EuroVis Workshop on Visual Analytics (2019)
C. Turkay and T. von Landesberger (Editors)

Visualizing Event Sequences as Oscillating Streams

Chris Weaver[1] and Ronak Etemadpour[2]

[1]University of Oklahoma, Computer Science, USA. and [2]The City College of New York, CUNY, Computer Science, USA

Abstract

In this paper, we introduce a new method to visually represent sequence structure in data. Like other methods for visualizing temporal or ordinal data, the representation directly maps absolute time or relative ordering of events from left to right horizontally. Unlike other methods, it also accumulates subsequences of events into streams that oscillate up and down vertically. By interactively adjusting the number of steps between vertical reversals, one can rapidly switch perspectives to show variation in event densities over time (one step), overall patterns of event accumulation (all steps), or short-range patterns of event accumulation (in between). In between, the representation reverses stream direction every N steps, accentuating variations in event accumulation while at the same time preserving visual continuity. We present a user study that compares the stream representation to Dotplots. The study validates the readability of the representation for effective visualization of sequence information in text data. It also shows how pairing stream and Dotplot views outperforms both of them individually for some analysis tasks.

1. Introduction

A variety of time-centric visualization techniques exist to help people explore and analyze the dynamics and evolution of systems by looking at time as points, intervals, or cycles. In most techniques, visual representation of time focuses primarily on the *quantitative* characteristics of measurements made with clocks and calendars. Only a few focus on the *ordinal* character of time, in which the primary concern is seeing the ordering of events. Many natural, social, and built systems exhibit complex event dynamics. The exploration and analysis of such systems often involves identifying, characterizing, describing, and explaining patterns in sequences of events.

In this paper, we describe a new visual representation for exploring and analyzing patterns in ordinal data. Unlike most prior visualizations of temporal or ordinal data, the representation uses vertical space to show accumulation of events monotonically, in addition to the usual mapping of events from left to right horizontally. Successive events take vertical steps in a sequence that oscillates up and down, preserving visual continuity. This combination of vertical oscillation and horizontal flow spreads out variations in event accumulation in 2-D, revealing patterns of event occurrences within a stream—or between multiple interwoven substreams, if events are partitioned into sub-sequences by category. The number of steps between vertical reversals can be adjusted interactively, letting one view streams as event accumulations, oscillations, or densities.

In this paper, we describe one design of the visual representation that combines arc diagrams [Wat02] with simple accumulation plots made up of cubic curve segments. We use an example with text from a historic speech to illustrate how sequence information can be effectively visualized using the new representation (Figure 1). To evaluate the design, we conducted a controlled user study to compare it to a well-known representation of sequence data, the Dotplot [CH93]. For the Dotplot we used a common variation consisting of parallel rows, each showing one category/substream of events from left to right. Although Dotplots have lower visual load, we found that oscillations reveal essential information about event patterns, resulting in higher performance on some analytic tasks.

2. Related work

Methods of visually representing time and events build on knowledge of how people perceive and reason about temporal structures and relationships [Fre92, All83, ZT01]. The particular importance of instants and intervals means that visualizations of ordinal data should clearly depict both individual items and their *relative* locations in an ordering. Although oscillating streams are visually continuous, their cubic curve endpoints mark individual item locations within the overall ordering, which is preserved horizontally.

The topological relationships of events and intervals are well-studied in visualization [Tom06, VJC09, DK10, AMST11], both for concrete data items, such as words [CH93], and for more abstract data items, such as co-occurrences of concepts in text [ASW12], and temporal summaries of patient events [WPS*09]. Horizon graphs nest time within time, focusing on temporal variation in meteorological data such as temperatures and wind directions [SMY*05]. Numerous visualizations of time with category attributes involve filtering and aggregation [ZCPB11, ADG11, DWS*12], links between views [ZCCB13], queries to drill down into orderings [GS14], stacking of event streams [HHWN02, Wat05], summarization of words into categories represented in text corpora [SWL*10], grouping of related events that are vertically merged/split [CLT*11, CLWW14], or incrementally organiz-

DOI: 10.2312/eurova.20191127

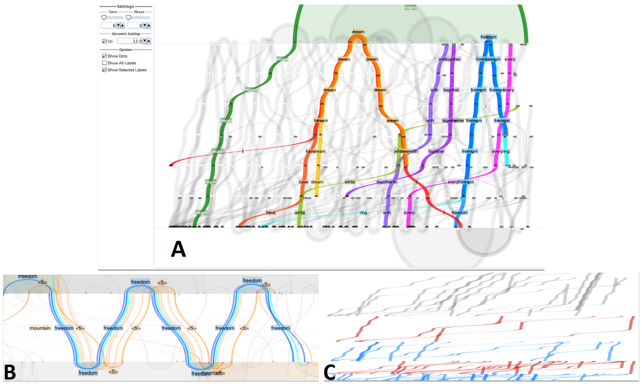

Figure 1: ***(A)*** *Oscillations (N=8) showing word recurrences in Martin Luther King, Jr.'s "I Have a Dream." Words related to race, inclusiveness, and the phrases "I have a dream" and "let freedom ring" are highlighted in (thematically intentional) rainbow colors. Several words have been phase-shifted to reveal the regular cadence of recurrences of the two phrases. Additional shifts reveal the parallel use of inclusiveness words. Whereas "negro" is used almost exclusively at the beginning, "white" appears in seeming transitions between three major phases of the speech, and "together" and "freedom" appear prominently toward the end, reflecting parallels between the speech's progressive theme and its construction.* ***(B)*** *Density (N=1) shows rhythmic repetition of the phrase "let freedom ring." Orange streams show how several common words ("of", "from") parallel the phrase in multiple sentences in a row.* ***(C)*** *Accumulation (N=24) with phase-shifting shows layers of non-stop words that occur many times in the speech. The presence of so many steep streams shows how localized repetition of words is a prominent characteristic of his oration style. Repetition also clusters strongly at the speech's start, middle, and end.*

ing/refining/compressing a layout [THM15]. Multiple views are often used to visualize time [WFR*07, ZJGK10, KBK11]. The oscillating stream representation is most similar in visual form and purpose to the graphical timetables discussed by Tufte [Tuf90, Tuf97].

3. Visual representation

The visual representation focuses on the *ordering* of items in a data set. It represents primarily the relative positions of items in sequence. If given discrete time data, it does position events precisely along the horizontal, but generally de-emphasizes the quantitative (metric) character of individual event times. In the vertical dimension, the representation can be seen as a descendant of Wattenberg's Arc Diagrams [Wat02]. An arc diagram connects two related points along a line using a half-circle. A horizontal sequence of points can be linked together by drawing a series of circles connecting successive pairs. The half-circles in arc diagrams can be drawn upward, downward, or both. For instance, ThreadArcs [Ker03] show messages in an email thread in time order, alternating between upward and downward arcs to more clearly show how replies propagate.

In oscillating streams, curves are repeated in series in the same way as arcs; see Figure 2. When a set of curves is copied and shifted to the top of the original set, flows naturally emerge as a visual side effect. Flowing is a consequence of the vertical continuity that results when the ending point of one curve has the same horizontal position in an ordering as the starting point of a later curve (Figure 2A). This quality can be exploited to visualize sequences in data by chaining together the items in each sequence using a curve for each successive pair.

To reverse the oscillation at the top and bottom of the view, arcs provide a convenient and elegant way to avoid abrupt reversals of direction in the visual flow (Fig 2B). Upward arcs turn upward-flowing streams downward. Downward arcs turn downward-flowing streams upward. The result is a layout in which a pair of arc diagrams sandwich multiple tiers of streams that "flow" upward and downward. Each stream's succession of segments provides a richly detailed shape that supports identification, tracing, and comparison of sequences and the details of their internal structure at different scales.

The number of steps N between reversals affects the overall appearance of flows. With one tier (N=1), streams oscillate up and down to create a waveform appearance (e.g., Figure 1B). With enough tiers to draw all streams without reversal, streams take on the monotonically increasing appearance of accumulation plots (e.g., Figure 1C). In between, streams flow back and forth for a number of cycles determined by their lengths. The number of steps can be interactively adjusted from 1 to the maximum stream length. Individual streams can also be phase-shifted to start at a step from 0 to the current number of steps. These interactions are used to align streams and reduce overlap, allowing one to more readily identify and compare streams at particular points in the overall ordering.

The visual representation is designed to be a general-purpose technique for visualizing ordinal information from virtually any data source. Consider the orderings of related objects within some larger collection of objects. As a data structure, the set of sequences in that collection is represented as a list of $(identity, ordinal)$ pairs. Each item in the list positions an object in some ordered set of things that share a common identity: ordinals in the list are global and can be sparse, objects that are in multiple sequences can take on multiple identities, and a given ordinal can appear with the same identity multiple times to represent multiple objects that have the same position in a sequence. These characteristics allow represen-

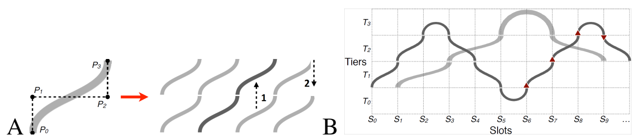

Figure 2: *Stream layout.* ***(A)*** *Curve segments have the same starting point P_0 as arcs, but displace the ending point P_3 vertically. Control points P_1 and P_2, displaced halfway vertically, define a cubic Bézier curve.* ***(B)*** *Grid layout of flowing streams. For each sequence, curve segments flow up and down tiers, capped by arcs in the topmost and bottommost tiers. Sequences with irregular separation of items (gray) contain elongated segments and larger arcs. Marks (red arrows) can be overlaid at points connecting curves and arcs to indicate the start, end, and flow direction of a stream.*

tation of discrete 1-D quantitative data as well as ordinal data that lacks any underlying quantitative interpretation.

More concretely, the input is simply a table of string identifiers and integer positions. We refer to these data values as sequence *keys* and *slots*. A *succession* algorithm maps the two-column input table into a four-column output table that contains the information needed to visually encode segments in the oscillating layout. This algorithm captures information about the piecewise relationships between successive pairs of slots in a sequence defined by a key. The *head* is the start slot, the *tail* is the end slot, and the *rank* is the relative position of the item in the sequence identified by *key*. From left to right, segment position increases strictly monotonically. Horizontal position is calculated by normalizing head and tail values over all segments: $x_{head} = (head - rank_{min})/(rank_{max} - rank_{min})$ and $x_{tail} = (tail - rank_{min})/(rank_{max} - rank_{min})$.

From bottom to top to bottom, segment position increases monotonically *in cyclic fashion*. Vertical position is calculated using integer modulus arithmetic: $tier = (rank + shift + phase)\%(2k+2)$, in which k is the number of non-arc steps in the stack, ranging from one to the length of the longest sequence minus one. Each band is allocated a fixed height large enough to reasonably depict the launch curvature of even very large arcs under most circumstances. The curve steps are allocated about four times that height in total, regardless of the number of steps. This ratio of 1:4:1 for the arcs-curves-arcs "sandwich" appears to strikes a good balance between level of detail and amount of screen space for most of the data sets we have tried it on, including the one used in the user study.

4. User study

The central role of ordering and sequence in text makes it a prime candidate for application of the oscillating representation. Martin Luther King, Jr.'s 1963 "I Have a Dream" is an extraordinary example of inspiring, effective public rhetoric. We performed simple tokenization of a transcription of the speech [mlk] to create a data set for visualization, using token as sequence *key* and token number as sequence *slot*. Figure 1 shows an example of using the visualization to analyze how sequencing of words and phrases provide rhetorical structure and help convey meaning. Figure 1B shows an example of changing the number of steps to take a closer look at ordering relationships between substreams for the words in the phrase "let freedom ring." Several substreams appear as phase-shifted waveforms that indicate verbal parallelism at the sentence level as well. Figure 1C shows how streams of words can be visualized as accumulations using a large number of steps, in this case also grouping related sets of words into distinct layers using phase-shifting.

A controlled user study validates the effectiveness of our approach considering different design elements compared to Dotplots [CH93]. The study was conducted over several days with 30 participants (12 females and 18 males) who were undergraduate or graduate students with little to no prior experience with visualization tools. Including a 10-minute training period and set-up, it took approximately 50 minutes for each individual to complete assigned tasks. Participants were divided into three groups. Each group was assigned to one of three visualization configurations. Individuals performed the same tasks on their assigned configuration. The Flow group **(F)** performed tasks using only the stream visual representation. The Dotplot group **(D)** performed tasks using only our Dotplot variant. The Flow-Dot group **(FD)** performed the tasks using both visual representations together. (See Figure 3.)

The stream visualization allows the number of steps to be interactively adjusted. We turned off this feature to keep the number of steps constant for all participants; for each task, we set up the visualization to use a pre-determined number of steps. (We determined the number for each task in a pretest process with three participants. We asked them to adjust the number of steps to identify sequences, count repeated phrases, and interpret ordering correctly.) Streams were labeled with their respective words. We highlighted individual sequences and kept the colors the same for a given task for all participants. Sequences in the Dotplot were also labeled and used the same color coding for dots, to provide a consistent color scheme for the same words across all three participant groups.

To define representative user tasks (Fig 3), we identified major questions raised when visually analyzing speech text, each associated with our user study goals. We identified three main tasks: (1) counting, to identify the number of occurrences of a repeating event such as a word or sequence; (2) relation-seeking, to identify a phrase or sequence of a particular structure; and (3) ordering, to characterize the relative ordering of phrases or words. <Since The sentences and sequences are different in complexities, each of the tasks were divided into complicated and simple. The harder a phrase is to find, the more complicated the task is. For example, finding the phrase "let freedom ring", repeated 10 times in the text, is much easier to find than "lives on a lonely island", used only once in the text. Some phrases are also inherently easier to distinguish than others. The tasks and their level of difficulties were assessed through a question and answer session with three participants, and put into groups prior to the actual test.>

Task targets were to find phrases or counting of word/phrase occurrences, while task constraints were to consider the whole data set or a subset. On the premise that the oscillating representation is designed specifically to help people find and study complex ordering patterns, the target was identification of information about sequences, then comparison of them to discern the broader structure of data. This order is critical for correct interpretation of the data structure and to understand temporal flow relationships [GS14].

4.1. Quantitative evaluation

Given ground truth, we computed the errors on given answers for each task. For tasks that required the individual to estimate a number (i.e., number of repeated occurrences), the error is computed as $e = \frac{|n_{true} - n_{answer}|}{n_{true}}$, in which n_{true} is the estimated ground truth and n_{answer} is the reported answer. For the ranking tasks (i.e., identification of sequences), we estimated the number of swaps required to get from the reported answer to the ground truth. The error is computed by the number of necessary swaps for the reported answer divided by the number of necessary swaps for the worst answer. For example, if (s_1, s_2, s_3) is the correct ranking and (s_3, s_1, s_2) the reported answer, one needs to first swap s_3 with s_1 and then with s_2 to get from the reported answer to the correct one. Hence, the number of swaps is two. For the given example, the number of swaps for

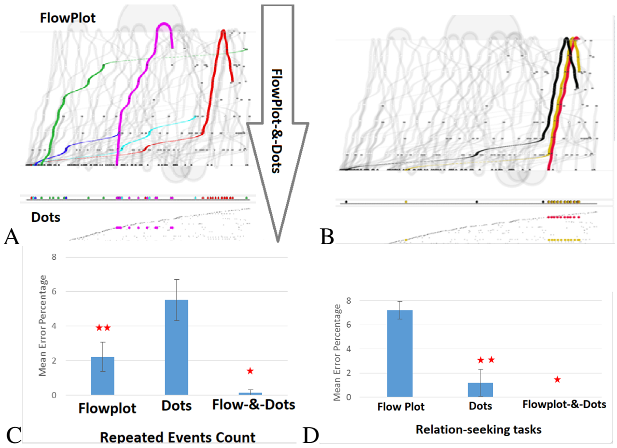

Figure 3: *(A) Counting task: "freedom" red, 13; "negro" green, 8; "American" blue, 3; "nation" teal, 4; "dream" pink, 11. (B) Phrase identification task: "let freedom ring" with "let" Red → "freedom" black → "ring" yellow. "Freedom" and "ring" are flowing together all along the text. (C) Counting task results. There are pairwise significant differences. The winner is the **FD** configuration (single asterisk). Second is the F configuration (two asterisks), with significant difference compared to the **D** configuration. (D) Relation-seeking task results. There is a significant difference between **FD** and **D**. Although the error for **F** is higher than **D**, the statistical test showed the error for **FD** is much less than **D**. (Both bar charts show mean error and standard error from the mean.)*

the worst answer (s_3, s_2, s_1) would be 3 and the error would be $\frac{2}{3}$. We first tested the distribution of the error values against normality using a Shapiro-Wilk test. In case of non-normal distribution, we applied an independent non-parametric Kruskal-Wallis test and the Mann-Whitney U tests for post-hoc analyses. We highlight some of the important results below.

Figure 3C summarizes the comparative analysis of the counting (identification of repeated occurrences) task for the three visualization configurations. A post-hoc Mann Whitney U test showed that the error in the **F** group was statistically higher than in the **FD** group ($U = 1072.000, p = 0.013$). In addition, the error in the **D** group was significantly higher than in the **F** group ($U = 972.500, p = 0.014$). Therefore, the oscillating representation performs better than Dotplots on the task to identify repeat occurrences.

Similarly, Figure 3D summarizes the comparative analysis of the relation-seeking task for the three visualization configurations. There was a statistically significant difference in the errors calculated from the ground truth among different layouts ($\chi^2(2) = 23.086897, p = 0.000010$), with a mean rank of 57.52 for the **F** group, 40.98 for the **D** group, and 38.00 for the **FD** group. According to the Mann-Whitney U test, the error in the **F** group was significantly higher than in the **FD** group ($U = 255.000, p = 0.000062$). However, the test reveals a statistically significant difference between the **D** group and the **FD** group ($U = 255.000, p = 0.001$). The statistical comparisons in terms of complexities for two groups, simple versus complicated showed, for more complicated tasks a significant difference among mean errors calculated from the ground truth in different layouts ($\chi^2(2) = 17.623535, p = 0.000149$) with a mean rank of 23.15 for the **F** group, 12.35 for the **D** group, and 11.00 for the **FD** group. The post-hoc analysis showed that the error in the **FD** group was statistically significantly less than in the **F** group ($U = 10.000, p = 0.001$). Considering whole tasks (count, relation-seeking, and ordering), there is also statistically significant higher error in the **D** group compared to the **FD** group (U = 3142.500, p = 0.000005). However, looking at errors, the Mann Whitney test reveal no significant difference between groups **D** and **F** (U = 3972.500, p = 0.768703). Taken together, these results reveal that the individuals who fulfilled the tasks in **FD** group using both layouts were performing statistically significantly better when individuals performed only on the oscillating representation, and the complexity of tasks didn't influence the performance of the **FD** group. Therefore, including the oscillating representation in a Dotplots configuration can help to increase task performance.

4.2. Qualitative observations

We observed that determining an optimum number of steps for the oscillating representation was not an easy task. For the real test, the average of steps was calculated through a pretest, and preset before each task via a hidden interface seen only by the examiners. In practical use, adjustment of the number of steps is a frequent part of task performance. Sparsely occurring words were especially hard to discover because of occlusion effects. In particular, words that only occur once are invisible if marks are not turned on. It was also hard to track words across long horizontal spans, likely due to reduced visual curvature in curve segments, even with labels on (floating in the middle of long, squat curves). Turning on marks to help reveal the head and tail positions of segments would also be useful in those cases. In cases of many occurrences of a word across a short horizontal span, the labels can interfere due to overlap.

Nevertheless, according to the participants in the **FD** group, the oscillating representation helped them build understanding of higher levels of text structure, such as frequencies of word occurrence, structure which was not as obvious in the Dotplot. This may be why task error for this group was statistically lower than for the other groups. Most participants could correctly identify the occurrence of a phrase once the first word of the phrase was identified. Some participants devised alternative means, such as a physical vertical ruler laid on the screen, to help perform the counting task. This helped them reach more accurate counts more quickly, suggesting the need for a grid overlay or similar feature as a virtual counting aid. Overall, the oscillation representation had a longer learning curve, but as participants became increasingly familiar with the layout, they were able to perform tasks more accurately.

5. Conclusion

Oscillating accumulations effectively utilize the vertical dimension to visualize event sequences in ordinal data. An increased capability to examine sequences from different perspectives can facilitate identification and characterization of ordinal phenomena, which are essential steps in bridging the foraging and sensemaking processes of visual data analysis. We believe that our technique can be generalized and be used for other application areas including historical records and sports schedules that will be examined in the future.

References

[ADG11] ALBERS D., DEWEY C., GLEICHER M.: Sequence surveyor: Leveraging overview for scalable genomic alignment visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (December 2011), 2392–2401. 1

[All83] ALLEN J. F.: Maintaining knowledge about temporal intervals. *Communications of the ACM 26*, 11 (November 1983), 832–843. doi:10.1145/182.358434. 1

[AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of Time-Oriented Data*. Springer-Verlag, London, 2011. 1

[ASW12] ANGUS D., SMITH A., WILES J.: Conceptual recurrence plots: Revealing patterns in human discourse. *IEEE Transactions on Visualization and Computer Graphics 18*, 6 (June 2012), 988–997. doi:10.1109/TVCG.2011.100. 1

[CH93] CHURCH K. W., HELFMAN J. I.: Dotplot: a program for exploring self-similarity in millions of lines of text and code. *Journal of Computational and Graphical Statistics 2*, 2 (June 1993), 153–174. 1, 3

[CLT*11] CUI W., LIU S., TAN L., SHI C., SONG Y., GAO Z. J., TONG X., QU H.: TextFlow: Towards better understanding of evolving topics in text. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (December 2011), 2412–2421. 1

[CLWW14] CUI W., LIU S., WU Z., WEI H.: How hierarchical topics evolve in large text corpora. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (December 2014), 2281–2290. doi:10.1109/TVCG.2014.2346433. 1

[DK10] DASGUPTA A., KOSARA R.: Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of Visualization / Information Visualization 2010) 16*, 6 (November-December 2010), 1017–1026. 1

[DWS*12] DOU W., WANG X., SKAU D., RIBARSKY W., ZHOU M. X.: Leadline: Interactive visual analysis of text data through event identification and exploration. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)* (Seattle, WA, October 2012), Ward M., Santucci G., (Eds.), pp. 93–102. doi:10.1109/VAST.2012.6400485. 1

[Fre92] FREKSA C.: Temporal reasoning based on semi-intervals. *Artificial Intelligence 54* (1992), 199–227. 1

[GS14] GOTZ D., STAVROPOULOS H.: DecisionFlow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (December 2014), 1783–1792. doi:10.1109/TVCG.2014.2346682. 1, 3

[HHWN02] HAVRE S., HETZLER E., WHITNEY P., NOWELL L.: ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (January-March 2002), 9–20. doi:http://doi.ieeecomputersociety.org/10.1109/2945.981848. 1

[KBK11] KRSTAJIĆ M., BERTINI E., KEIM D. A.: CloudLines: Compact display of event episodes in multiple time-series. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (December 2011), 2432–2439. 2

[Ker03] KERR B.: THREAD ARCS: An email thread visualization. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)* (Seattle, WA, October 2003), IEEE, pp. 211–218. 2

[mlk] [online]URL: http://www.americanrhetoric.com/speeches/mlkihaveadream.htm. 3

[SMY*05] SAITO T., MIYAMURA H. N., YAMAMOTO M., SAITO H., HOSHIYA Y., KASEDA T.: Two-tone pseudo coloring: Compact visualization for one-dimensional data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)* (Minneapolis, MN, October 2005), pp. 173–180. 1

[SWL*10] SHI L., WEI F., LIU S., TAN L., LIAN X., ZHOU M. X.: Understanding text corpora with multiple facets. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)* (Salt Lake City, UT, October 2010), IEEE, pp. 99–106. 1

[THM15] TANAHASHI Y., HSUEH C.-H., MA K.-L.: An efficient framework for generating storyline visualizations from streaming data. *IEEE Transactions on Visualization and Computer Graphics (preprint)*, 99 (2015), 1–1. doi:10.1109/TVCG.2015.2392771. 2

[Tom06] TOMINSKI C.: *Event-Based Visualization for User-Centered Visual Analysis*. PhD thesis, Universität Rostock, June 2006. 1

[Tuf90] TUFTE E.: *Envisioning Information*. Graphics Press, Cheshire, CT, 1990. 2

[Tuf97] TUFTE E.: *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, Cheshire, CT, 1997. 2

[VJC09] VROTSOU K., JOHANSSON J., COOPER M.: ActiviTree: Interactive visual exploration of sequences in event-based data using graph similarity. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (November-December 2009), 945–952. 1

[Wat02] WATTENBERG M.: Arc diagrams: Visualizing structure in strings. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)* (Boston, MA, October 2002), IEEE, pp. 110–116. 1, 2

[Wat05] WATTENBERG M.: Baby names, visualization, and social data analysis. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)* (Minneapolis, MN, October 2005), IEEE, pp. 1–7. 1

[WFR*07] WEAVER C., FYFE D., ROBINSON A., HOLDSWORTH D. W., PEUQUET D. J., MACEACHREN A. M.: Visual exploration and analysis of historic hotel visits. *Information Visualization 6*, 1 (February 2007), 89–103. doi:10.1057/palgrave.ivs.9500145. 2

[WPS*09] WANG T. D., PLAISANT C., SHNEIDERMAN B., SPRING N., ROSEMAN D., MARCHAND G., MUKHERJEE V., SMITH M.: Temporal summaries: Supporting temporal categorical searching, aggregation and comparison. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (November-December 2009), 1049–1056. 1

[ZCCB13] ZHAO J., COLLINS C., CHEVALIER F., BALAKRISHNAN R.: Interactive exploration of implicit and explicit relations in faceted datasets. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (December 2013), 2080–2089. doi:10.1109/TVCG.2013.167. 1

[ZCPB11] ZHAO J., CHEVALIER F., PIETRIGA E., BALAKRISHNAN R.: Exploratory analysis of time-series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (December 2011), 2422–2431. 1

[ZJGK10] ZIEGLER H., JENNY M., GRUSE T., KEIM D. A.: Visual market sector analysis for financial time series data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)* (Salt Lake City, UT, October 2010), IEEE, pp. 83–90. 2

[ZT01] ZACKS J. M., TVERSKY B.: Event structure in perception and conception. *Psychological Bulletin 127*, 1 (2001), 3–21. 1