Hierarchical Temporal Patterns and Interactive Aggregated Views for Pixel-based Visualizations

T. Lammarsch[*], W. Aigner[*], A. Bertone[*], J. Gärtner[†], E. Mayr[‡], S. Miksch[*], and M. Smuc[‡]

[*] Department of Information and Knowledge Engineering, Danube University Krems, Austria

www.donau-uni.ac.at/ike

[†] XIMES GmbH

www.ximes.com

[‡] Department for Knowledge and Communication Management, Danube University Krems, Austria

www.donau-uni.ac.at/wuk

Abstract

*Many real-world problems involve time-oriented data. Time data is different from other kinds of data—explicitly harnessing the structures of time in visualizations can guide and support users' visual analysis processes. State-of-the-art visualizations hardly take advantage of the structures of time to aid users in understanding and exploring the data. To bring more flexibility to the analysis process, we have developed interactive visual methods incorporating the structures of time within a pixel-based visualization called GROOVE (**gr**anular **o**verview **ove**rlay). GROOVE uses different techniques to visualize time-oriented data by overlaying several time granularities in one visualization and provides interactive operators, which utilize the structures of time in different ways to capture and explore time-oriented data.*

1 Introduction

The natural structures of time, such as years and seasons, influence the behavior of persons. For instance, the weekly time structure is tightly connected to business (market), religious (ceremonies), and other social activities (shopping behavior) [22]. The structures of time thereby influence activities but also provide a framework for analysis and comparison of time-oriented data.

In business contexts, human resource planners build personnel planning upon the analysis of time-oriented data. Such data is common in many domains, e.g. in call centers, health care, and public services. Typically, time-oriented data is multivariate and stems from heterogeneous data sources. The number of data points is usually much larger than the amount of data that fits into a single visualization, making methods, such as aggregation or scrolling necessary for conventional visualizations. Analyzers of temporal data have to gain an overview of these huge datasets to identify patterns and relations across different time granularities.

This work was carried out in the context of the project **DisCō**[1] which aims at designing novel interactive Visual Analytics methods that support users to discover temporal patterns and their relationships. In doing so, a large number of time-related aspects which can be classified in domain, primitives, determinacy, perspectives, and calendars [8] need to be considered. Following an initial analysis of our users and their tasks (see section 2), it became clear that one aspect that is especially worth exploring is the calendar aspect. The structures of time strongly determine phenomena which can be found in time-oriented data. For instance, the pattern of monthly sales may vary strongly due to differences in the arrangements of workdays, weekends, and holidays.

Since time-related attributes need to be treated differently from other data attributes and since time aspects belong to the most influential factors in business consulting, it is a key issue to provide appropriate visual and analytical methods to analyze them. Based on our users' needs, our focus is on appropriate visualizations that cope well with large time-oriented datasets in an intuitive and easily interpretable way by exploiting the inherent structures of time. State-of-the-art visualizations utilizing the structures of time hardly ease users to explore them in a way according to the tasks presented in section 2. To address this problem, we present a new visualization method, called **gr**anulary **o**verview **ove**rlay (GROOVE). Our approach enhances pixel-based visualizations to overcome present limitations and provides additional features for the exploration of time-oriented data. The ongoing development of GROOVE visualizations is complemented by a participa-

[1] **DisCō** project web site: http://www.donau-uni.ac.at/disco

tory design process to ensure that GROOVE visualizations meet the users' requirements.

2 Design Requirements

Our target users handle application scenarios of data analysis in different industrial or service sectors (e.g., transportation, call centers, retail, health care), and the public sector. The participatory design process started with initial interviews to assess their current task situation. As the intended users of our Visual Analytics methods work in different domains, the methods have to be applicable across domains and have to enable a wide, flexible access to the data (e.g., aggregation, filter, overview, detail) according to the task at hand. Users' tasks are to analyze, plan, and forecast personnel demand and to evaluate organizational interventions in this field.

Users reported that they are often confronted with ill-defined problems [7]. Although they are aware that they may have a problem or opportunity, they are frequently unable to nail it down. To solve such problems, temporal analysts have to identify temporal patterns in the data that are influenced by different time granularities. An example might illustrate the nature of our users' tasks:

A hospital wants to better align its personnel plans with customer requirements. To analyze data from this hospital our temporal analysts first have to understand and take into account two factors: (1) the internal, hospital-defined shifts, (2) the customers' demands. Then they have to find out how both are driven by structures of times—business days and seasons—and how "special events" such as holidays can influence the given data.

To be able to solve such a task our users have to (1) gain an overview of the data set, (2) identify relevant and define specific time granularities (e.g., one business day can last from 6am to 6pm, from midnight to midnight, etc.), and (3) find anomalies and relevant patterns, trends, and relations within this data set. GROOVE has to facilitate these analysis steps to support our users. For the first step it is necessary to provide an overview at an appropriate aggregation level, as the following example illustrates:

A business consultant analyzed a dataset from cash registers of a supermarket. He did not find any patterns using line plots of aggregated daily data. By visualizing the whole data set, he recognized unexpected anomalies that could be tracked down to temporally displaced accountings on some days.

Therefore, we concluded that aggregation should be kept at a minimum, as it can easily hide important facts. We also encountered that automated analysis is not a silver bullet, as it might miss information in the data that is unexpected and not covered in the search parameters.

For the second step, the Visual Analytics method has to build upon those time granularities that are relevant for the task at hand, and to provide smart control for them, as users work with time-oriented data of different granularities with inherent social and natural time structures.

For the third step, users can identify anomalies in the data set to identify "special events", data errors, and outliers and react on them by transformations of the visualization. Additionally, the Visual Analytics method should enable users to formulate hypotheses on patterns, relations, and trends.

Starting at the requirements defined above, we investigated related work to assess whether they can be applied in this context in the upcoming section.

3 Related Work

By studying surveys of visualizations for time-oriented data such as the one by Aigner et al. [1] it becomes apparent that scientific work on visualizations of time-oriented data applies a wide range of methods for improving different visualization techniques. These techniques solve typical problems of time-oriented data, but most focus on specific low-level tasks, or at least one single high-level task [2]. Temporal pattern searching has been improved in tools, like TimeSearcher [10], and van Wijk and van Selow [20] are making use of a data mining algorithm that classifies days as several different types. Methods like Focus + Context [5] are used in visualizations like TrendDisplay [4] or, as fisheye view, in calendar software like DateLens [3].

However, employing the structures of time is a method most visualizations neglect. Exceptions are cycle plots [6] which deal with the fact that time-oriented data often contains cycles and trends. As an example, Cleveland shows a time series that has a raster size of months, being several years long. He groups the data by month of year, showing the course of the whole data (without aggregation) for each month of year separately. Therefore, the cycle of months is separated from the trends among each month of year. This can be done by using any granularity for the grouping, like day of week or hour of day when there is a smaller raster and a shorter dataset.

Another exception are the recursive pattern visualizations by Keim et al. [11], as well as the multi-scale visualization by Shimabukuro et al. [14]. These pixel-based visualizations are able to use two or four different granularities in order to show patterns visually. More details are explained in section 4. These approaches are very promising regarding our requirements. Still, several topics are not considered by the authors. There is no visual guideline that helps users to mentally integrate the overview and detail level. The structures of time are only shortly covered in the discussion of suitable arrangements. Possibilities of user interaction are mostly neglected. These are the areas we address with GROOVE.

Other authors have made different approaches to improve recursive pattern visualizations. Luboschik and Schumann [12] propose to interactively generate exploded views similar to Figure 1. We consider this view appropriate for explaining the ordering of a pixel-based visualizations to new users.

Hao et al. [9] are going in another direction integrating different detail levels in pixel-based visualizations. Instead of aggregation, they rely on drawing blocks of different sizes. This method is sensible when a certain area of interest (e.g., the most recent data) is of special importance. For our tasks, relying on the structures of time is more suitable.

The existing visualizations are able to provide overview of data sets, but most of them do not take into account the structures of time. Recursive pattern and multi-scale visualizations provide few means to identify relevant granularities. With all existing visualizations, it is difficult to find patterns that are related to the structures of time, especially those that span more than two different granularities. Therefore, we developed GROOVE with the need to accomplish these tasks in mind.

4 GROOVE Concepts

Taking into account the definitions by Mackinlay [13], the visualization has to be expressive regarding the structures of time and the data. Owing to the complexity of the structures of time, in most cases it is impossible to express all aspects entirely as static visualization. We combine two methods to overcome this problem. The amount of complexity in one view is maximized by using Tufte's principle of micro-macro reading [18]. To further increase expressiveness, we resort to user interaction.

Pixel-based visualizations, as described by Keim et al. [11], use position in two dimensions and color to encode data. For time-oriented data, the position within a two-dimensional grid can be determined by one time granularity for the position along the horizontal axis (e.g., day of week) and the value of another time granularity for the position on the vertical axis (e.g., week of month). In such visualizations, it is necessary to find a way to encode the value of single data points, as both axes are used for time granularities. One way is using different colors for different values.

To further increase the number of granularities, it is possible to resort to recursive patterns. Figure 1 shows how four different granularities are used to order pixels recursively. In this example, the data is always ordered from left to right which is contrary to the proposal of Keim et al. who are in favor of a back and forth arrangement, at least on the detail level. Keim et al. state that values that are next to each other in the dataset should also be next to each other in the visualization. However, our users reported that they need an arrangement for comparing not only values next to each other, but also corresponding values among different granules of all the granularities used. In the example of Figure 1, such a task would be comparing the first months of each period of the winter semester with each other. Such an arrangement is easier to understand, too, as it is similar to the arrangement of socially established calendars.

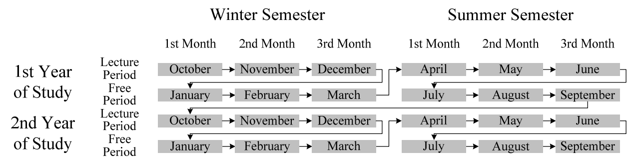

Figure 1: An example how recursive patterns are used to order data points. Each block represents one semester, ordered by month of period and period of semester. The blocks are ordered according to semester of study year and year of studies.

Even with a left-to-right-ordering, it is important for users to have a guideline to better discern the granularities. In compliance with the data-ink-ratio introduced by Tufte [19], we wanted this guide to provide further information about the dataset itself. This need is fulfilled in the tripartite multi-scale visualizations, which have been developed by Shimabukuro et al. [14]. They use a parallel overview of average values as a guidance and information source as well. However, the overview is not optimal as a guidance, as it is spatially apart from the detail. The eye of the beholder constantly has to jump between the two parts, which is rather a long distance, comprising the danger of mistakes, and straining the working memory. It is also difficult to mentally integrate patterns found among coarser granularities with patterns found among finer granularities. Therefore, we propose an overlay between the granularities of overview and detail. We have found three basic possibilities for the overlay.

4.1 Color-based Overlay

A color can be deconstructed in different ways. One possibility is a composition of hue, chroma, and lightness in CIELUV color space. By mapping the overview value on one aspect of the color (e.g., the hue) and the detail value on another one (e.g., the lightness), it is possible not only to overlay both levels, but, at the same time, do halve the space needed for the visualization. This visualization supports users in detecting patterns on an overview and detail level. The fact that the extraction of exact data values is very difficult does not impede this task [16].

An example of this kind of overlay is shown in Figure 2.a. Daily turnover data from a shop has been plotted for one year, each block depicting one month. The hue varies from blue to red for lower or higher monthly averages. The lightness for a particular day is higher in case the turnover of that day is higher. For illustration purposes, we have kept the arrangement simple and the amount of visualized data low. Empty space emerges due to the irregularities imposed by having to combine the granularities week and month.

4.2 Opacity Overlay

Several users prefer color palettes that require more than one aspect of colors to be used [16] (e.g., a palette that reflects the glow of heated metal, ranging from black over dark red and orange to yellow and white). Also, some datasets are especially suited for diverging palettes. In such cases, the use of color aspects for the overlay overly complicates the visualization. Alternatively to color-based overlay, we have included the possibility of using transparency for the overlay. The level of transparency can be adjusted by users interactively: The overview is always drawn. In the same space, the details are drawn using the same color palette, but semi-transparent. Users can freely adjust the level of opacity, shifting the logical focus back and forth between overview and detail without having to shift their visual focus. An example visualization, using the same data and arrangement as Figure 2.a, is shown in Figure 2.b. The opacity here amounts to 0.5, meaning that the color for each pixel is determined to 50% by the average value and to 50% by the detail value. It is difficult to impart the strengths of this variant in a static view, as the visualization is highly dependent on user interaction.

(a) (b)

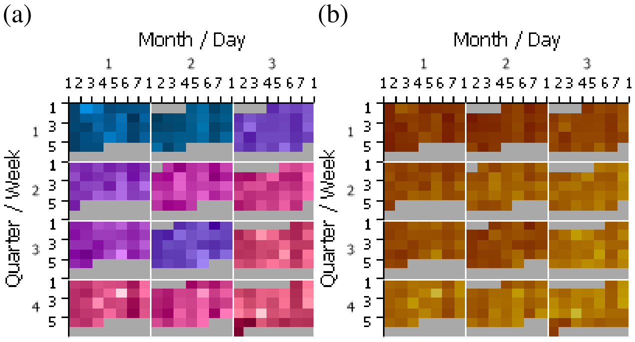

Figure 2: (a) Color-based overlay: Low average values are displayed as blue hues, high average values as red hues. Low detail values are displayed as dark pixels, high detail values as light pixels.
(b) Opacity overlay: Dark red represents low and light yellow represents high values. In this example, a monthly overview is combined with daily details, using an opacity of 0.5.

4.3 Spatial Overlay

Instead of showing overview and detail in exactly the same space, it is also possible to show them spatially very close to each other. In this GROOVE variant, the detail view of each block is surrounded by a border in a color based on the overview value. Thereby, no restrictions are imposed on the palette. The visualization does not depend on a user interaction like the one from subsection 4.2, but is very suitable for the ones presented in subsection 4.5.

Figure 3 shows the dataset and arrangement from Figure 2, this time using spatial overlay. Each month is surrounded by a border that visually encodes the average monthly value. In addition, we have also added weekly averages in this visualization, shown as bars above each week. Below each of these bars, the days of that week are shown in a row. We have added labels inside the visualization as an additional guideline for understanding.

This GROOVE variant requires considerably more space than the other variants, thus constraining the amount of data that can be visualized. On the other hand, the color coding is easier with spatial overlay. Therefore, we consider this variant as most comparable to the multi-scale visualization by Shimabukuro et al. [14]. The novel arrangement, however, makes the integration of overview and detail much easier for users.

4.4 Granularity Definition and Arrangement Variants

Based on the initial recursive pattern arrangement we used for GROOVE, users can gain new insights into different temporal patterns by interactively changing the order of granularities while keeping the same set of granularities. For instance, in Figure 2, it is relatively easy to compare one hour with the same hour of day on the days before and after. By switching the granularities of hour and for 5-minute-interval, it is possible to compare one hour with the same hour on the same day of week in neighboring weeks. This is equivalent to rotating the blocks by 90 degrees.

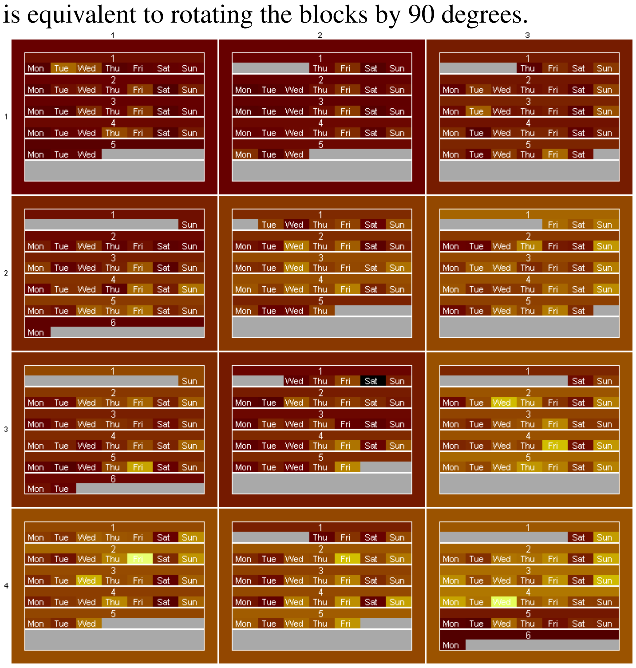

Figure 3: Spatial overlay: Dark red represents low values, light yellow represents high values. Each month is surrounded by a border showing the average value. Each week's average is shown by a bar, with the daily detail below.

When configuring granularities, it is important to keep in mind that they do not simply partition time. Granularities such as business week, leaving out weekends, or holidays, leaving out work days are examples of a wide range of possible applications. These example contain gaps of undefined time, enabling users to actively hide spans that are not important. Also, start and end times of granularities do not need to agree with common social defaults—a day can start and end at any time, e.g., at the time a shift change occurs. It is possible to use automated data mining methods to define granularities, instead of having users defining them.

Sometimes, it is beneficial to change the order of overview and detail. By ordering the blocks according to finer granularities (in our previous examples, that would be day and week) and the values inside the blocks according to coarser granularities (e.g., quarter and month), it is also possible to gain new insights. The development over the year becomes visible for each day of month separately. Therefore, monthly cycles become separated from the trend over the year. This is a an effect similar to the one in Cleveland's cycle plots [6].

For GROOVE, there are actually more variants in arrangement than the ones defined by Keim et al. for recursive patterns. These variants stem from the separate detail layers. Important variants we have encountered are (1) the use of a simple data arrangement of line-by-line for the detail level and calculating the average of each line for the overview level and (2) the use of a simple data arrangement of line-by-line for the detail level and calculating the average for certain fractions of the line for the overview level. The reason for using these variants is the fact that some users prefer the simple data arrangement over recursive pattern but still want to exploit more than two granularities. We actually found that there are several kinds of users who prefer different variants of GROOVE visualizations.

4.5 User Interaction Operations

With GROOVE, instead of using geometric zooming and panning, it is beneficial to use semantic zooming and paging. Paging is performed by defining a granularity coarser than the ones used by one view and showing only one granule at a time (based on user interaction). Semantic zooming can be performed by switching to coarser or finer combinations of granularities. As stated before, the granularities can be configured freely. However, users can also predefine combinations for quick switching.

The flexibility with respect to granularities is important, as the exploration of a dataset is often an iterative process. Assumptions made by users based on observations using one granularity configuration can be hardened or invalidated by switching to a different granularity configuration.

As an alternative to showing all possible data at once, it is possible to follow Shneiderman's Visual Information-Seeking Mantra [15]: overview first, zoom and filter, details on demand. It is possible to show only the overview level at first and interactively present the details for all blocks or only for some blocks once the user actively requests them. The details do not need to be presented immediately on a pixel level. It is also possible to switch from overview to showing only the rows of detail level and later switch to showing every pixel.

In the spatial overlay it is possible to treat each pixel on the detail level as a block on its own and drill down to further levels of even finer granularities. It is also possible to combine more than one kind of overlay, for example, using color overlay for a rough overview and spatial overlay for a more detailed overview as well as a detail view. Regarding arrangement, such a three-level overlay is "fully recursive" as defined by Keim et al. [11].

5 User Evaluation

We evaluated GROOVE visualizations in a user study [16, 17] as well as in a long-time process of active use in real-world scenarios of customer data by our industry collaborator.

5.1 User Study

We tested GROOVE mock-ups with respect to their intuitive comprehensibility and potential improvements in semi-structured interviews with six potential users. They were asked to think-aloud while exploring the data and to suggest improvements and additional interactions that could optimize their workflow. We analyzed the think-aloud protocols with respect to users' generation of insights. The most promising outcome of these mock-up test is that all users intuitively gained an understanding of GROOVE within short time. The metaphor of a calendar helped users to understand the visual alignment of the different time granularities. We coded and counted users' data insights considering the referred granularity. Users generated insights on an overview and on a detailed level; still, most insights focused on the overview (73%) and only a fourth on the details (27%). When we analyzed the time course of users' insights on both levels we found that most users are able to switch between the two levels of detail and overview and that they sometimes even integrate these levels. But there are also some indicators showing that the small number of detailed insights was due to the mock-ups of GROOVE being more an overview than a detail visualization. This is mitigated by the fact that there is the possibility of interactive semantic zooming that we could not use in our mock-ups. Furthermore, there is always the possibility for users to choose further visualizations once they have gained initial insights.

5.2 GROOVE in Real-world Application

The business consultants of our collaborator found GROOVE to be a valuable asset to gain much information from a dataset in a short amount of time. GROOVE also enabled them to correct misinterpretations they had made using conventional visualizations. They used GROOVE for several tasks in preprocessing (before trying to understand the data) as well as at later stages of data analysis:

1. Detection of structural changes (e.g., starting at a certain point of a time period, value levels are different for a certain time of day)
2. Finding missing values (which cannot easily be done automated as they are difficult to distinguish from time periods when there was no data to record)
3. Determining whether data has been collected at a wrong time or with wrong time stamps
4. Defining special events (e.g., refitting, stocktaking) when they have not been accounted for in data collection
5. Detecting cycles and trends

The flexibility of granularity definition has been adopted very well, especially regarding the definition of different variants of granularities (e.g., days from 6 am to 6 am). The business consultants also applauded the ability to define more abstract granularities like “week type” with granules like “5-day-business-week”, “week with holiday on Friday”, “week with holiday on Monday” and so on, in order to analyze the effect of the holidays on work days.

6 Discussion and Future Work

We have analyzed the need of users from business consulting to combine their domain knowledge with Visual Analytics by using the structures of time. Our main task for this paper is the development of a visualization that enables users to spot patterns and phenomena related to the structures of time visually. Furthermore, the visualization needs to provide users the insight into the dataset needed to pursue an iterative process of information extraction. We have developed GROOVE visualizations containing novel methodologies, especially regarding the structures of time and user interaction. In testing GROOVE, we realized that it is generally able to solve the intended tasks, but also shows promise for a number of further tasks. As our users are normally confronted with datasets that are not only new, but also are likely to contain new patterns unknown to them, GROOVE exhibits potential advantages compared to automated and specialized solutions that are only looking for well-known phenomena. GROOVE especially supports initial and explorative examinations when there is no a-priori information about patterns or even tasks.

Pixel-based visualizations surpass other visualizations regarding the number of data points that can be drawn. Perceptually, it is very difficult and impractical to extract exact values from the color coded visualizations as compared to line or bar plots. However, this is not the main focus and a rough comparability is sufficient in our task context. Exact values are shown as tooltips when hovering the mouse cursor over a certain pixel. Using the lines and columns as a basis for showing the structures of time has a high effectiveness, as edge detection is a very basic feature of human vision, a fact that has been pointed out by Ware [21].

An important application area for GROOVE are situations when aggregated or filtered values hide or even distort information. The number of datapoints that can sensibly be shown by GROOVE is dependent on the size of the display. We have successfully displayed around 60,000 datapoints on 22” computer screens. An example with 8,064 datapoints is shown in Figure 4.

Our current state of introducing multi-dimensional datasets as well as methods of automated data mining into GROOVE also is in need of further development and investigation, regarding the needs of our users and new possibilities that open up. Particularly, we see great promise using custom granularities generated by data mining methods. The work of van Wijk and van Selow [20] shows that data mining algorithms which classify granules into several different types can be superior to simple aggregation. Such methods of classification might be a sensible alternative to the mean values we use for the overview level of GROOVE.

Acknowledgements This work has been supported by the program “FIT-IT Visual Computing” of the Federal Ministry of Transport, Innovation and Technology, Austria (Project number: 813388). We would also like to thank our colleagues from the University of Rostock for their input about overlay methods.

References

[1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing Time-Oriented Data - A Systematic View. *Computers & Graphics*, 31(3):401–409, June 2007.

[2] R. Amar and J. Stasko. Knowledge Precepts for Design and Evaluation of Information Visualizations. *IEEE TVCG*, pages 432–442, 2005.

[3] B. B. Bederson, A. Clamage, M. P. Czerwinski, and G. G. Robertson. DateLens: A Fisheye Calendar Interface for PDAs. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 11(1):90–119, 2004.

[4] D. Brodbeck and L. Girardin. Interactive Poster: Trend Analysis in Large Timeseries of High-Throughput Screening Data Using a Distortion-Oriented Lens with Semantic Zooming. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis03)*, pages 74–75. IEEE, 2003.

[5] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, 1999.

[6] W. Cleveland. *Visualizing Data*. Hobart Press, Summit, NJ, USA, 1993.

[7] A. Colman. *A Dictionary of Psychology*. Oxford University Press New York, 2001.

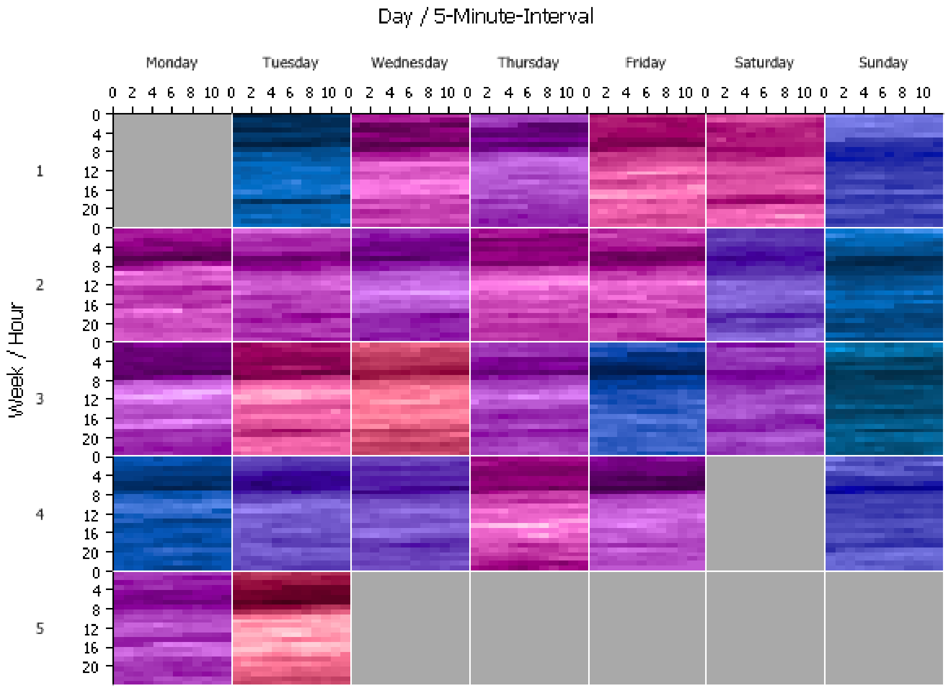

Figure 4: GROOVE with data of one month. The data is from police assignments. For intervals that are five minutes long, the number of deployed units is shown. Each block represents one day. Inside the blocks, the hours are shown in rows. Each row has one pixel for every five-minute-interval. On three days of this month, the data is missing.

[8] I. A. Goralwalla, M. T. Özsu, and D. Szafron. *An Object-Oriented Framework for Temporal Data Models*, pages 1–35. Springer, 1998.

[9] M. Hao, U. Dayal, D. Keim, and T. Schreck. Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. In *Proc. of Eurographics/IEEE-VGTC Symposium on Visualization*, pages 27–34, 2007.

[10] H. Hochheiser. Interactive Querying of Time Series Data. In *CHI '02 extended abstracts on Human Factors in Computing Systems*, pages 552–553. ACM Press, 2002.

[11] D. Keim, H.-P. Kriegel, and M. Ankerst. Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data. In *Proceedings of IEEE Visualization (Vis95)*, pages 279–286, 1995.

[12] M. Luboschik and H. Schumann. Explode to Explain-Illustrative Information Visualization. In *Proceedings of the 11th International Conference Information Visualization*, pages 301–307. IEEE Computer Society Washington, DC, USA, 2007.

[13] J. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*, 5(2), 1986.

[14] M. Shimabukuro, E. Flores, M. de Oliveira, and H. Levkowitz. Coordinated Views to Assist Exploration of Spatio-Temporal Data: A Case Study. In *Proceedings of the 2nd International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV04)*, pages 107–117. IEEE, July 13 2004.

[15] B. Shneiderman. The eyes have it: a task by data type taxonomy for informationvisualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343, 1996.

[16] M. Smuc, E. Mayr, T. Lammarsch, A. Bertone, W. Aigner, H. Risku, and S. Miksch. Visualizations at First Sight: Do Insights Require Traininig? In *Proceedings of 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society (USAB 2008)*, pages 261–280. Springer, November 2008.

[17] M. Smuc, E. Mayr, T. Lammarsch, W. Aigner, S. Miksch, and J. Gärtner. To Score or not to Score? How to Triple Insights for Participatory Design. In *IEEE Computer Graphics and Applications, Special Issue*, In Print.

[18] E. Tufte. Envisioning Information. Cheshire, CT, 1990.

[19] E. Tufte and G. Schmieg. The Visual Display of Quantitative Information. *American Journal of Physics*, 53:1117, 1985.

[20] J. J. van Wijk and E. R. van Selow. Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of the IEEE Symposium on Information Visualization 1999 (InfoVis99)*, pages 4–9, 1999.

[21] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2004.

[22] E. Zerubavel. *The Seven Day Circle: The History and Meaning of the Week*. University Of Chicago Press, 1989.