Eurographics Conference on Visualization (EuroVis) 2017
B. Kozlíková, T. Schreck, and T. Wischgoll (Editors)

Short Paper

Visualizing Large Time-series Data on Very Small Screens

Yang Chen

I4 Data, USA

Abstract

In this paper, we present a space-efficient visualization and an interaction technique for exploring time-series data on very small screens. The visualization is based on a simultaneous display of contextual visualization in the display border and focused interfaces in the center region of displays. The interaction technique utilizes fisheye distortion to facilitate data selection and navigation interactions in the contextual visualization. The proposed techniques could benefit a wide range of analysis applications on wearable devices. Two applications are highlighted to demonstrate the usefulness of the techniques.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [INFORMATION INTERFACES AND PRESENTATION]: User Interfaces—Graphical user interfaces (GUI)

1. Introduction

Wearable devices like smartwatches or smart accessories have emerged as a promising new way to help people gather and monitor time-series data in different application domains. In many of the applications, there is an urgent need to analyze the time-series data in real-time (e.g., stock price prediction) and thereby take a swift response or make decisions. This has promoted people to think about bringing data visualization in ultra-small factors to harness its analytical power anytime and anywhere. In practice, many smartwatch manufacturers have bundled their products with health monitoring or stock market apps that can deliver real-time time-series data and display graphical summaries (e.g., line- or bar-charts) of the data.

The time-series data is readily available in large quantities (e.g., years of stock prices), but the ultra-small interfaces of the wearable devices limit the uses of conventional visualization techniques, even for moderately sized data. By trivially down-scaling the aforementioned simple charts to encompass ultra-small sizes, users either must accept crowded displays that are confusing and cumbersome to understand, or rely on sequences of laborious interactions for zooming and navigating visualizations. The "fat finger" issue [XGF15], exacerbated by the ultra-small display size, makes the selection of data elements slow and laborious. Moreover, effective time-series visualization often requires advanced query and filter capabilities [HS04]. However, integrating augmented query or filter interfaces with the simple charts yields additional display space that is less available for the ultra-small interfaces.

This work facilitates our preliminary efforts in developing highly adaptable time-series visualization techniques while using wearable devices with Very Small Screens (VSSs). For "very small" we look at touch-screens with diagonals around 1.7" (42mm). Our approach to overcoming the limitations of VSSs combines *Border Display*, a space-efficient interface for visualizing and querying time-series data, with *Region Selection*, an algorithmic strategy to interact with the data on VSSs. Inspired by the off-screen visualization techniques [BR03], the border display features a simultaneous display of foci and context of data and enables dynamic queries and filters of the data for clutter-free displays [HS04]. The region selection utilizes a fisheye technique [Fur86] to magnify elements that are of interest to users while keeping their context visible for easy navigation. We present two real-world use cases to highlight the possible applications of the approach.

2. Related Work

Researchers have explored various techniques to visualize complex information spaces on miniaturized computing devices, such as smartphones and tablets. The most popular techniques are perhaps the overview+detail representations [KKS03] that display the information space in two separate views: a downscaled overview of the entire data and a detailed view of a highlighted portion of the overview. The approaches become problematic for VSSs, because the screen space that can be assigned to the overview is typically insufficient to support an efficient navigation and relation to the detail view. In contrast, focus+context techniques [SZ00] simultaneously display magnified foci and their scaled-down context. This permits more flexible layout and navigation methods. Related approaches such as distorted timelines [HDKS07] have been considered effective for visualizing large time-series data. However, none of them has been studied on VSSs. Another type of techniques are off-screen visualizations that convey the presence of off-screen objects on limited display spaces [BR03]. They inspired us the idea of using the border region to display information. Researchers have

DOI: 10.2312/eurovisshort.20171130

also attempted to use alterative visual metaphors [FFM*13] or layouts (e.g., spiral layout [WAM01]) to reconstruct the small factors of visualizations for time-series data. We borrow the concept and combine it with new display and interaction techniques.

Existing techniques for visualizing time-series data fall into three categories: aggregation approaches, lensing techniques, and layout distortions [WBJ16]. The aggregation approaches (e.g., [KL06]) divide data points into segments of time and use visual mappings to depict their underlying statistical features. While the approaches are useful for assessing the big picture of large time-series, using them standalone will result in a loss of detailed information. Lensing techniques use different magnification factors (e.g., [Kin10]) to visually enhance a focus area of time-series, promising an intuitive navigation of time-series. However, the navigation efficiency is largely dependent on successful interactions with touch techniques and is hardly achievable on VSSs due to the fat finger issue. The layout based techniques use advanced spatial arrangement such as matrix displays [MMKN08] and focus+context [HDKS07] to magnify a focus area of a time-series while enabling tracking and navigation of the context. Nonetheless, none of the above techniques has been developed and studied on VSSs.

3. Tasks and Requirements

To summarize important exploratory tasks supported by existing time-series visualizations, we surveyed 12 related technical papers that were published on major visualization conferences between 2006 and 2016. It led to two fundamental tasks that are common to most reviewed application domains:

- Finding patterns: which involves searching a time-series for a specific time-dependent pattern. The patterns could be either known (e.g., search by examples) or unknown (e.g., motif discovery) dependent on different analytical needs.
- Finding trends: which consists of general increasing/decreasing trends and abrupt changes (anomaly detection).

From the tasks, we derived a set of functionality requirements for guiding the design of general time-series visualizations. The requirements, which were based on our assessment of existing systems in the paper survey, include:

R1: Support aggregation tasks. Users often start the tasks by deriving a high-level view, that is, to explore aggregate properties of data [BRS*12]. For example, searching patterns over a long time-series requires users to divide the entire time into intervals where features of the patterns (e.g., mean values) can be located and compared [CAFG12].

R2: Enable multi-resolution displays. Most existing systems for visualizing large time-series provide multi-resolution displays [HDKS07], e.g., a high resolution display of intervals that are relevant to users' foci and an aggregation view for providing context. The multiple views should be simultaneously displayed to facilitate efficient navigation among them.

R3: Enable multi-focus displays. Comparing values in multiple intervals of a time-series is fundamental for all derived tasks. Accordingly, visualization should offer the capabilities to simultaneously display the multiple focused intervals at high details while retaining their context and temporal distance [JE10].

The limited display size is another key factor that has to be considered for designing visualization interfaces and interactions on VSSs. While previous research (e.g., [KKS03] and [Chi06]) has suggested several requirements that can adapt visualization designs for small, phone-size screens, directly leveraging them for VSSs might result in inefficiency since the latter have a much higher size limitation. Hence, our analysis process involved not only a summary of the existing design requirements, but also a refinement process in which we carefully justified the adaptability of the requirements to VSSs. The process led to the following requirements:

R4: Reduce the information complexity. Plotting large time-series data on ultra-small screen spaces can lead to occlusion displays that hinder efficient visual exploration. A simple but intuitive solution is to reduce the amount of displayed information, e.g., only displaying a subset of the data that matches users' interest.

R5: Reduce the display complexity. The limited display capability also yields techniques that can decrease the complexity of graphical presentations. In particular, two major sources of the complexity has been identified, namely the occlusion display and the multi-view display (e.g., displaying two interfaces in adjacent screens). Addressing the former calls for space-efficient visualization layouts. The latter should be replaced by integrated interfaces such that users do not have to mentally connect information on separated screens.

R6: Simplify interactions. Due to the limited input area, touch interactions can be easily prone to the fat finger problem, and thus become laborious and inaccurate on VSSs [XGF15]. Therefore, complex interactions, such as those that require two-finger operations or sequences of swipes, should be avoided and/or replaced by basic and easy to handle touch functions. Special attention should be paid to the data selection because selecting objects of small sizes is challenging both cognitively and physically.

4. Approach

Based on the derived design requirements, we have developed a visualization approach that seamlessly combines two techniques: a Border Visualization for efficiently representing large time-series data on VSSs and a Region Selection that facilitates the data selection in the visualization. They are presented in this section.

4.1. Data Modeling and Analysis

In order to reveal the underlying properties for large time-series (R1), the approach divides an input data into many discrete segments, each of which records values in a specific interval. For example, a fitness heart rate data might consist of small segments that are collected from individual training sessions in a month. If an inherent segmentation of data is not available (e.g., continuous time-series), it can be partitioned based on user-defined time units (e.g., day, week) or underlying features (e.g., change points). For example, a stock price data can be studied at a daily or weekly basis based on particular analytical needs. To extract the aggregate properties from each segment, we implemented the list of feature calculations that are presented in [DW13]. The calculated features are fed into the visualization where users can query them for individual segments or explore their overall trend for the entire data.

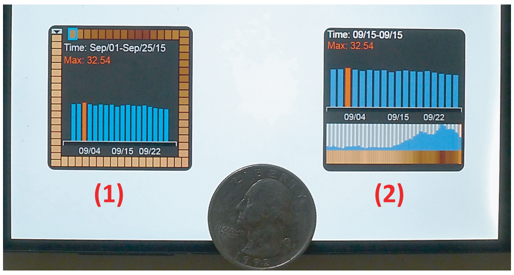

Figure 1: *Simulate the (1) border visualization and the (2) overview+detail bar charts on a smartphone. The same screen resolution (200 × 200 px) and stock price data (69 segments) are used in the two visualizations.*

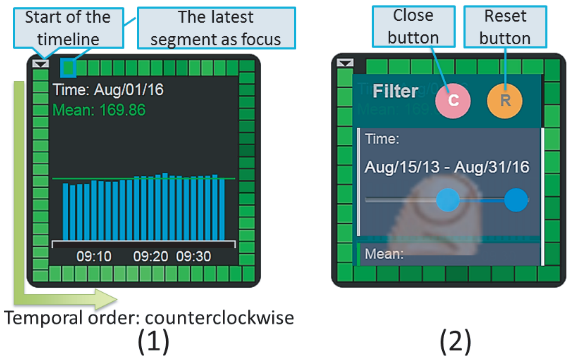

Figure 2: *(1) Visualizing the mean values of a stock price data (65 segments) in the border visualization. The focused bar chart (screen center) displays the latest segment and highlights its mean value; (2) Filtering the segments in the query interface. A time range is specified and the border display shows 37 segments that match the time filter.*

4.2. Border Visualization

As shown in Fig. 1 (2), our initial design was an overview+detail bar visualization that displayed all segments of data (overview) below a detailed view of a focused segment. Features were represented by colored rectangles under the respective segments in the overview. However, the limited display space of the overview (with a width of 200 px) led to cluttered views that tremendously hinder visual exploration and interactions (R5). Moreover, splitting the screen means less display space is available for the detailed visualization, yielding difficulties in comparing multiple segments (R3).

Inspired by the off-screen visualizations [BR03] and space-filling time-series displays [SBM*14], our improved design (Fig. 1 (1)) utilizes dedicated screen border regions to display the overview (denoted as border display) and leaves relatively ample space in the screen center for visualizing foci (R2). A query interface (Fig. 2(2)) can be displayed on-demand over the screen center to enable the filter and query capabilities (R4). By allocating more space for the overview display (to a width of 740 px) and providing the filtering capability, the design significantly increases the visual scalability (R5) and eliminates the needs for zooming and panning interfaces (R6). The capability to simultaneously display the query interface and the overview allows users to dynamically filter and/or query data [Shn94] without switching to separate screens (R5). Compared with the overview+detail design, the design is more easily to generalize to screens of arbitrary shapes (e.g., circular screens) because it only depends on the border region for displaying segments.

Border display as context: As shown in Fig. 2 (1), the segments of time-series data are displayed as a sequence of rectangles and are sorted to retain their temporal order. The sequence is filled into the border region in a counter-clockwise direction beginning from the top-left corner (indicated by the motif). The direction is consistent with how time is ordered in focused bar charts in the center region. Users can customize it based on their preferences. The height of the rectangles (also the width of the border display) was carefully measured (14 px for the 200 × 200 px resolution [CGF14]) to compact the border display while ensure that the segments can be easily tapped via finger touches (R6). Pre-defined color was manually assigned to each feature displayed on the segments. The saturation of the color is used to encode properties of the feature based on particular analytical needs (see the Section 5).

Center display as foci: To visualize a focused segment, we implemented a bar chart for its perception efficiency [FFM*13]. Two focused segments can be displayed in a juxtaposition for easy comparisons (R3) (e.g., Fig. 3(3)). The focused segments are highlighted in the border display using the same bar chart colors, allowing an easy identification of their temporal distance (R3). Two methods are provided to specify and switch focused segments: (1) directly selecting them in the border display, which is much quicker; and (2) swiping a bar chart to display its surrounding neighbor in the border display, which requires less selection accuracy.

Query interface as an overlay: Tapping a bar chart will trigger the query interface that summarizes the details of the respective segment (Fig. 3(1)). In particular, attributes (e.g., time) and features of the tapped segment are displayed as a list of widgets, together with their value ranges for all segments that provide a context. Tapping a feature widget would display the feature in the border display (encoded by segment colors), enabling a quick examination of overall patterns (R2). Users can specify value ranges for features or attributes to filter segments in the border display. For example, in Fig. 2(2), a user uses the collapsible slider to specify the time range as a filter. Then, the border display shows only those segments that match the time filter (comparing with Fig. 2(1) without filtering).

4.3. Region Selection

To facilitate precise selection of segments in the border display (R6), we propose an interaction technique that combines the fisheye distortion technique [Fur86] with a progressive refinement [KBB11]. As shown in Fig. 3(2-3), the technique first magnifies multiple segments of a specific region (denoted as selection region) that requires less precision to select. Then, the targeting segments can be easily selected in the magnified region.

There are multiple ways to specify the selection regions in the

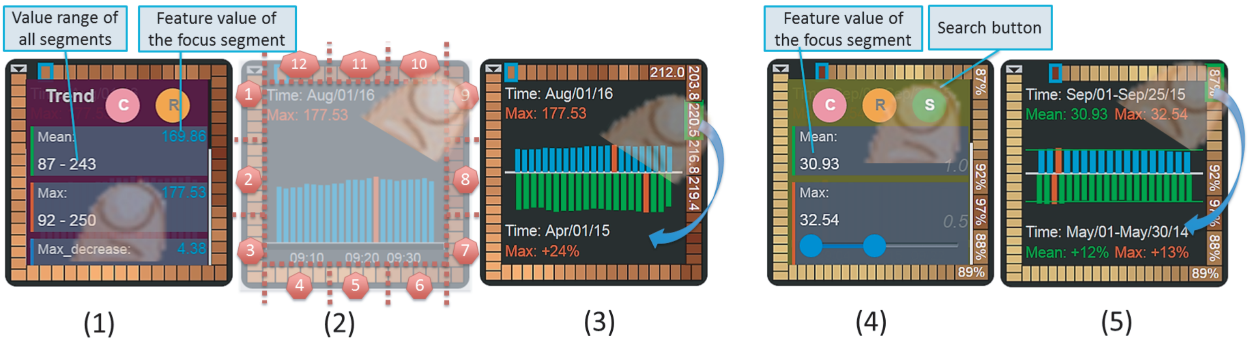

Figure 3: *Two applications of the proposed techniques; (1-3): use Trend Explorer to explore a heart rate data; (1) select the max feature to visualize its values in the border display; (2) trigger a selection region (9) to explore the segments in more details. The grid and numbers of the regions are shown for demonstration purposes; (3) add a magnified segment as a new focus for detailed comparisons. (4-5): use Feature Searcher to explore a stock price data; (4) query segments that have similar features (mean and max, weights 1.0 and 0.5, respectively) with the focused segment; (5) compare the features between a magnified segment and the focused segment.*

border display, e.g., using a bounding box drawn by users. Inspired by the spatially stable layout designs [LGCG16], our approach utilizes a 3×3 grid to generate fixed selection regions on 200×200 px rectangular screens (Fig. 3(2)). The design enables a spatially-stable input space that eliminates the efforts for drawing the bounding box. When users tap an unmagnified segment, the system identifies its region and magnifies all the segments within the region, and meanwhile decreases segment sizes in surrounding regions. Smooth animations are also provided to facilitate the perception of the size changes. Once segments are magnified, they are displayed with additional information (e.g., the feature values shown in Fig. 3(3)). By dragging a magnified segment to the screen center, users can either add a new focused chart or replace an existing focused segment (Fig. 3(3)). While the technique has a limitation exploring very dense segments displays (e.g., > 100 segments), we allow users to dynamically filter the segments to reduce the complexity.

5. Applications

We introduce two applications built upon the proposed techniques to demonstrate their usefulness and efficiency. The applications were deployed on smartphones (e.g., Fig. 1(1)) for the purposes of rapidly prototyping and experimenting with full touch capability.

5.1. Trend Explorer

Trend Explorer is an application for supporting our trend exploration task. As a motivating scenario, an athlete who just finishes a fitness training session uses her smartwatch to examine the recorded heart rate data and compare its peak value against her monthly or yearly trends. The capabilities of quickly exploring and comparing such historical trends allow the athlete to evaluate her performance in time and adjust her training plan in advance.

Fig. 3(1-3) illustrate an exploration of the heart rate data, where the focused bar chart visualizes the record of the latest training session and the border display shows the historical records. By displaying the feature "max value" in the border display (encoded by the segment color), the athlete can observe an overall trend of the peak values (encoded by the saturation of the color). Here, she focuses on some recent training sessions that have higher peak values than the latest record (Fig. 3(3)). Tapping and magnifying the region of these segments allow the athlete to quickly examine their exact peak values. She adds a magnified segment as a focus and compares its detailed values and features with the latest record.

5.2. Feature Searcher

For wearable applications, the task "finding patterns" is particularly useful for monitoring and predicting stock prices: image a stock analyst receiving an alert on her smartwatch about sudden price rises of a stock. She wants to survey the history of the stock in an effort to search similar patterns for predicting the prices. The Feature Searcher is developed to support this task in real-time.

The application is based on a "query-by-example" analysis [DW13]: users select a segment from a data overview to search all similar segments. To enable a flexible search, the similarity of segments is calculated based on a weighted combination of features [DW13]. In Fig. 3(4), for example, the stock analyst searches segments that have similar mean and max values with the current stock prices. The search result is conveyed by the saturation of segments in the border display, where darker colors indicate higher similarities. The top-five most similar segments are magnified with their similarity values displayed, enabling a quick examination of the result. To explore their similarities in more details, the analyst adds the magnified segments to the focus region where their values and features can be efficiently compared with the target.

6. Conclusion

In this work, we have presented a novel approach that pairs a space-efficient visualization with a fisheye-based interaction technique for exploring large time-series data on VSSs. In the future, we will conduct user studies to evaluate the approach with real users and real-world data, e.g., to compare it with the classic overview+detail visualization (Fig. 1 (2)). The approach will be also explored with more use cases and data.

References

[BR03] BAUDISCH P., ROSENHOLTZ R.: Halo: a technique for visualizing off-screen objects. ACM, pp. 481–488. 1, 3

[BRS*12] BERNARD J., RUPPERT T., SCHERER M., KOHLHAMMER J., SCHRECK T.: Content-based layouts for exploratory metadata search in scientific research data. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries* (2012), ACM, pp. 139–148. 2

[CAFG12] CORRELL M., ALBERS D., FRANCONERI S., GLEICHER M.: Comparing averages in time series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 1095–1104. 2

[CGF14] CHEN X., GROSSMAN T., FITZMAURICE G.: Swipeboard: a text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (2014), ACM, pp. 615–620. 3

[Chi06] CHITTARO L.: Visualizing information on mobile devices. *Computer 39*, 3 (2006), 40–45. 2

[DW13] DANG T. N., WILKINSON L.: Timeexplorer: Similarity search time series by their signatures. In *International Symposium on Visual Computing* (2013), Springer, pp. 280–289. 2, 4

[FFM*13] FUCHS J., FISCHER F., MANSMANN F., BERTINI E., ISENBERG P.: Evaluation of alternative glyph designs for time series data in a small multiple setting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), ACM, pp. 3237–3246. 2, 3

[Fur86] FURNAS G. W.: Generalized fisheye views. ACM, pp. 16–23. 1, 3

[HDKS07] HAO M. C., DAYAL U., KEIM D. A., SCHRECK T.: Multi-resolution techniques for visual exploration of large time-series data. In *EUROVIS 2007* (2007), pp. 27–34. 1, 2

[HS04] HOCHHEISER H., SHNEIDERMAN B.: Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization 3*, 1 (2004), 1–18. 1

[JE10] JAVED W., ELMQVIST N.: Stack zooming for multi-focus interaction in time-series data visualization. In *Visualization Symposium (PacificVis), 2010 IEEE Pacific* (2010), IEEE, pp. 33–40. 2

[KBB11] KOPPER R., BACIM F., BOWMAN D. A.: Rapid and accurate 3d selection by progressive refinement. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on* (2011), IEEE, pp. 67–74. 3

[Kin10] KINCAID R.: Signallens: Focus+ context applied to electronic time series. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 900–907. 2

[KKS03] KARSTENS B., KREUSELER M., SCHUMANN H.: Visualization of complex structures on mobile handhelds. Citeseer. 1, 2

[KL06] KINCAID R., LAM H.: Line graph explorer: scalable display of line graphs using focus+ context. In *Proceedings of the working conference on Advanced visual interfaces* (2006), ACM, pp. 404–411. 2

[LGCG16] LAFRENIERE B., GUTWIN C., COCKBURN A., GROSSMAN T.: Faster command selection on touchscreen watches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), ACM, pp. 4663–4674. 4

[MMKN08] MCLACHLAN P., MUNZNER T., KOUTSOFIOS E., NORTH S.: Liverac: interactive visual exploration of system management time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), ACM, pp. 1483–1492. 2

[SBM*14] STEIGER M., BERNARD J., MITTELSTÄDT S., LÜCKE-TIEKE H., KEIM D., MAY T., KOHLHAMMER J.: Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 401–410. 3

[Shn94] SHNEIDERMAN B.: Dynamic queries for visual information seeking. *IEEE Software 11*, 6 (1994), 70–77. 3

[SZ00] STASKO J., ZHANG E.: Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on* (2000), IEEE, pp. 57–65. 1

[WAM01] WEBER M., ALEXA M., MÜLLER W.: Visualizing time-series on spirals. In *Infovis* (2001), vol. 1, pp. 7–14. 2

[WBJ16] WALKER J., BORGO R., JONES M. W.: Timenotes: a study on effective chart visualization and interaction techniques for time-series data. *IEEE transactions on visualization and computer graphics 22*, 1 (2016), 549–558. 2

[XGF15] XIA H., GROSSMAN T., FITZMAURICE G.: Nanostylus: Enhancing input on ultra-small displays with a finger-mounted stylus. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (2015), ACM, pp. 447–456. 1, 2