TimeElide: Visual Analysis of Non-Contiguous Time Series Slices

Michael Oppermann* Luce Liu† Tamara Munzner‡

University of British Columbia

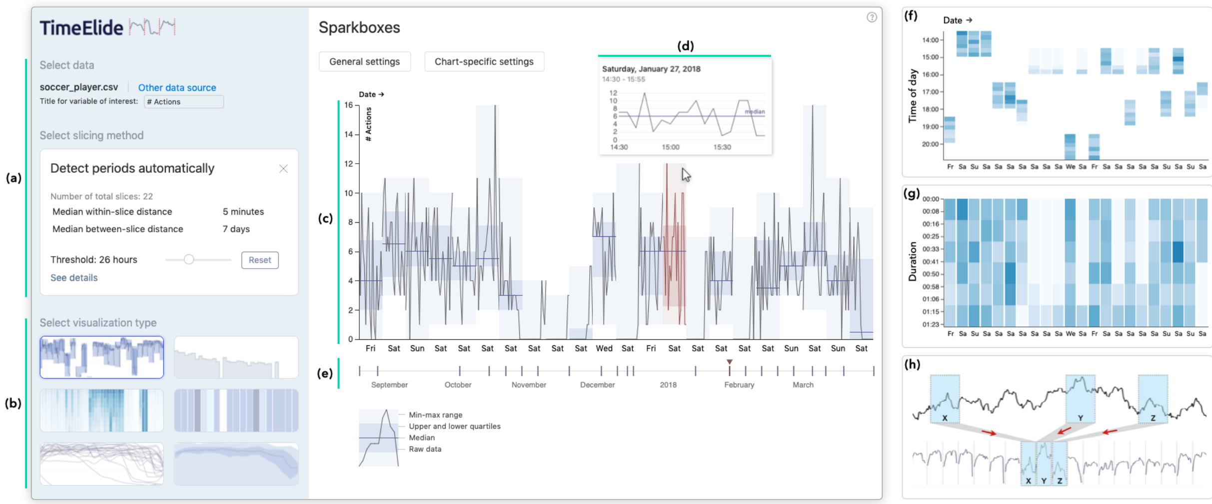

Figure 1: TimeElide. (a) Select data source and slicing method, (b) select visualization type, (c) sparkbox visualization to show the number of per-player events during 22 soccer matches, (d) tooltip for the currently active slice, and (e) global timeline. Alternative visual encodings: (f) 2D heatmap with absolute within-slice timing, and (g) 2D heatmap with normalized within-slice timing. (h) Illustration of extracting non-contiguous time series slices according to given periods of interest (manual slicing).

Abstract

We introduce the design and implementation of TimeElide, a visual analysis tool for the novel data abstraction of non-contiguous time series slices, namely time intervals that contain a sequence of time-value pairs but are not adjacent to each other. This abstraction is relevant for analysis tasks where time periods of interest are known in advance or inferred from the data, rather than discovered through open-ended visual exploration. We present a visual encoding design space as an underpinning of TimeElide, and the new sparkbox technique for visualizing fine and coarse grained temporal structures within one view. Datasets from different domains and with varying characteristics guided the development and their analysis provides preliminary evidence of TimeElide's utility.

Index Terms: Human-centered computing—Visualization—Visualization systems and tools;

1 Introduction

Time series data is often visualized in a way that emphasizes the continuous semantics of discretized data samples, where there is a single contiguous range of data between a start and end time. Visualization systems support this data type through interactive exploration in order to help users in identifying interesting time periods. We consider a problem scenario that differs from this standard paradigm. We want to support analysis tasks that relate to specific time periods of interest known in advance rather than determined on the fly, and that pertain to dozens of non-contiguous time intervals rather than just a few selected time ranges. Examples with these temporal data characteristics can be found across many domains: working shifts for job scheduling, public holidays as times of heavy museum attendance, occasional bike rides recorded with a smartphone, and so on. We refer to the combination of known time intervals and time series data as *non-contiguous time series slices*.

When working on visualizations for building occupancy [15] we repeatedly experienced the need to analyze this type of non-contiguous temporal data. We found other examples in several domains, but existing tools do not support such analysis explicitly.

To fill this gap, we contribute the design and implementation of TimeElide, a domain-agnostic tool for visualizing non-contiguous time series slices. We present a detailed characterization of this data type and a visual encoding design space that underlies TimeElide. As a secondary contribution, we propose the new *sparkbox* technique for visualizing coarse and fine grained temporal structures within one view. We provide preliminary evidence of the system's utility based on multiple real-world datasets and four usage scenarios.

Similar in spirit to Trrack [6], we present our contributions as an *application note* to the visualization community: a short paper describing the development of, and the design rationale behind our open-source visualization tool that supports the visual analysis of a specific data type, relevant for many domains and use cases.

Online demo `https://cs.ubc.ca/group/infovis/time-elide`
Supp. material `https://osf.io/yqvmf/`

*e-mail: opperman@cs.ubc.ca
†e-mail: thisisluce@gmail.com
‡e-mail: tmm@cs.ubc.ca

2 Related Work

Visualizations for temporal data have been well studied and applied to many applications. Aigner et al. [1] characterized time-oriented data types and surveyed visualization techniques, but did not discuss the combination of time intervals and sequences of time-value pairs explicitly.

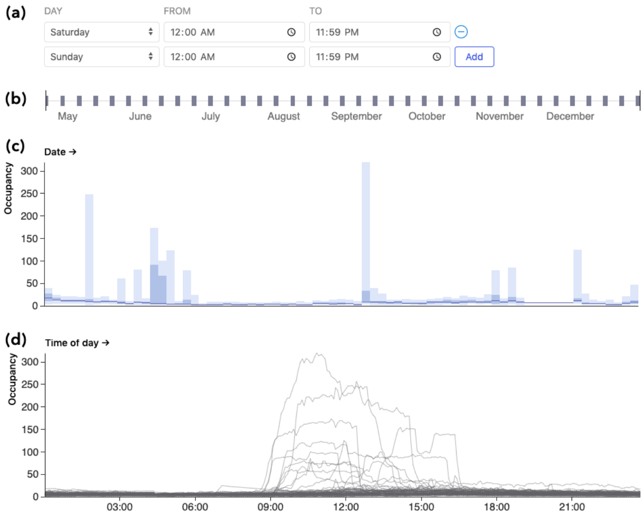

Figure 2: (a) Manually specify desired slices. (b) Global timeline shows the regularity of slice intervals. Building occupancy on weekend days visualized as (c) box-plot-bars and (d) multi-series line chart with absolute within-slice timing.

The bulk of the previous work applies to *contiguous time series*. We broadly classify methods to interactively explore and select time periods of interest into three groups: *navigation and filtering*, such as zooming & panning [7]; *lensing* to delimit a period of interest to be put in focus [4, 12, 22]; and *overview+detail linked views* to drill down to smaller temporal regions while maintaining the high-level context [9, 20]. These methods are typically limited to one time slice, or a very small number of them. Some exceptions exist, such as ChronoLenses [24] which combines lenses and linked views to let users assemble pipelines of many lenses for exploratory visual analysis. Lee et al. [13] introduced a tool that lets users select time slices of events that are visualized as node-link diagrams. However, all these methods are most suitable for a data-driven exploration; in contrast, our focus is on the investigation of known time periods.

Although the term *time slice* appears in the visualization literature, it has been used for different purposes, such as to describe the segments in the ThemeRiver [8] visualizations where time periods are connected with continuous curves, or to mean the interactive selection of one time period of interest [23].

Considering the sparkbox technique in particular, some previous work focuses on *interval* or *event* data that does not have finer-grained internal structure within the interval [5, 11, 17]. Other approaches have been proposed for visualizing *multiple time series*, ranging from handling unevenly-spaced samples [2] to navigating very large sets of time series [14]. In contrast, our goal with the sparkbox technique is to provide an adequate level of detail for the internal structure given a moderate number of time series. Previous methods leave out the details that our technique illustrates, as they are not aimed at our proposed data abstraction. Szafir et al. [21] used composite charts, composed of line graphs on top of bar charts, in their evaluation of aggregations in time series visualization. The authors aggregated data from continuous intervals while we assume non-contiguous, unevenly spaced periods. The most related previous work is *box-plot-bars* [3], which we directly build on.

3 Non-Contiguous Time Series Slices

We characterize data and tasks for non-contiguous time series slices.

3.1 Data

We used four real-world datasets with different characteristics and cardinalities to validate the utility of TimeElide:

- **Soccer player in-match activity** [16]: numbers of events (pass, shot, free kick, duel, ...) of one player, recorded during 22 matches and aggregated into 5-minute intervals.
- **Building occupancy** [15]: estimated occupants recorded in multiple university buildings every 5 minutes, over more than a year.
- **Bakery sales** [18]: number of items sold, logged over several months and aggregated into 15-minute bins.
- **Bike rides** [19]: 20 bike rides recorded with Strava and converted into CSV format by computing the median speed every 5 minutes.

All four datasets are available in the online demo, and further details about them are provided in SUP-1.

We distinguish between two main types of non-contiguous time series slices: (1) *existing*: data is naturally divided into slices since data is not collected during periods of non-interest (bike rides, soccer matches), (2) *specified*: data is continuously recorded but the analyst manually specifies time periods of interest, as illustrated in Fig. 1h (building occupancy).

Time slices may vary in length but should be within the same order of magnitude, and TimeElide only handles slices less than or equal to 24 hours. Data can be sampled at both regular intervals such as every 5 minutes, or at irregular intervals. A single time slice can contain dozens to hundreds of data points. A dataset can contain dozens to hundreds of time slices. The gap between time slices can be fixed (every Monday 8-10am) or variable (occasional bike rides at different times). The variable of interest, such as occupancy, speed, or number of actions, must be quantitative.

3.2 Tasks

We identified five abstract tasks to support the visual analysis of non-contiguous time series slices:

T1 **Extrema and Averages** within and across slices
T2 **Trends** across all slices
T3 **Patterns** within and across slices; outliers or unusual patterns
T4 **Comparison** between pairs of slices
T5 **Durations** of slices and how they change over time

Questions related to facilities management initially motivated and guided our investigation. Example questions elicited in previous interviews [15] include *When do people usually leave on Friday evenings? Are they coming and going?* and *Does it happen often that people occupy the space or are there irregular recurring events?*

From that domain-specific scenario, we abstracted four high-level tasks most pertinent to non-contiguous time series data, *T1-T4*. To ensure our approach generalizes, we then verified that these tasks align with questions users may ask when analyzing the other three datasets. We recognized the need to explore within-slice data (*T3*), to gain a better understanding of the gist of all slices, and to understand changes related to the temporal order (*T5*). We provide a detailed mapping from domain language into abstract tasks in SUP-2.

Finally, we added Task *T5* for analyzing and comparing the duration of slices, which was not relevant for the occupancy use case but is certainly applicable to other examples such as bike rides.

4 The TimeElide Tool

We introduce TimeElide, an open-source visual analysis tool, that instantiates a design space for visualizing non-contiguous time series slices. The accompanying video and online demo illustrate the look and feel of the interactive interface. Here, we first describe the tool itself, then cover the design space in depth in the subsequent section.

4.1 Overall Design

TimeElide incorporates a two-column layout with the sidebar on the left and the main view on the right, as shown in Fig. 1. The sidebar shows the three steps—*select data source*, *select slicing method*,

Figure 3: Visual encoding design space for non-contiguous time series slices. Supported, Partially supported, Not supported

and *select visualization type*—and allows users to navigate between those flexibly. The data source page or the visualization page are alternately displayed in the main view while the slicing method can be changed directly in the sidebar.

The user uploads a CSV file or selects an example on the data source page. The tool automatically detects existing slices or the user manually specifies custom slices. After the data source and slicing option have been chosen, TimeElide automatically visualizes the data as sparkboxes or as a 2D heatmap, depending on the number of slices and the available pixels in the main view.

Sparkboxes reveal finer-grained structure within slices, by showing the raw data in the foreground and aggregated data in the background, while the 2D heatmap scales better for a larger number of slices. The user can switch between six primary visualization types in the sidebar, shown in Fig. 1b, and can configure the visual representation using the settings menu above the visualization. Sparkboxes combine line charts and box-plot-bars, and therefore we group those three options and users can show or hide layers.

The global timeline (Fig. 1e) provides additional context by showing where each slice falls along the continuous range between start and end date, revealing potential irregularity in the timing. Linked highlighting connects the timeline with the main view.

After a series of usability tests, we refined the default settings, added explanations for the available visualization types, and provided the option to customize the axis title for the variable of interest.

4.2 Manually Specify or Automatically Detect Slices

Manual slicing is intended to be used when the user is interested in clearly defined, recurring periods (e.g., every Monday from 3-5pm). Automatic detection of existing slices is appropriate when the user cannot pinpoint exact start and end times of slices, such as bike rides that take place on different days of the week and vary in length.

To automatically detect existing slices, we determine the differences between consecutive timestamps in seconds, and eliminate the top and bottom 1% to minimize the influence of outliers. We then compute the standard deviation and use it as a threshold when iterating through the data to decide if a data point is within the current time slice or not. The threshold and the median within-slice and between-slice distances are displayed in the left sidebar of the interface. The user can also adjust this threshold manually.

Further details are provided in SUP-3 for manually specifying slices and SUP-4 for automatically detecting existing slices.

4.3 Implementation

TimeElide is a serverless web application composed of nested reactive components built using the Svelte[1] framework. All visualization types are wrapped in reusable components with minimal dependencies. Svelte compiles all components into efficient imperative code to allow quick updates of the Document Object Model which is particularly suitable for interactive visualizations with many elements.

5 Visual Encoding Design Space

We introduce a design space for visualizing non-contiguous time series slices. It includes our proposed visualization technique, sparkboxes, to support the analysis of within-slice and across-slice patterns simultaneously. Fig. 3 shows an overview of the design space.

5.1 Targeting Different Levels of Detail

The design space spans a set of eight visual encodings, that comprise different layouts and target different levels of detail (LoD), to support the outlined analysis tasks.

Non-contiguous slices are essentially individual time series where, in most cases, the order matters. A natural way to visualize these slices is to juxtapose individual **line charts** along the x-axis (Fig. 3b), which leads to a high LoD because the values are not aggregated. However, it may be difficult to discern trends (*T2*), especially in cases of high-frequency changes. The critical factor is the combination of the number of slices and the count of points within each slice.

Box-plot-bars [3] are an alternative visual encoding that are a minimalist representation of Tukey box plots displaying the min-max range, interquartile range, and the median (or average) value (Fig. 3c). Although box-plot-bars were originally proposed to visualize contiguous high-frequency time series data, they can also be used to show summaries of time slices (*T1*, *T2*).

Aggregating all values within a slice, such as in a **stepped area chart** or in **heat stripes** (Fig. 3d-e), facilitates overviews and helps users recognize trends and extrema. However, these compact representations may conceal important nuances within slices.

The **2D heatmap** addresses the trade-off between LoD and scalability (Fig. 3f). The slices are again aligned along the x-axis but instead of aggregating all values into a single number, each slice is divided into a desired number of vertical bins, depending on the

[1] `https://svelte.dev`

specified resolution. The per-bin aggregated values are visualized as color-coded cells in the 2D space.

The superimposed **multi-series line chart** promotes local comparisons of within-slice patterns [10], for use when the temporal order of slices is not relevant (Fig. 3g). However, distinguishing the identity of individual lines is challenging. The **banded multi-series line chart** (Fig. 3h) is an alternative to avoid the visual clutter in case of many slices. The within-slice values are binned and then aggregated across all slices: min-max range, interquartile range, and the median value. The banded multi-series chart helps users to better understand typical patterns but prohibits the option to look up individual time slices. TimeElide does not support panning and zooming for these two unordered encodings, as it does for all others.

Depending on the intended task, users may want to visualize the *within-slice timings* differently for time slices with varying lengths, such as bike rides. We distinguish between absolute duration and normalized duration, where the latter case means that all slice widths are equal. The y-axis in the 2D heatmap and the x-axis in both multi-series charts can also denote the absolute time of the day. Fig. 1f-g provides a comparison between absolute time and absolute duration for 2D heatmaps. In the soccer example, the absolute and normalized duration are equal because play time is standardized to 90 minutes.

5.2 The Sparkbox Technique

To present the coarse and fine grained temporal structure of non-contiguous time slices simultaneously, we propose **sparkboxes** — a combination of box-plot-bars and sliced line charts that are superimposed in the same space, as shown in Fig. 1c and Fig. 3a. The sliced line chart provides a high LoD but impedes seeing the overall structure at a glance. The box-plot-bars are more concise visually and capture the coarse-grained structure. Sparkboxes combine the advantages of both, providing a very high LoD for dozens of slices.

TimeElide supports zooming and panning but sparkboxes can be also used as a static representation without any manual interaction, for example in a print medium. Different choices for the color palette can shift the focus to selectively emphasize fine or coarse structures in the data in order to support different tasks. Our informal scalability testing suggests a minimum width of 20 pixels per slice, at our target cardinality of dozens of points per slice; at smaller sizes, we consider the 2D heatmap to be superior.

6 Usage Scenarios

We present two example usage scenarios to illustrate the user experience in TimeElide. The first scenario is based on soccer match data that we acquired online, and the second exemplifies building occupancy questions raised by a previous project stakeholder.

6.1 Soccer Player In-Match Activity

A soccer player wants to understand how their performance changed throughout the season and during each of the 22 matches. The player uploads the file containing all events and selects the option to automatically detect existing slices (Fig. 1a). TimeElide detects 22 matches of 90 minutes each and shows a sparkbox visualization (Fig. 1c). The activity varies within each match, but falls into the range of 2-10 actions across each 5-minute period for most games. They hover over a sparkbox to see more details in a popup (Fig. 1d). They see they were only substituted in for the last few minutes in six of the matches, because the number of actions is zero for most of the time. On the global timeline (Fig. 1e), below the sparkboxes, they notice that they did not play at all in September, but note that the activity during the first match after their return looks normal.

The soccer player changes the visualization type to *2D heatmap* to analyze if there are any differences between afternoon and evening games (Fig. 1f). The slices are again juxtaposed along the x-axis. The y-scale is set to *absolute time* and shows the range between the earliest match start time (1:30pm) and latest end time (9:00pm). The number of in-match actions are aggregated to bins and shown as color-coded rectangles. They adjust the number of bins to 30, so each bin lasts 15 minutes. They quickly spot that their coach typically substituted them in at 2:30pm during afternoon matches.

6.2 Building Occupancy on Weekend Days

A large university campus serves as a test bed for a new technology to record building occupancy. A custodial analyst, who oversees facilities across the campus, wants to leverage the data to investigate claims that lecture rooms in a building are untidy after weekends.

They upload a CSV file to TimeElide that contains 5-minute samples of occupancy information, spanning a period of several months. The tool automatically suggests the file should be parsed as two columns with time and occupancy values, and the analyst confirms the choice. They choose to manually specify slices and select Saturdays and Sundays as time periods of interest (Fig. 2a). TimeElide slices the continuous recordings by removing extraneous data outside of the given periods, and visualizes the result as sparkboxes. The analyst sees from the global timeline that the dataset contains weekends between April and December (Fig. 2b). After they use the settings menu to collapse slices without any data available, they then choose the visually concise *box-plot-bars* technique (Fig. 2c) to show only aggregate data. They notice a few weekends with significant activity, and hover over the slices to learn that most events occurred on Saturdays. In general, the occupancy rate is relatively low which is expected for a building that is primarily used for lectures. They do not recognize any repeating patterns.

The analyst selects the visualization type *multi-series line chart*, which disaggregates completely to show all time slices superimposed (Fig. 2d), to examine how the occupancy typically changes throughout the day. They learn that weekend activity rapidly decreases after 4-5pm and no evening events take place. Sending a custodian for a check-up in the late afternoon should suffice and the night shifts can remain unchanged.

7 Discussion and Future Work

We validated the effectiveness of TimeElide and the underlying design space with real-world data from four domains, illustrated through two usage scenarios in Sect. 6 and two more in SUP-5.

We made a simplifying assumption in designing the TimeElide user interface by restricting slices to a maximum of 24 hours. However, the presented design space is applicable to large-scale slices beyond the 24 hour limit, for example, to analyze the GDP during recession periods over the last century.

TimeElide focuses on the detailed investigation of time slices belonging to one time series, such as occupancy in one building or region over time. Future work could further investigate techniques and extensions of the interface for analyzing and comparing multiple independent time series at the overview level (e.g., analyze multiple buildings). Visual encodings with a lower LoD, such as stepped area charts or heat stripes, highlight the coarse structure of slices in a way that scales to lower resolutions, and can be used, for example, in a small multiple visualization with dozens of time series.

8 Conclusion

In this paper we have presented a characterization and abstraction of non-contiguous time series slices. Visualization systems tend to focus on the unbounded exploration of time series rather than the investigation of known time periods of interest. To better support this latter type of analysis, we introduce the open-source tool TimeElide, and a flexible design space including the sparkbox technique that provides the conceptual foundation. All visualization designs were informed by a detailed analysis of data characteristics and alternative visual encodings. Two scenarios from different domains illustrate how TimeElide facilitates an in-depth analysis of non-contiguous time series slices.

References

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011.

[2] A. Aris, B. Shneiderman, C. Plaisant, G. Shmueli, and W. Jank. Representing unevenly-spaced time series data for visualization and interactive exploration. In *IFIP Conf. on Human-Computer Interaction*, pp. 835–846. Springer, 2005.

[3] R. Bade, S. Schlechtweg, and S. Miksch. Connecting Time-Oriented Data and Information to a Coherent Interactive Visualization. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pp. 105–112, 2004.

[4] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and Magic Lenses: The See-Through Interface. In *Proc. SIGGRAPH 93, Annual Conference Series*, pp. 73–80, 1993.

[5] M. Brehmer, B. Lee, B. Bach, N. H. Riche, and T. Munzner. Timelines Revisited: A Design Space and Considerations for Expressive Storytelling. *IEEE Trans. Visualization and Computer Graphics*, 23(9):2151–2164, 2017.

[6] Z. Cutler, K. Gadhave, and A. Lex. Trrack: A library for provenance-tracking in web-based visualizations. In *IEEE Visualization Conference (VIS)*, pp. 116–120, 2020. doi: 10.1109/VIS47514.2020.00030

[7] R. Dachselt, M. Frisch, and M. Weiland. FacetZoom: A Continuous Multi-Scale Widget for Navigating Hierarchical Metadata. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pp. 1353–1356, 2008.

[8] S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: Visualizing Theme Changes Over Time. In *Proc. IEEE Symp. on Information Visualization (InfoVis)*, pp. 115–123, 2000.

[9] H. Hochheiser and B. Shneiderman. Dynamic Query Tools for Time Series Data Sets: Timebox Widgets for Interactive Exploration. *Information Visualization*, 3(1):1–18, 2004.

[10] W. Javed, B. McDonnel, and N. Elmqvist. Graphical Perception of Multiple Time Series. *IEEE Trans. Visualization and Computer Graphics*, 16(6):927–934, Nov 2010.

[11] N. W. Kim, S. K. Card, and J. Heer. Tracing Genealogical Data with TimeNets. In *Proc. Intern. Conf. on Advanced Visual Interfaces (AVI)*, pp. 241–248, 2010.

[12] R. Kincaid. SignalLens: Focus+Context Applied to Electronic Time Series. *IEEE Trans. Visualization and Computer Graphics*, 16(6):900–907, 2010.

[13] A. Lee, D. Archambault, and M. Nacenta. Dynamic network plaid: A tool for the analysis of dynamic networks. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pp. 1–14, 2019.

[14] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: Interactive Visual Exploration of System Management Time-Series Data. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pp. 1483–1492, 2008.

[15] M. Oppermann and T. Munzner. Ocupado: Visualizing location-based counts over time across buildings. *Computer Graphics Forum*, 39(3), 2020. doi: 10.1111/cgf.13968

[16] L. Pappalardo, P. Cintia, A. Rossi, E. Massucco, P. Ferragina, D. Pedreschi, and F. Giannotti. A public data set of spatio-temporal match events in soccer competitions. *Nature Scientific Data*, 6(1):1–15, 2019.

[17] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman. LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records. In *Proc. American Medical Informatic Association Annual Fall Symposium*, pp. 308–312, 2003.

[18] S. Sarwar. Transactions from a bakery. `https://www.kaggle.com/sulmansarwar/transactions-from-a-bakery`, 2018.

[19] S. Sharan. Strava bike rides. `https://www.kaggle.com/synapsizeus/strava-bike-rides-from-2020-gpx-files`, 2020.

[20] C. Stolte, R. Bosch, P. Hanrahan, and M. Rosenblum. Visualizing Application Behavior on Superscalar Processors. In *Proc. IEEE Symp. on Information Visualization (InfoVis)*, pp. 10–17, 1999.

[21] D. Szafir, M. Correll, and M. Gleicher. Task-Driven Evaluation of Aggregation in Time Series Visualization. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pp. 551–560, 2014.

[22] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. A Survey on Interactive Lenses in Visualization. *Proc. EuroVis*, 3(2), 2014.

[23] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the Non-Visual: Spatial Analysis and Interaction With Information From Text Documents. In *Proc. IEEE Symp. on Information Visualization (InfoVis)*, pp. 51–58, 1995.

[24] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory Analysis of Time-series With ChronoLenses. *IEEE Trans. Visualization and Computer Graphics*, 17(12):2422–2431, 2011.