapurath, Sankaragopal Velayudhan, and Manu Kumar, V. On building next generation data centers: Energy flow in the information technology stack. In *Proceedings of the 1st Bangalore Annual Compute Conference*, COMPUTE '08, pages 8:1–8:7, New York, NY, USA, 2008. ACM.

[Sta15] Petr Stanicek. Color Scheme Designer 3, 2015. http://colorschemedesigner.com/, accessed April 2015.

[SWvdW+11] Roeland Scheepens, Niels Willems, Huub van de Wetering, Gennady Andrienko, Natalia Andrienko, and Jarke J van Wijk. Composite density maps for multivariate trajectories. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2518–2527, 2011.

[SWvdWvW11] Roeland Scheepens, Niels Willems, Huub van de Wetering, and Jarke J van Wijk. Interactive visualization of multivariate trajectory data with density maps. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 147–154. IEEE, 2011.

[Tay07] James W Taylor. Forecasting daily supermarket sales using exponentially weighted quantile regression. *European Journal of Operational Research*, 178(1):154–167, 2007.

[TGC03] Marjan Trutschl, Georges Grinstein, and Urska Cvek. Intelligently resolving point occlusion. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 131–136. IEEE, 2003.

[TGM83] Edward R. Tufte and P. R. Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.

[THo00] Tsuyoshi Taki and Jun-ichi Hasegawa. Visualization of dominant region in team games and its application to teamwork analysis. In *Computer Graphics International, 2000. Proceedings*, pages 227–235. IEEE, 2000.

[Tob87] Waldo Tobler. Experiments in migration mapping by computer. *The American Cartographer*, 14(2):155–163, 1987.

[Tuf90] E.R. Tufte. *Envisioning information*. Number Bd. 914 in Envisioning Information. Graphics Press, 1990.

[UCE07] UCEI (University of California Energy Institute), Berkeley, California, CA. New york city building energy map, 2007. http://www.visualizing.org/visualizations/new-york-city-building-energy-map.

[Uni10] United States Energy Information Administration. Annual energy review, 2010.

[US 08] US Department of Energy. Energy efficiency trends in residential and commercial buildings, 2008. http://apps1.eere.energy.gov/buildings/publications/pdfs/corporate/bt_stateindustry.pdf.

[Vas97] I.R. Vasiliev. Mapping Time. *Cartographica*, 34, 1997.

[VJN+15] Katerina Vrotsou, Halldor Janetzko, Carlo Navarra, Georg Fuchs, David Spretke, Florian Mansmann, Natalia Andrienko, and Gennady Andrienko. SimpliFly: A Methodology for Simplification and Thematic Enhancement of Trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 21(1):107–121, January 2015.

[vLBSF13] Tatiana von Landesberger, Sebastian Bremm, Tobias Schreck, and Dieter Fellner. Feature-based Automatic Identification of Interesting Data Segments in Group Movement Data. *Information Visualization Journal*, 2013.

[VW90] M. Visvalingam and J. D. Whyatt. The Douglas-Peucker Algorithm for Line Simplification: Re-evaluation through Visualization. *Comput Graph Forum*, 9(3):213–225, September 1990.

[VW93] M. Visvalingam and J. D. Whyatt. Line generalisation by repeated elimination of points. *The Cartographic Journal*, 30(1):46–51, 1993.

[VWF09] Fernanda B Viegas, Martin Wattenberg, and Jonathan Feinberg. Participatory visualization with wordle. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1137–1144, 2009.

[VWVdW99] Jarke J Van Wijk and Huub Van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 73–78. IEEE, 1999.

[WAM01] Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing time-series on spirals. In *proceedings of the IEEE Symposium on Information Visualization*, pages 7–13, 2001.

[War08] Colin Ware. *Visual Thinking for Design*. Morgan Kaufmann, 2008.

[WD08] Jo Wood and Jason Dykes. Spatially Ordered Treemaps. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1348–1355, 2008.

[Wik15] Formation (association football), 2015. http://en.wikipedia.org/wiki/Formation_(association_football), accessed April 2015.

[Win60] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342, 1960.

[WK12] Lei Wang and Arie Kaufman. Importance driven automatic color design for direct volume rendering. *Computer Graphics Forum*, 31(3pt4):1305–1314, 2012.

[WLY+13] Zuchao Wang, Min Lu, Xiaoru Yuan, Junping Zhang, and Huub Van De Wetering. Visual traffic jam analysis based on trajectory data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2159–2168, 2013.

[WVDWVW09a] N. Willems, H. Van De Wetering, and J.J. Van Wijk. Visualization of vessel movements. *Computer Graphics Forum*, 28(3):959–966, 2009.

[WVDWVW09b] Niels Willems, Huub Van De Wetering, and Jarke J. Van Wijk. Visualization of vessel movements. *Computer Graphics Forum*, 28(3):959–966, 2009.

[WWR+06] Leland Wilkinson, D Wills, D Rope, A Norton, and R Dubbs. *The grammar of graphics*. Springer, 2006.

[You15] Behavior of a back-four formation (German), 2015. http://www.youtube.com/watch?v=P_pFKGDEgUc, accessed April 2015.

[ZM12] Hai-xiang Zhao and Frédéric Magoulès. A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews*, 16(6):3586–3592, 2012.

Colophon

This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Arno Pro, designed by Robert Slimbach in the style of book types from the Aldine Press in Venice, and issued by Adobe in 2007. The layout of this thesis is an adapted version of a template released under the permissive MIT (X11) license. The original template can be found online at github.com/suchow/ or at suchow@post.harvard.edu.