First publ. in: IEEE Transactions on Visualization and Computer Graphics ; 17 (2011), 12. - pp. 2432-2439

http://dx.doi.org/10.1109/TVCG.2011.179

CloudLines: Compact Display of Event Episodes in Multiple Time-Series

Miloš Krstajić, Enrico Bertini, and Daniel A. Keim

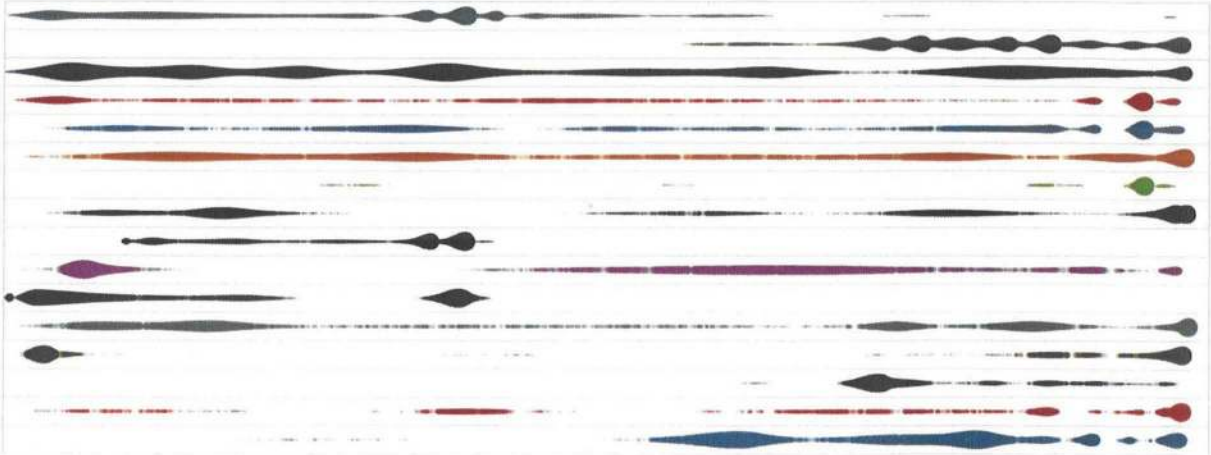

Fig. 1. CloudLines visualization of key politicians appearing in the news during February 2011. The technique allows detection of visual clusters in a compressed view of multiple time series, enhanced with distortion interaction techniques for individual data item analysis.

Abstract—We propose incremental time-series visualization technique with interactive distortion as a way to deal with time-based representations of large and dynamic event data sets in limited space. Modern data analysis challenges in the domains of news publishing, network security and financial services require scalable solutions that will help the users to analyze the event data on atomic level while retaining the temporal context. The incremental nature of the data implies that visualizations have to necessarily change their content and still provide comprehensible representations. In this paper, we deal with the need to keep an eye on recent events together with providing a context on the past and making relevant patterns accessible at any scale. Our method adapts to the incoming data by using a decay function to let the items fade away according to their relevance. Since access to details is also important, we also provide a magnifying lens technique which takes into account the distortions introduced by the logarithmic time scale to enhance readability in selected areas of interest. We demonstrate the validity of our techniques by applying them on incremental data coming from online news streams in different time frames.

Index Terms—Incremental Visualization, Event-based Data, Lens Distortion.

1 Introduction

Online news sources publish news that report on real-world events of different importance. The data is generated at non-equidistant time intervals, making it different from other time series data, such as any type of measurement data, which is sampled at fixed time intervals. Each data item in news streams carries valuable information in both its timestamp and its content. On the other hand, when these *virtual events* come from different sources, but are related to the same real-world event, they create sequences (*event episodes*) that differ in importance based on their volume and time density. This dual nature of such event-based data creates significant challenges for its analysis.

In the last years a lot of research has been conducted on time series representations. One of the lines in research is how to deal with recent data, which has more importance than the data that was acquired before, and how to put the incoming data in the context of the past. Also, in many applications, there is a need to have direct access to individual data items in multiple time series. However, time series datasets are often very long, and the amount of data makes analysis of individual data items very difficult. Usually, data is either aggregated or sampled to reduce overplotting, thus leading to the loss of information. Besides using automated algorithms for analysis, such as pattern detection or measuring similarity, many tasks could be more efficiently solved by using interactive visualization tools that would include the analyst at different stages of the exploration process.

Another important aspect of today's time series data is that it usually comes in dynamic information streams, which puts additional constraints for visual representation of such data. The visualization has to be incrementally updated, without distracting the user and his understanding of the data that he is working with. In many applications, such as the analysis of news, network traffic or financial data, concurrent analysis of multiple series is needed, which allow the analyst to make sense about ongoing relationships between the different categories of time series data. Limited screen space makes this type of analysis difficult. While some research has been conducted on visual

- *Miloš Krstajić is with the University of Konstanz, Germany. E-mail: milos.krstajic@uni-konstanz.de.*
Enrico Bertini is with the University of Konstanz, Germany. E-mail: enrico.bertini@uni-konstanz.de.
Daniel A. Keim is with the University of Konstanz, Germany. E-mail: daniel.keim@uni-konstanz.de.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

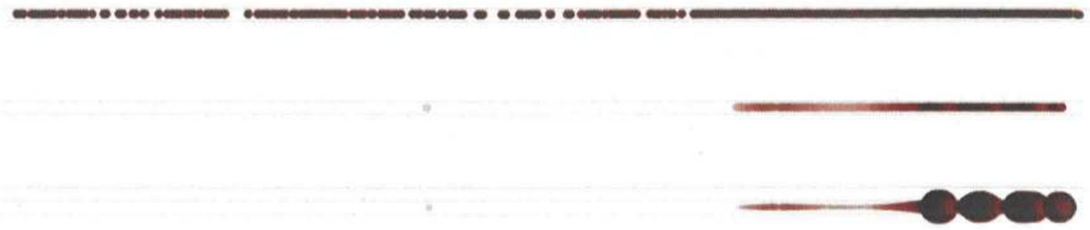

Fig. 2. Step-by-step construction of a single CloudLine. *(top)* A sequence of event data on a linear timeline, where each event is represented by a circle. It is practically impossible to distinguish between the high and low density areas.*(middle)* The same sequence after mapping the importance function to opacity of each circle. Importance function is described in Section 4. The change in opacity levels shows the structure of the sequence better. However, overplotting and the fact that maximum opacity can be reached very quickly don't allow for finer details in high density areas to be shown. *(bottom)* Importance function mapped to both opacity and size of the circles. Several event episodes can be identified much more easily.

representations that would allow the user to explore the low level details of interest within long time series, to the best of our knowledge there is little work that deals with multiple time series event data.

To address these issues we developed CloudLines, a technique for interactive visual analysis of multiple time series event data. We use timeline distortion techniques to accommodate more space for recent data for interaction with individual data items. At the same time, we employ Focus+Context approach to allow inspection of data points in highly compressed areas. While giving direct access to individual data points (*events*), our technique also allows quick detection of *event episodes*, i.e. high density sequences of events.

In this paper we describe the following contributions:

- A compact visualization for time series event data, which: 1) takes into account the distribution of the data points, 2) shows recent data with minimal overlap, and 3) allows easy detection of important event episodes.
- A lens-based interaction for direct access to overlapping events.
- Demonstration of the approach by applying it to multiple time series acquired from the real-world news streams.

2 Problem Description

In this paper, we are addressing the problem of designing a compact display for multiple event-based long time series data, which would provide global overview of the whole dataset and allow *atomic* analysis of individual events. Besides the fact that the limited screen real estate causes high overlapping of these data, we are also addressing the problem of putting the more relevant recent data in the context of the past. Our space-efficient solution helps in comparing multiple time series at once, while it can also be used for displaying single time series when vertical space is limited (for example, on the screens of mobile devices).

Data . We're working with *events*, i.e. time-stamped data points that arrive in data streams at non-equidistant time intervals. The changing rate at which the events arrive creates new data processing and visualization challenges, which usually do not exist when working with streaming data that comes in at regular intervals, such as any type of measurement data. Additionally, each event contains not just a numerical value, but is rich in content, providing meta information that is of great interest for the user. Therefore, it is important to allow direct interaction with events on the atomic level. Events can belong to different categories, or we can acquire different event data streams for which simultaneous analysis is needed. Since our timestamps are not equidistant, sequences of events that are close in time can be defined as *event episodes*. These high density areas are of great importance, because they show important local features of the event data stream. On the other hand, low density areas could be under certain conditions regarded as noise.

3 Related Work

3.1 Time Series Visualizations

A lot of research has been conducted on visual analysis of time series. A framework for categorization of different approaches is described by Aigner et al. [1]. In [24], Müller and Schumann provide a review of time-dependent data techniques. Pixel visualization for time series in circle segments is described in [2], time series bitmaps used for working with large time-series databases in [21], while more details about multi-resolution techniques for large time series can be found in [10]. Javed et al. [14] compare different techniques for visualization of multiple time series. In [25], Palpanas et al. describe amnesic functions for approximation of the time series data with fidelity proportional to its age.

Data streams are continuously updated, and time-stamped data arrive in rapid, time-varying fashion making the volume of the stream unpredictable and unbounded [4]. Design of effective visualizations is strongly constrained by these dynamic properties [8]. LiveRac, a tool for visual exploration of system management time series data is described in [23], where semantic zooming of time series is presented.

3.2 Focus+Context

SignalLens [16] is a method for Focus+Context analysis of single long time series from electronic measurements. Details about lens construction can be found in the Framework for unifying presentation space by Carpendale and Montagnese [6] and details about achieving high magnification in [7]. Sigma lenses [26] use dynamic translucence to create transition between focus and context.

3.3 Text Data Streams

In the analysis of textual time series collections, visualization of term trends in a collection of documents in a stacked graph is described in ThemeRiver [11]. Text visual analytics system TIARA [28] extends the stacked graph visualization metaphor used in ThemeRiver to show development of topics in collections of emails and patient records by integrating tag clouds in the topic layers. Thread Arcs [15] visualizes threads from a collection of emails. EventRiver [22] is a visual analytics system for analysis of temporal development of news events and event groups retrieved from broadcast news data. In [19], the authors present an algorithm for real-time news aggregation into article threads with a dynamic visualization technique, where *important* article threads are shown aligned on a time axis to give insight into their temporal development. Incremental change in collection of documents which are projected in 2D plane by highlighting new (fresh) and old (stale) documents is presented in [13]. The use of animation for transitions in time series can be found in the taxonomy by Heer and Robertson [12].

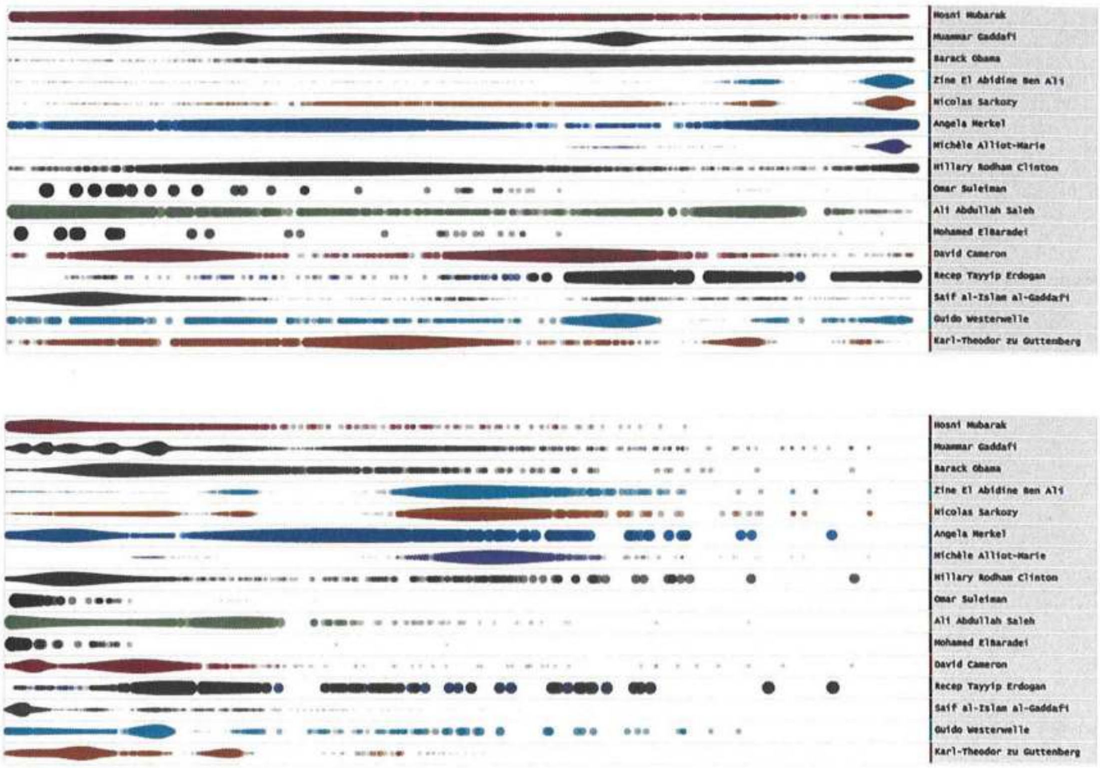

Fig. 3. Linear (top) and logarithmic (bottom) timeline with multiple time series. While both visualizations allow detection of event episodes, logarithmic timeline allows direct interaction with recent events. The data used in this figure is weekly data described in Section 6.

4 Design

When recent events have higher importance, we allow logarithmic distortion of the timeline to provide easy access to these events while keeping an informative context of the past. Each event (data item) is displayed as a circle of minimum object size for visibility and interaction. Circular objects allow better distinction than rectangular in case of slight overlap. Logarithmic timeline allows for smooth transition within the distorted display space, while other approaches that deal with space partitioning were proposed in the past [10].

Let's take a naive approach. Linear scaling of one month of event data without overlapping and aggregation on a 1,200 pixels wide display, with event object width of 5 pixels, would allow a total of 240 events. With 1 pixel not overlapping this number could go up to 1,200, thus creating a line of indistinguishable event data. By using object transparency this number could theoretically go up to 120,000 (assuming that each object is shown at 1% of full opacity). However, this hypothetical example would make individual item inspection completely meaningless. Besides, this would mean around 40 pixels for each day for one month of data, which is less than 2 pixels per hour.

By giving more space for recent data, and compressing the past, under the same circumstances, we provide much more space for the inspection of individual data items in the present, while creating high compression of the data in the past.

Event episode as visual cluster. In a less radical scenario, where the ratio of *available space* to *observed time window of interest* is more favourable and the distortion of the time axis is not needed, our visualization can allow the user to easily distinguish between areas with high density (event episodes) and areas with low density. Typically, detection of event episodes can be solved by combining data aggregation and different visualization techniques, e.g. by plotting frequencies at discrete time points, interpolation and stacking [11] or histogram and cubic spline interpolation [22]. However, these approaches display pre-calculated clusters of data, while we're working on the atomic level, which reveals finer structure of the data and shifts the decision making process of what is a cluster from an automated algorithm to the user. Also, this approach makes the precomputation less expensive.

Color . Multiple time series require compact visualization technique that will use as little amount of vertical space as possible. To support easy row identification, besides using positioning, we're coloring time series converted from color map suggested by ColorBrewer [5], which suggests a total of 11 different colors for qualitative data. Since the positioning along the y-axis is fixed in our case, we can repeat this color map on a larger number of categories without confusing the user. During our experiments with different time frames, datasets, and distortion functions, we noticed that the technique also works well in analysis of single time-series data for getting a quick overview and detection of potentially interesting areas. In these cases, color could be mapped to a different feature of the data.

By using real-world data, we have identified two overlapping factors that make detection of event episodes difficult, regardless of the linear or distorted timeline. The first factor is overlapping that is due to limited space, which can lead to event episodes overlapping each other. This effect is especially noticeable when working with distorted timelines. The second factor is the low density event data that is actually noise. To deal with this problem, we combine *kernel density estimation*, *truncation function* and *lens distortion as focus plus context technique*.

4.1 Density estimation

The overplotting effects in time series are a well known problem that may be solved to some extent by aggregation or sampling, thus leading to the loss of information [10]. In our approach, we are keeping aggregation to the minimum and leave the visual cluster identification to the user. Since we have a high overlap of event data in the past due

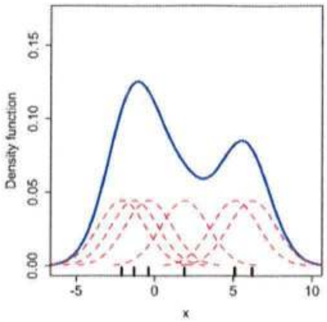

Fig. 4. Kernel density estimate (blue curve) showing individual Gaussian kernels constructed around data points on the horizontal axis (red dashed curves) [29].

to high logarithmic compression, we need to find a way to "penalize" low density regions, while "rewarding" high density regions.

As we have described earlier, overplotting can be caused by different factors and can not be easily avoided when working with data on atomic level. To overcome this problem, we propose a method that combines event importance function with density estimation and cut-off (truncation) function. Kernel density estimation is a non-parametric method for estimation of probability density function of a variable. The theoretical background can be found in [27] and a simple example of constructing a kernel density estimate from several data points using Gaussian kernel is shown in Figure 4.

The idea behind using kernel estimator is to amplify the signal coming from the high dense areas and at the same time to reduce the signal in the low density areas.

4.2 Importance function

For an event e_i, where $i \in (1,...,N)$, importance function $imp(e_i)$ is defined as

$$imp(e_i) = f(t) \times g(d_i(t)) \tag{1}$$

where t is the *age* of the item e_i, $f(t)$ is *decay function* and $g(d_i(t))$ is a *density factor*.

Importance function for each individual item is mapped to its size and opacity. Decay function is position dependent part of the importance function. Depending on the amount of compression, noise and data volume, $f(t)$ can be defined as exponential decay function:

$$f(t) = f(0) \times e^{(-t/\tau)} \tag{2}$$

where τ is mean lifetime and f(0) = 1.

Density factor is density-dependent part of the importance function and it is calculated as

$$g(d_i(t)) = Y_{max} \times \frac{d_i(t)}{D_{max}(t)} \tag{3}$$

Here, $d_i(t)$ is kernel density estimate of event e_i at time t, $D_{max}(t)$ is the maximum density estimate in the current time window, and Y_{max} is coefficient used to normalize the height of the objects to maximum allowed height for each time series. We use the density factor for highly compressed areas, where the overlap is high because of the size of each individual item, thus rewarding high density areas and penalizing low density areas. It is recalculated at user-specified time intervals to accommodate new data, with smooth transitions between the old and the new values.

4.3 Cut-off function

By estimating densities of our event data points, we decreased the impact of low density areas on the information display. However, the high volume of long time-series in these areas can still significantly affect the calculations, interaction and rendering of the visualization. This is especially the case when long time series contain *bursts* [18] of events around few peaks. When working with news data, like in our motivating example, we can expect to have such *data shapes* and low density areas can be treated as noise. Time-series can be truncated to areas around the peak, like in [30], where data is truncated from 1 year to 128 hours, when it falls to around 10% of the peak value.

In our case, the newer data is considered more important than the older data and this method can't be directly applied. Since the old events are less important then the new events, the noise removal threshold should increase with the age of data. Therefore, a cut-off function is needed, which is used to decide if the item e$_i$ should be visible or not. As a positive side effect, the compressed old data will leave only the most important event episodes due to a stronger truncation.

CloudLines technique implements different cut-off functions, such as logarithmic, square root, exponential or quadratic to compare their effect. Experiments with different datasets showed that concave functions in general give better results.

$$cutoff(x) = \frac{D_{max}(t)}{X_{max}} \times log(\frac{X_{max}}{T_{max}} \times t) \tag{4}$$

where $D_{max}(t)$ is the maximum density in the current time window, X_{max} is the width of the display, and T_{max} is the width of the time window. Therefore, if density value d_i of an event e_i happening at time t_i is greater than the value of the cut-off function at time t_i, the item should be visible.

The results of mapping the importance function and the cut-off function to opacity and size of objects representing event data in a single time-series are shown in Figure 2. To demonstrate the idea of saving screen space with our technique, we have compared a line chart and a CloudLine, showing the same sample dataset, in Figure 5.

4.4 Static vs Dynamic Visualization

The technique could be used in both static and dynamic scenarios. It is known that animation is not suitable for describing change of many objects on the screen, but under some circumstances, when the user is given the possibility to control the playback, the animation can be useful. In these cases, it is necessary to find a good way to show new data that was acquired while the user was analyzing the old dataset.

In our tool, it is possible to automatically adjust the time window of the observed time series and switch between the static visualization, where new data is added at fixed display coordinates, and the dynamic visualization, where data items move from present to the past according to the underlying trajectory function. In our approach, although not all of the used techniques and algorithms are incremental by nature, it would be possible to extend the approach into a fully incremental solution. Currently, the algorithm used to calculate kernel density estimates at observed data points is rerun at constant user-defined time intervals on the whole dataset, but it could be replaced with an online kernel density estimation algorithm, to adapt the densities on the incremental updates only. To avoid abrupt changes in the display that could be caused by sudden change of the density values, we used smooth animated transitions that allow the user to understand the changes.

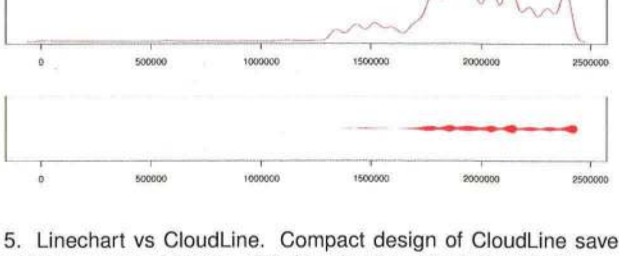

Fig. 5. Linechart vs CloudLine. Compact design of CloudLine saves expensive screen real estate, while keeping important visual features of the time series shape.

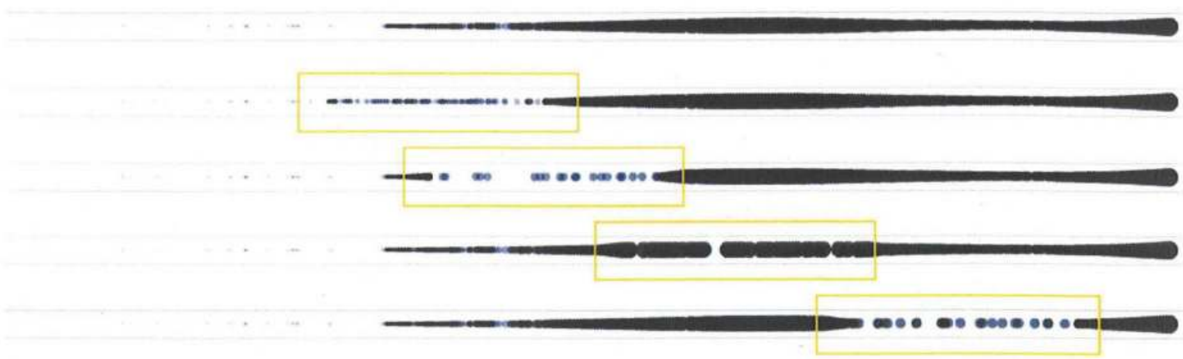

Fig. 6. Lens magnification of a single time series in different regions of interest. The time series shows weekly event data (Feb 21 - Feb 28, 2011) for Barack Obama. The details about the dataset are given in the Section Examples. From top to bottom: 1) Original CloudLine without magnification; 2)-4) Magnified areas are shown with the yellow rectangle.

5 Interaction

5.1 Lens magnification with event data

The visual density of the display can lead to the occlusion of fine details in a large dataset. This is especially noticeable in the regions of "maximum age" and high data density, when the logarithmic distortion of the timeline is used. To overcome this problem, we couple proposed visualization method with magnification lens distortion technique, which allows a fine-detailed analysis of individual event data points. Focus+Context approach provides a detailed view [9] of an interesting short interval within the global overview of the long time series. The proposed focus plus context technique is further improved to allow undistorted view in the focus region, while providing smooth transition in the areas connecting the focus and context space.

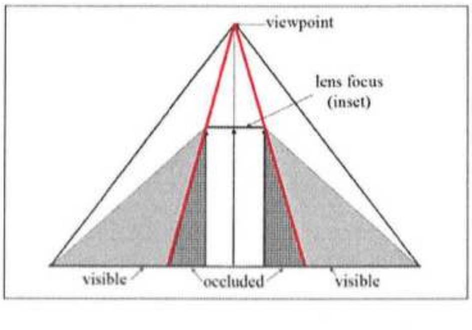

Fig. 7. Lens schema from [7]

Lenses have been extensively researched in the human computer interaction and information visualization communities. Only recently [17] high magnification lenses have been more thoroughly researched as a suitable Focus+Context technique in long time series. In our approach, we employ a similar idea with two important differences: 1) our time series are event-based, not aggregated, and 2) we propose an improvement of a classical lens design that provides undistorted display of the focus region in a distorted space.

Similarly to SingalLens design [17], our lens design is based on a model described by Carpendale and Montagnese in their Elastic Presentation Framework [6]. Each lens is described by a focus region of a width f, magnification factor m and drop-off areas d_f in which different functions can be used to give smooth transition between focus and context regions. In [7], Carpendale et al. have described different distortion functions used to achieve high magnification in Elastic Presentation Framework.

In CloudLines, we employ two different types of magnification lenses: 1) lens with linear magnification, and 2) lens with exponential magnification of logarithmic content.

Interaction mechanics. The size of the lens, as well as magnification level and the width of the focus region can be set by the user. When the lens is turned on, the user can inspect the time series by click-and-drag, and when the mouse button is released, the lens is left in its last position. When an object inside the focus region is selected, the respective event details are given on demand. If the user clicks outside of the lens area, the lens is repositioned and can be moved around the screen until this feature is turned off.

Details on demand. Working with time series on item level is meaningful when each item carries information and this information has to be available for direct access for the user. For example, when working with news event data, where an event represents a time-stamped news article, each item can be enriched with different metadata information, such as the url of the article, title, image, etc. This requirement is reflected in our first decision to show a single event as an object that has a minimum size that is appropriate for interaction. Sometimes, more items can arrive at exactly the same time, and in these cases, the user should be provided with details on demand for all underlying data.

6 Examples

To demonstrate our approach, we applied it to the news entities time series data collected by EMM news service [3] [20]. We present two examples that differ in volume and time range and show how CloudLines can be used in two different scenarios: 1) exploration of multiple long time series from one month of news event data; and 2) monitoring of 24 hours of news event data, and discuss the benefits and drawbacks of our solution.

6.1 CloudLines for multiple long time series news exploration

The entity event data represent the most mentioned politicians during February 2011 and each event contains information about a news article that is mentioning a specific person (shown in Figure 8). When working with multiple time-series, finding an optimal way to sort the time-series is a difficult problem that goes beyond the scope of this work. The problem is addressed in detail in [30] and relates to the question of finding an optimal time series similarity measure. The volume of distinct time series can vary greatly and can be difficult to compare. When working with incremental data, this problem has to take into account re-sorting and has to allow for efficient computation. In our case, we have chosen 16 time series with highest overall popularity during the whole month and sorted them in a descending order.

Normalization. Time series can be normalized according to the different criteria, such as normalization to the overall volume of each time series, or to the peak value, etc. In Figure 8, the time series are normalized to the maximum peak value for each distinct time-series,

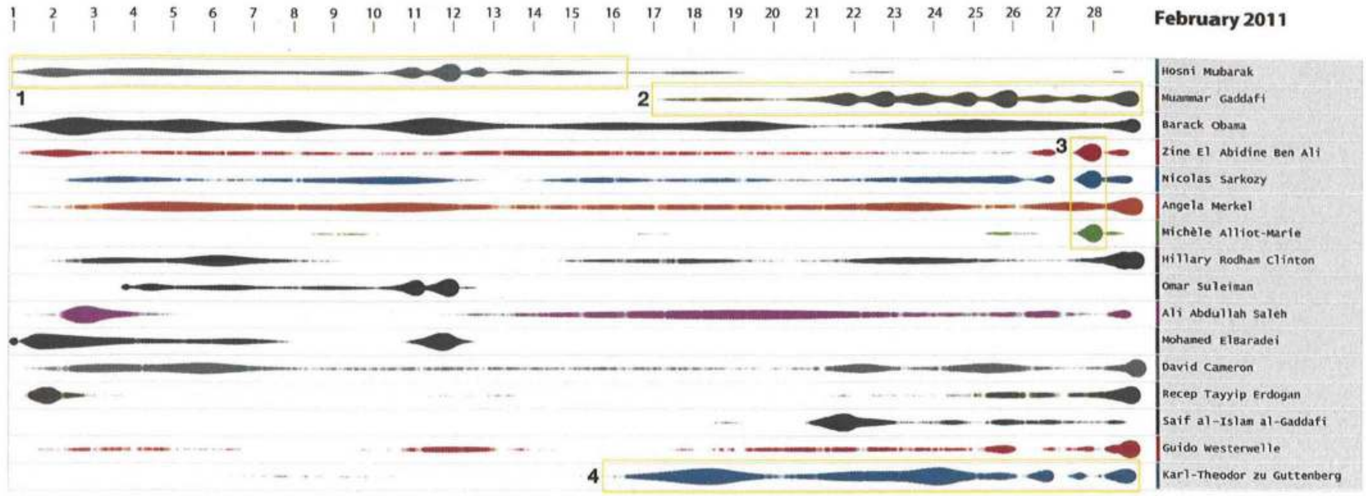

Fig. 8. CloudLines applied on the news entities data taken during February 2011. Dataset consisting of around 117,000 events for 16 politicians with the highest volume during the whole month are shown (names are displayed on the right side). Compact visualization immediately reveals important event episodes. The fine structure of the data, such as the "pulse" of online publishing is also visible, showing higher activity in the afternoons. Detailed analysis of the visualization is given in the Subsection 6.1.

since we would like to be able to see similar event episodes when they occur across different time series. Alternatively, the user can also choose to normalize the data to the maximum overall peak in our tool.

Looking at the visualization, several event episodes stand out and details can be revealed through interaction, which is described in the Section 5. After analyzing these interesting visual clusters, the highlighted and numbered regions in Figure 8 relate to the following events:

1. **The crisis in Egypt during the first half of February 2011.**

2. **The crisis in Libya in the second half of February 2011.**

3. **A short event episode related to Michele Alliot-Marie, France's foreign minister and her ties to Tunisia's ousted president Ben Ali, after which she resigned from her position.**

4. **The (German defense minister) Guttenberg plagiarism scandal.**

Additionally, the visualization reveals the fine structure of the data, such as a higher volume of published articles in the afternoon, describe the "pulse" of online news sources. The goal of the examples in our work is to demonstrate the validity of our technique for quick detection of important temporal event sequences in multiple long time series, although deeper analysis of each of these events is also possible.

The example shows one important feature of this type of event data: *the data varies greatly in shape and it can't be easily modeled as bursts [18].*

6.2 CloudLines in News Monitoring

In our second example (shown in Figure 9), we are presenting the results of our approach on 24 hours of news event data in a monitoring scenario with the same choice of named entities as in the previous example. This time window is chosen for a scenario in which the user is aware of the current events, but needs to be able to follow any new incoming data. The visual density is much lower and allows the analyst to inspect individual events more easily.

The Figure 9 shows transformation of the visualized data with our method applied in consecutive steps, giving the final result in the last subfigure. Direct access to the new items is available immediately, and the user can interact with the newest events without magnification.

7 Discussion and Conclusion

Discussion . In this section we will give some additional remarks regarding the scalability and weaknesses of our approach. The techniques used in our approach can be adjusted to various application scenarios to satisfy different scalability and performance criteria. For the analysis of non-incremental data, offline density estimation has to be computed only once and the rendering could be optimized for highly overlapping events. In our first use case, around 117,000 events are processed and shown in Figure 8. To speed up the rendering, the overlapping events could be truncated and represented by a single object whose opacity would be the sum of underlying events, if their distance is less than a pre-defined ε, which depends on the loss of information that is acceptable for the user. The challenge, which is a part of our current research efforts, is seamless retrieval and re-rendering of truncated events in the focus region during user interaction. In our second case (Figure 9), which is focused on a shorter time window (24 hours), around 7,000 events are shown.

Kernel density estimators are known for having problems with tail regions of the distribution. This is especially problematic with the latest events, which will get undersmoothed due to the applied method. Second, density estimation is performed offline and re-run at user-specified time intervals. This fact currently prevents our design from being fully incremental for (near) real-time data stream monitoring. However, this problem can be overcome by applying an appropriate online density estimation algorithm and will be part of our future work.

The basic idea of our approach is to display multiple time series on atomic (event) level, i.e. without aggregation, and to allow *in situ* analysis of single events. The size of the event objects can introduce overplotting, especially in high density areas. The objects would have to be 1 pixel wide to minimize this problem, however the interaction would be practically impossible with such a design. There is no easy solution to overcome this problem for each and every dataset and limited screen space. Therefore, when high density areas (event episodes) are identified, the magnifying lens is used for direct inspection, thus removing overplotting in the focus region. In Figure 9, it can be seen that circles of the same size have different transparency. This is due to the fact that the events occur at the exact same points in time. The difference in size between circles strongly depends on the chosen kernel bandwidth and local density estimate. The effect of alpha-blending overplotting increases with circle size significantly and this perceptual issue is a limitation that we would like to address in our future work.

Conclusion . In this paper, we have presented a novel interactive visualization approach for the analysis of multiple time series in lim-

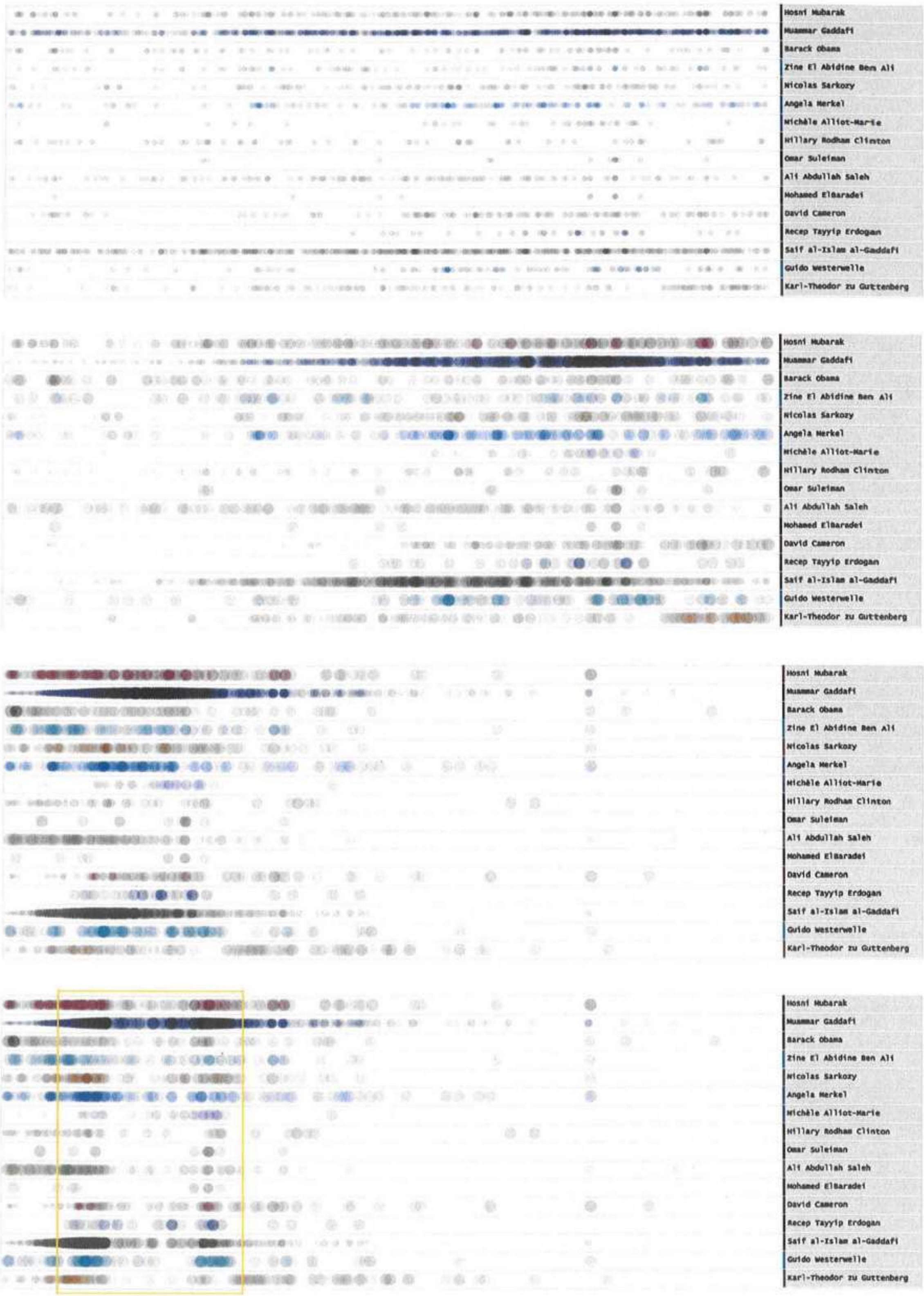

Fig. 9. News monitoring scenario: Transformation of news entity event data in a 24 hour time window, taken on February 21, 2011. The data is appearing on the x-axis from left to right (12:00 AM is the boundary on the left, 11:59 PM is on the right). 6,756 events are shown. From top to bottom: 1) Raw data with high overlapping on a linear timeline; 2) Density based transformation; 3) Logarithmic distortion of the timeline gives more space for the latest events and compresses the past data into visual clusters; 4) Magnification lens applied on the past - focus region with lower density is now visible, and drop-off functions create more compressed areas that connect focus and context. Interaction with individual events is now possible.

ited space. The idea behind CloudLines design is to explore the limits of large scale event data analysis on *atomic* level in limited space. Our first results show that our method with proposed Focus+Context techniques is worth further research. In our future work, we would like to evaluate our method with a formal user study. While we haven't conducted such a user study so far, the examples show the promising direction of our approach.

To demonstrate the usefulness and effectiveness of our approach, we have applied the techniques on a high-volume real world dataset in two extreme case examples. In our first example, we worked on one month of news entities event data and showed how the proposed technique can be used to:

- detect important event episodes in the time series
- show the fine structure of the event episode shapes
- interact with time series on *atomic* level

In our second example, we used 24 hour of news entities event data and demonstrated how the technique could be used for real-time monitoring in short time frames. Additionally, our experiments showed that the proposed visualization method could also be suitable for single time series. We believe that our approach can scale well with different data volumes, display resolutions and time ranges.

As part of our future work, we would like to improve the current lens magnification technique. Currently, we're using a cut-off function which truncates the data in order to remove the noise in the overview and increase performance. We allow the interaction in the regions of interest by using lens magnification as a Focus+Context technique. This technique could be improved by integrating a variation of semantic zoom, which would allow the retrieval of removed items at different zoom levels, based on their importance values.

ACKNOWLEDGMENTS

The German Research Society (DFG) under grant GK-1042, "Explorative Analysis and Visualization of Large Information Spaces" partially supported this work.

REFERENCES

[1] W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski. Visualizing time-oriented data–A systematic view. *Computers & Graphics*, 31(3):401–409, 2007.

[2] M. Ankerst, D. A. Keim, and H.-P. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Visualization '96, Hot Topic Session, San Francisco, CA*, 1996.

[3] M. Atkinson and E. Van der Goot. Near real time information mining in mulitlingual news. In *WWW '09: Proceedings of the 18th international conference on World Wide Web*, pages 1153–1154. ACM, 2009.

[4] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '02, pages 1–16, New York, NY, USA, 2002. ACM.

[5] C. A. Brewer. ColorBrewer, 2011. http://www.ColorBrewer.org, accessed on March 28, 2011.

[6] M. S. T. Carpendale and C. Montagnese. A framework for unifying presentation space. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pages 61–70, New York, NY, USA, 2001. ACM.

[7] S. Carpendale, J. Ligh, and E. Pattison. Achieving higher magnification in context. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 71–80, New York, NY, USA, 2004. ACM.

[8] G. Chin, M. Singhal, G. Nakamura, V. Gurumoorthi, and N. Freeman-Cadoret. Visual analysis of dynamic data streams. *Information Visualization*, 8(3):212–229, 2009.

[9] G. W. Furnas. A fisheye follow-up: further reflections on focus + context. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 999–1008, New York, NY, USA, 2006. ACM.

[10] M. Hao, D. Keim, U. Dayal, and T. Schreck. Multi-resolution techniques for visual exploration of large time-series data. In *Eurographics/IEEE-VGTC Symposium on Visualization, 23 - 25 May 2007, Norrkoeping, Sweden*, 2007.

[11] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[12] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE transactions on visualization and computer graphics*, 13(6):1240–1247, 2007.

[13] E. G. Hetzler, V. L. Crow, D. A. Payne, and A. E. Turner. Turning the bucket of text into a pipe. In *INFOVIS '05: Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 12. IEEE Computer Society, 2005.

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

[15] B. Kerr. THREAD ARCS: An Email Thread Visualization. *Information Visualization, IEEE Symposium on*, 0:27–218, 2003.

[16] R. Kincaid. Signallens: Focus+context applied to electronic time series. *IEEE Transactions on Visualization and Computer Graphics*, 16:900–907, November 2010.

[17] R. Kincaid. Signallens: Focus+context applied to electronic time series. *IEEE Transactions on Visualization and Computer Graphics*, 16:900–907, 2010.

[18] J. Kleinberg. Bursty and hierarchical structure in streams. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 91–101. ACM, 2002.

[19] M. Krstajic, E. Bertini, F. Mansmann, and D. A. Keim. Visual analysis of news streams with article threads. In *StreamKDD '10: Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques*, pages 39–46, New York, NY, USA, 2010. ACM.

[20] M. Krstajic, F. Mansmann, A. Stoffel, M. Atkinson, and D. Keim. Processing online news streams for large-scale semantic analysis. In *1st International Workshop on Data Engineering meets the Semantic Web*, 2010.

[21] N. Kumar, N. Lolla, E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *SIAM 2005 Data Mining Conference*, pages 531–535. SIAM, 2005.

[22] D. Luo, J. Yang, M. Krstajic, W. Ribarsky, and D. Keim. Eventriver: Visually exploring text collections with temporal references. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints), 2010.

[23] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: interactive visual exploration of system management time-series data. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1483–1492. ACM, 2008.

[24] W. Müller and H. Schumann. Visualization for modeling and simulation: visualization methods for time-dependent data - an overview. In *Proceedings of the 35th conference on Winter simulation: driving innovation*, WSC '03, pages 737–745. Winter Simulation Conference, 2003.

[25] T. Palpanas, M. Vlachos, E. Keogh, and D. Gunopulos. Streaming time series summarization using user-defined amnesic functions. *IEEE Trans. on Knowl. and Data Eng.*, 20:992–1006, July 2008.

[26] E. Pietriga and C. Appert. Sigma lenses: focus-context transitions combining space, time and translucence. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1343–1352, New York, NY, USA, 2008. ACM.

[27] M. P. Wand and M. C. Jones. *Kernel Smoothing (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, 1 edition, Dec. 1994.

[28] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 153–162, New York, NY, USA, 2010. ACM.

[29] Wikipedia. Kernel density estimate, 2011. http://www.wikipedia.org, accessed on July 1, 2011.

[30] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *ACM International Conference on Web Search and Data Minig (WSDM)*. Stanford InfoLab.