

# Periphery Plots for Contextualizing Heterogeneous Time-Based Charts

Bryce Morrow, Trevor Manz, Arlene E. Chung, Nils Gehlenborg, David Gotz

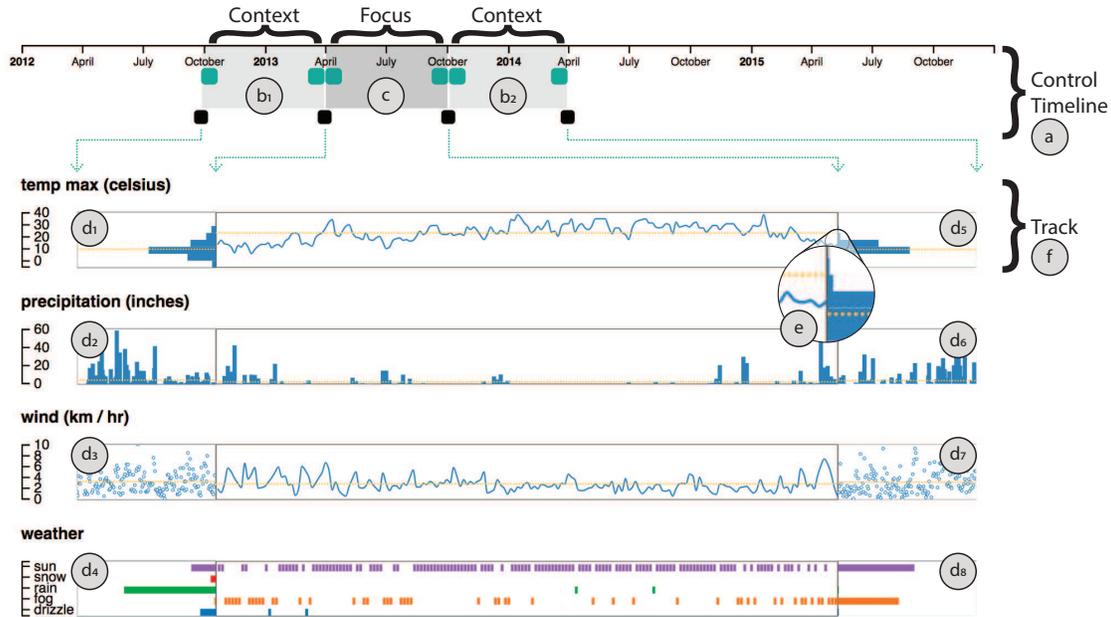


Figure 1: Periphery plots used within a visualization of multi-variate climate data. The (a) control timeline includes interactive controls to define (b<sub>1</sub>, b<sub>2</sub>) context and (c) the focus zone. This example includes four tracks (f), each with (d<sub>1</sub>-d<sub>8</sub>) two periphery plots. (e) Annotations support quick comparisons, such as the seasonal variation between average maximum temperatures in summer months (the focus region) and the preceding and subsequent winter periods (d<sub>1</sub>, d<sub>5</sub>).

## ABSTRACT

Patterns in temporal data can often be found across different scales, such as days, weeks, and months, making effective visualization of time-based data challenging. Here we propose a new approach for providing focus and context in time-based charts to enable interpretation of patterns across time scales. Our approach employs a focus zone with a time and a second axis, that can either represent quantities or categories, as well as a set of adjacent periphery plots that can aggregate data along the time, value, or both dimensions. We present a framework for periphery plots and describe two use cases that demonstrate the utility of our approach.

**Keywords:** Time-based data, Focus + context techniques, Health informatics, mHealth, Patient-generated health data.

## 1 INTRODUCTION

Temporal data is found frequently in various application domains and comes in many forms. Data can range from sparse point-based and interval events (e.g., procedures and medication changes) to densely sampled time series with repeated scalar measurements (e.g., longitudinal climate measurements). Often, data have combinations of variables and data types (e.g., daily stock prices and trading volumes along with the dates of key news events).

Based on these temporal data sets, users must often explore data and answer complex temporal questions. For example, consider a clinician exploring data gathered from mobile health (mHealth) devices (e.g., fitness trackers and mobile phones) along with patient-

reported data (e.g., diet and patient-reported outcomes about health status) and clinical data (e.g., medications and diagnoses). To make sense out of this kind of data, one must: (1) view data across a range of time scales (e.g., daily exercise fluctuations vs. monthly patient-reported outcomes vs. semi-annual doctor visits), (2) correlate across variables (e.g., does a change in medication correlate with changes in patient-reported outcomes?), and (3) compare across time intervals (e.g., has a patient's current condition improved compared to historical data?).

Given the ubiquity of temporal data, a wide variety of visual analytics techniques have been proposed for representing time-based events. This includes multi-variate techniques as well as a range of different *focus + context* methods. However, these techniques often focus on homogeneous types of data (e.g., scalar valued time series) or have other constraints that limit their use (see Section 2). This paper presents *periphery plots*, a technique for augmenting time-aligned temporal charts with a set of horizontally aligned peripheral views that provide multi-resolution contexts before and after a focal temporal period. This approach is applicable to heterogeneous data types and varied time samplings (regular vs. irregular; hourly vs. annually). Moreover, it is capable of representing a variety of previously proposed *focus + context* methods, while simultaneously supporting new contextualization capabilities. More specifically, this research has three key contributions:

- **A Framework for Periphery Plots.** The framework includes key concepts of focus and context zones. It also includes an axis preservation design space for plots that allow visualizations with varying levels of summarization.
- **A Periphery Plot Toolkit.** The framework is implemented within an open-source toolkit that includes a collection of specific view types within a general purpose architecture.

- **Example Applications.** Two use cases (longitudinal climate data analysis and pattern finding within patient-generated health data) demonstrate the utility of periphery plots.

## 2 RELATED WORK

This section provides a brief review of the most relevant themes from prior research on temporal data visualization.

### 2.1 Visual Representations of Temporal Data

Time series data sets are ubiquitous and are most commonly represented as line charts. Prior research has mainly focused on developing alternative visual encodings to convey equal information while using less space, such as horizon charts [8] and braided graphs [12]. For horizon charts, interpretability can be preserved and improved while reducing space usage by optimizing chart parameters (number of bands, height of chart) [10]. Another approach is to create complex visual encodings with high information density. For example, both ThemeRiver [9] and RankExplorer [18] utilize stacked graphs as a base chart to display categorical partitions of multi-variate time series data, while the latter layers additional glyphs to represent changes in information density between categories and time points. Alternatively, Chronodes condenses multiple features into a single event timeline to highlight frequent event sub-sequences, where glyphs encode unique health related features [16]. The periphery plots framework is compatible with a wide range of these aforementioned techniques because it supports the use of both common and bespoke chart types. Moreover, it can dynamically transition between types in response to changes in data density.

### 2.2 Detail and Context Interfaces

Exploratory analysis often requires users to interact with data at a level of detail that cannot be displayed on a single screen. While interactive methods such as *zooming & panning* offer the ability to view details otherwise hidden due to summarization, these interactions alone may obscure context and hinder interpretation due to narrow fields-of-view [4]. A variety of techniques have been developed to address this challenge, including *overview + detail* and *focus + context* approaches. The former employs spatial separation to partition global and detailed information (e.g., Stack Zooming [11]) while the latter combines these displays in a single continuous view (e.g., PerspectiveWall [15] or fish-eye lenses [6, 19]).

Complementary approaches which incorporate these schemes with interactive techniques have yielded powerful dynamic tools for visual exploration, particularly for temporal data. SignalLens utilizes *focus + context* with panning to enable rapid navigation of time series data, but users are unable to define the initial focus from a global overview [13]. In contrast, MultiStream [7] combines *overview + detail* and *focus + context*, linking interactive brush selections from a global overview to contextualize the focus of a fisheye distortion in a more detailed multi-resolution view. Our proposed periphery plot framework uses a similar combined *overview + detail* and *focus + context* approach. However, it focuses on views of context zones (rather than the focus) and includes a design space for various levels of contextual summarization.

### 2.3 Design Space for Multi-variate Time-Based Charts

KronoMiner [20] presents a taxonomy for the design space of layouts for visualizing multi-variate, time-based data. It includes stacking (i.e., multiple views are stacked on an axis to create global time alignment) and overplotting (i.e., multiple visual encodings are layered in a single view). Our framework supports both techniques and adds the ability to utilize multiple non-overlapping encodings (one for each zone) in a single stacked track.

## 3 PERIPHERY PLOTS FRAMEWORK

The periphery plots framework provides multi-scale contextualization capabilities for viewing heterogeneous time-based data, and includes several core concepts for how periphery plots connect with

each other and other visualization elements. The framework also defines a two-dimensional design space for periphery plots themselves and a set of common user interactions.

### 3.1 Concepts

Periphery plots are defined by the relationship between temporally linked *focus* and *context zones*, which represent user-specified intervals along a dataset's time axis. These zones are used to determine the intervals of data to be visualized within time-based *tracks*. Each track includes a set of *periphery plots* and a *focus plot* with one plot for each zone. The zones can be manipulated via interactions with the plots on individual tracks or through a global *control timeline*.

**Focus and Context Zones.** Zones are the primary concept that underpins the framework for periphery plots. For a given temporal data visualization, periphery plots define a single focus zone and a set of context zones. Each zone is an interval of time defined by start and end points that are linked together in a non-overlapping, contiguous sequential ordering. Figure 1(b<sub>1</sub>,c,b<sub>2</sub>) shows an example that has two context zones, one to each side of the focus zone. The framework allows multiple context zones on either side of the focus as demonstrated in the accompanying video figure.

The focus zone corresponds to the time period of primary interest. The visualization used for the focus period is designed to show data at high resolution for detailed analysis (e.g., a line chart, a horizon graph, or an event timeline) with the visual design dependent, in part, on the specific data type and application. The focus zone is equivalent to the interval of temporal data displayed in a more typical visualization that does not include periphery plots.

In contrast, context zones are defined relative to the focus zone as either pre- or post-focus intervals of time. Context zones are not generally the user's primary focus of attention. Instead, they represent neighboring periods of time to the focus which provide context to users during interpretation of data found within the focus zone.

While it is common to define two context zones (one pre- and one post-focus), periphery plots allow multiple context zones to enable multi-scale contextualization (e.g. six context zones can be used to contextualize a focus time interval with a week, month, or year before and afterwards). Thus, an arbitrary set of  $n$  context zones are defined relative to a single focus zone.

Each context zone is visualized using periphery plots. As the screen space allocated to a periphery plot is typically more constrained than the focus, data summarization is a key element in designing effective periphery plots. This motivates the periphery plot design space outlined in Section 3.2.

**Tracks.** A track is a container that horizontally aligns a set of visualizations that correspond, typically, to a single variable. As shown in Figure 1(f), each track contains a periphery plot for each context zone and a single focus visualization corresponding to the focus zone. The visualizations within a track are arranged contiguously from left to right, ordered by the sequence of time intervals for the corresponding context and focus zones.

Multiple tracks can be used to visualize multiple variables simultaneously. In this case, all tracks are defined using the same set of context and focus zones. In addition, the tracks are stacked vertically with boundaries between zones aligned as shown in Figures 1 and 3. This facilitates comparison across tracks by lining up data from the same time interval in vertical columns. The visual encodings for each track can be configured independently, supporting the visualization of heterogeneous data types.

**Control timeline.** The specification of context and focus zones is supported via a single control timeline. This mechanism includes a set of multiple-linked brushes that map one-to-one to the individual zones. Any changes to the focus or context zones made through the control timeline are tightly coordinated with the tracks to support a range of interactions as described in Section 3.3.

### 3.2 Periphery Plot Design Space

Each context zone is visualized using one periphery plot per track. As previously described, periphery plots are typically space constrained which makes summarization a key design goal. We characterize the design space for periphery plots in terms of an **Axis Preservation Model** that includes a two-dimensional taxonomy for visual encodings of temporal data. The model categorizes encodings based on whether or not they map the *time domain* and/or *value domain* from data space to the *x* and *y* coordinates of screen space, respectively. This model is used for both continuous and discrete values (e.g., ordinal values or nominal categories).

This model leads to four types of periphery plots as illustrated in Figure 2: time-value-axis preserving (TVAP), time-axis preserving (TAP), value-axis preserving (VAP), and no-axis preserving (NAP).

TVAP plots, such as the bar charts in Figure 1(d<sub>2</sub>) or dot plots in Figure 1(d<sub>3</sub>), present the most detailed information. As data gets dense, aggregation by value or time can lead to more summarized views. For example, VAP plots aggregate over time but preserve values as in the histograms in Figure 1(d<sub>1</sub>,d<sub>4</sub>). TAP plots, in contrast, aggregate by value but preserve time. For example, a vertical band chart that shows the counts of observations per time period without distinguishing between values of observations as depicted symbolically in Figure 2(2). NAP plots adopt designs that preserve neither time nor value, such as grid-based or pixel-based visualizations.

We note that the axis preservation model applies to a variety of time- and value-based transformations. For example, a moving average envelope (e.g., Figure 2(1)) that aggregates values within a sliding time window would be classified as TVAP because it maintains time as a dimension despite the windowing transformation.

**Summarization.** Axis preservation can be viewed as an approach to summarization through relaxed constraints on visual encoding. Preserving both time and value places constraints on visual design that can make it difficult for periphery plots to effectively display the large scale volumes of data that often fall within a context zone. This problem is exacerbated by the limited screen space typically allocated to periphery plots.

Sacrificing an axis allows for more flexibility in visual encoding. This can make visualizations less informative in some ways (the omitted axis), but enables new types of more space-efficient visual representations. This observation imposes an ordering on plot types with increasing levels of summarization: TVAP → (VAP,TAP) → NAP. Transitions between periphery plot types are a natural approach to handling changes in data volume within a context zone, and two summarization paths fit naturally within the axis preservation model as shown by the arrows in Figure 2(a,b). Moreover, such transitions can be automated as discussed in Section 4.2.

**Annotations.** To facilitate comparisons between zones within a single track, our design introduces *annotations*. These are layered graphics that allow for comparisons of derived measures between plots, including plots that implement different visual encodings. Many common types of derived measurements can be compared via annotations, including averages, confidence intervals, and quantiles.

The *temp max* track in Figure 1 demonstrates the utility of average line annotations for identifying differences in values in the focus area and the context zones. As the orange dotted line annotations in Figure 1(e) show, the value within the focus zone is higher than in the subsequent time period. As demonstrated in the accompanying video, further insights can be obtained from dynamic changes in annotations in response to user interactions. This is because those interactions can result in different intervals of data being displayed in both focus and context zones.

### 3.3 Interactions

Periphery plots can be informative even for static visualizations. However, our framework for periphery plots includes a set of user

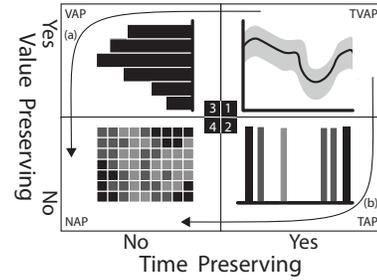


Figure 2: Periphery plot types can be characterized by their preservation of the time and value axes. Summarization can prioritize (a) value preservation or (b) time preservation.

interactions that build upon the core concepts defined in this section to help support interactive data exploration.

**Zooming and Panning.** Users can zoom and pan through the overall time scale to explore data. Users can execute these interactions from the focus zone on any track, using the scroll-wheel to zoom or dragging the mouse to pan. These interactions are not supported in periphery plots, which may have a drastically different time scale (or none at all). Similar zoom and pan mouse gestures can be performed via the control timeline. Regardless of where a pan or zoom originates, all tracks are kept synchronized to maintain temporal alignment of the various focus and context zones.

Additionally, mouse hovers within any focus zone will display a vertical time indicator across all tracks. This helps facilitate navigation of the time axis, identification of specific date and time values, and visual alignment of data in the time axis.

**Multi-Brush Zone Control.** Beyond the aforementioned pan and zoom interactions, all other manipulations of the focus and context zones are performed via the control timeline. A pan or zoom event may cause the extent of a context zone to extend beyond the limits of the time axis. This allows users to move the focus zone to the very start or end of the overall time axis without having to remove or adjust context zones.

Users can change the extent of a single zone by manipulating the zone’s corresponding brush on the control timeline. Each brush includes two *handles* representing the start and end of the corresponding zone. The handles are visible as green squares in Figure 1(b<sub>1</sub>,c,b<sub>2</sub>). Users can drag any handle to resize the corresponding focus or control zone. Other zones will retain their size, but will be “pushed” or “pulled” to maintain the requirement that all zones are contiguous without gaps between them.

The “push” and “pull” behavior is not always desired. For this reason, users can enable *locks* to anchor a zone boundary point in time. The locks, visible in Figure 1(b<sub>1</sub>,c,b<sub>2</sub>) as black squares, can be enabled or disabled by clicking. When a lock is enabled, the handles above it are disabled since the corresponding boundary cannot be modified until the lock is removed. Locks also impact panning behaviors. This allows, for example, a user to lock context zones at the very start and end of the control timeline to create periphery plots that summarize “all data before” and “all data after” the focus zone.

## 4 TOOLKIT AND EXAMPLE USE CASES

Periphery plots were developed in the context of a broader project (Precision VISSTA [2]) focused on patient-generated health data (PGHD). The techniques were then generalized within a reusable toolkit for easy application to support the analysis of other datasets and domains. This section describes the reusable periphery plot toolkit and two example applications.

### 4.1 Toolkit

The core components of the periphery plot framework were first developed in JavaScript as part of the patient-generated health data

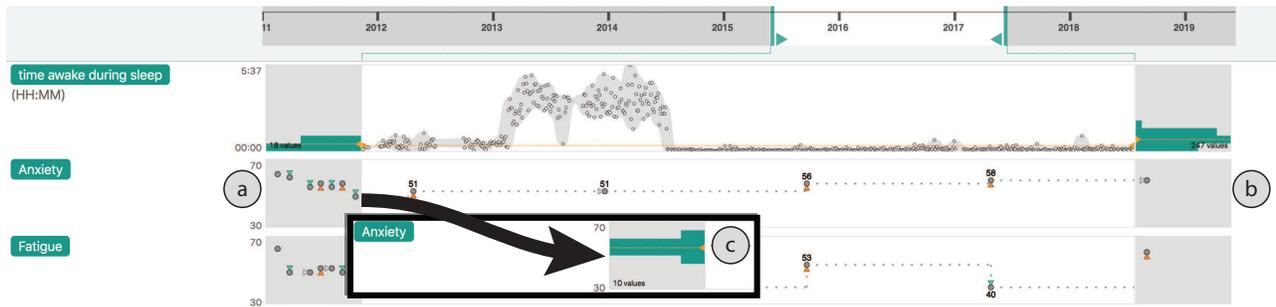


Figure 3: The Precision VISSTA App for data exploration uses periphery plots to help users understand how the values in the current focus area relate to data recorded prior to or after the focus time period. This example shows sparse and discrete values shown in two formats: (a,b) time and value preserving charts that condense the time axis, or (c) value-preserving histogram charts that eliminate the time axis in order to display data about a higher number of observations. Periphery plots can be configured to automatically transition between these modes.

platform described in Section 4.2. That application leverages the React framework and D3.js [5] to implement the control timeline, individual data tracks, and the corresponding periphery plots. To facilitate reuse, we extracted these components and bundled them as a generic periphery plot toolkit. The toolkit provides all core capabilities within a React-based component that can be: (1) easily integrated into React-based web applications; and (2) easily customized and extended with new graphic styles, chart types, annotations, or interactions. The source code has been released as open source software [1].

#### 4.2 Use Case 1: Patient-Generated Health Data

Periphery plots were first developed to support visualizations of PGHD for clinical interpretation [2]. This includes mHealth data gathered from various sensors (e.g., total steps or time awake during sleep) and manually entered logs (e.g., nutrition), as well as patient-reported outcomes data gathered via validated electronic surveys. PGHD contains multiple longitudinal variables with heterogeneous data types. Moreover, these variables are recorded at different rates with varied sparseness, and exhibit different forms of missingness. To make this complex combination of longitudinal records clinically useful, a visualization must allow users to quickly look for long term trends, outliers, patterns, and correlations across variables.

We applied the periphery plot framework to this problem as shown in Figure 3. The vertical alignment across tracks allows comparisons across variables. The most widely used periphery plot chart type was a value-preserving histogram showing the distribution of values in the pre- and post-focus context periods. This helps to show, for example, that the spikes in the *time awake during sleep* track are unusually high compared to values in the context zones (which are set to show all data prior to or after the focus).

When the data in a context region is sufficiently sparse, a value-and-time preserving periphery plot chart type is used. This was motivated by the relatively sparse nature of many of the PGHD variables which are collected manually (rather than via sensors). Figure 3(a) exploits this feature to show not only that earlier anxiety scores were higher, but that they have been decreasing leading up to the focus time period. This feature is very useful, but does not scale well as the number of values in the context increases. For this reason, periphery plots are designed to automatically transition from (a) the value-and-time preserving view to (c) a value-preserving summarization, if a user interaction results in a higher number of samples in the periphery that exceeds a defined threshold.

#### 4.3 Use Case 2: Climate Data

To demonstrate the versatility of periphery plots, we applied the toolkit to a second data set: a publicly available collection of Seattle weather observations [3]. This use case is highlighted in Figure 1.

Compared to the health application above, weather data is more homogeneous (mostly scalars over time) and is more uniformly sampled over time. Moreover, it exhibits less sparsity and missingness. Yet despite these differences, the periphery plot framework can be easily applied to this use case. This application was developed with minimal effort by connecting CSV data with built-in chart types.

The interactive application allows for the discovery of several interesting observations. The screenshot in Figure 1 shows examples, including the seasonal fluctuations in temperatures highlighted in the *temp max* track. Here, the focus and context zones were each set to approximately six months in duration. The annotations in that track, Figure 1(e), show that the focus period is much hotter than the period before and after. Panning the control timeline forward or back shows this pattern oscillate as the focus moves from winter to summer and back again. A second finding is shown in Figure 1(d<sub>4</sub>) and highlights some drizzling precipitation. This is quite different from the current focus, which is more similar to the distribution of observations shown in the post-focus context region in Figure 1(d<sub>8</sub>).

## 5 DISCUSSION AND CONCLUSION

The use cases in Section 4 illustrate the strengths of periphery plots in contextualizing detailed time-based data within a larger time frame. The examples demonstrate, in particular, how the periphery plot framework provides a flexible approach for visual encoding in the context zone. This includes the automated transition between plot types to adjust the level of information preservation based on available space. There are, however, some limitations in our current definition of the framework. For example, the framework does not explicitly cover a scenario in which multiple variables are shown within the same track. Furthermore, the framework does not ensure that periphery plots are easy to compare within or between tracks. In future work, we hope to explore solutions to this challenge by building upon prior research into the design space and best practices for multi-view comparability and consistency [14, 17].

In summary, we demonstrate that the periphery plot framework for providing contextual information for heterogeneous time-based data is generalizable and delivers insights that would not be possible without the context provided by our approach. Despite the relatively limited design space, periphery plots are an effective approach that is easy to implement and use. Furthermore, the approach does not require any complex visual representations or interactions to convey information critical to the interpretation of the data. Therefore, we anticipate that periphery plots have the potential to be effective for a wide range of users across multiple application domains.

## ACKNOWLEDGMENTS

This research was supported by the National Institutes of Health under award number 5R01EB025024. Additional support was provided by the National Science Foundation via Grant #1704018.

## REFERENCES

- [1] Periphery Plots Open Source Software. <https://github.com/PrecisionVISSTA/PeripheryPlots/>. Accessed: 2019-06-10.
- [2] Precision VISSTA: Enabling Precision Medicine through the Development of Quantitative and Visualization Methods. <http://precisionvissta.web.unc.edu/>. Accessed: 2019-06-10.
- [3] Seattle Weather Exploration. [https://vega.github.io/vega-lite/examples/interactive\\_seattle\\_weather.html](https://vega.github.io/vega-lite/examples/interactive_seattle_weather.html). Accessed: 2019-06-09.
- [4] D. Borland, W. Wang, and D. Gotz. Contextual Visualization. *IEEE Computer Graphics and Applications*, 38:17–23, 2018. doi: 10.1109/MCG.2018.2874782
- [5] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-Driven documents. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2301–2309, Dec. 2011.
- [6] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys*, 41(1):1–31, Dec. 2008. doi: 10.1145/1456650.1456652
- [7] E. Cuenca, A. Sallaberry, F. Y. Wang, and P. Poncelet. MultiStream: A Multiresolution Streamgraph Approach to Explore Hierarchical Time Series. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3160–3173, Dec. 2018. doi: 10.1109/TVCG.2018.2796591
- [8] S. Few. Time on the Horizon. *Visual Business Intelligence Newsletter*, p. 7, July 2008.
- [9] S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: visualizing theme changes over time. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, pp. 115–123, Oct. 2000. doi: 10.1109/INFVIS.2000.885098
- [10] J. Heer, N. Kong, and M. 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, CHI '09*, pp. 1303–1312. ACM, New York, NY, USA, 2009. event-place: Boston, MA, USA. doi: 10.1145/1518701.1518897
- [11] W. Javed and N. Elmqvist. Stack zooming for multifocus interaction in skewed-aspect visual spaces. *IEEE transactions on visualization and computer graphics*, 19(8):1362–1374, Aug. 2013. doi: 10.1109/TVCG.2012.323
- [12] W. Javed, B. McDonnel, and N. Elmqvist. Graphical Perception of Multiple Time Series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–934, Nov. 2010. doi: 10.1109/TVCG.2010.162
- [13] R. Kincaid. SignalLens: Focus+Context Applied to Electronic Time Series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):900–907, Nov. 2010. doi: 10.1109/TVCG.2010.193
- [14] S. Knudsen and S. Carpendale. View Relations: An Exploratory Study on Between-View Meta-Visualizations. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction, NordiCHI '16*, pp. 15:1–15:10. ACM, New York, NY, USA, 2016. event-place: Gothenburg, Sweden. doi: 10.1145/2971485.2971566
- [15] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: detail and context smoothly integrated Mackinlay color plates. In *Proceedings of the SIGCHI conference on Human factors in computing systems Reaching through technology - CHI '91*, pp. 173–176. ACM Press, New Orleans, Louisiana, United States, 1991. doi: 10.1145/108844.108870
- [16] P. J. Polack, S.-T. Chen, M. Kahng, K. DE Barbaro, R. Basole, M. Sharmin, and D. H. Chau. Chronodes: Interactive Multifocus Exploration of Event Sequences. *ACM transactions on interactive intelligent systems*, 8(1), Feb. 2018. doi: 10.1145/3152888
- [17] Z. Qu and J. Hullman. Keeping Multiple Views Consistent: Constraints, Validations, and Exceptions in Visualization Authoring. *IEEE transactions on visualization and computer graphics*, 24(1):468–477, 2018. doi: 10.1109/TVCG.2017.2744198
- [18] C. Shi, W. Cui, S. Liu, P. Xu, W. Chen, and H. Qu. RankExplorer: Visualization of Ranking Changes in Large Time Series Data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2669–2678, Dec. 2012. doi: 10.1109/TVCG.2012.253
- [19] M. Tory and T. Moller. Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):72–84, Jan. 2004. doi: 10.1109/TVCG.2004.1260759
- [20] J. Zhao, F. Chevalier, and R. Balakrishnan. KronoMiner: Using Multifoci Navigation for the Visual Exploration of Time-series Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11*, pp. 1737–1746. ACM, New York, NY, USA, 2011. event-place: Vancouver, BC, Canada. doi: 10.1145/1978942.1979195