Integrated Spatio-temporal Storyline Visualization with Low Crossover

No Author Given

No Institute Given

Abstract. A story is a chain of events that have multiple dimensions, including time, location, characters, actions, and context. Storyline visualizations attempt to visually present the multiple dimensions of a story's events and their relationships. However, integrating the temporal and spatial dimension in a single visualization view is often challenging. One of the main reasons is that spatial data is inherently 2D while temporal data is inherently 1D. We propose a storyline visualization technique that integrate both time and location information in a single view while maintaining regular ordering of temporal and spatial dimensions. In particular, we propose a heuristic method to reduce excessive line crossovers for large data sets. Through case studies, we show that this visualization method provide a good balance in integrating temporal and spatial dimensions.

1 Introduction

A story is an account of related events, and each event can be seen as a multidimensional data item. Storyline visualizations attempt to visually present the sequence of events and the relationships among multiple dimensions of these events. There has been a growing interest in storyline visualization in recent years. For example, Kosara and Mackinlay [1] have argued that storytelling is the next logical step for visualization and that storytelling is as important a research subject as visual exploration and analysis.

Storyline visualization can be seen as a type of multidimensional data visualization. Most events share some common dimensions such as characters, time, location, actions, and context. The many existing storyline visualization techniques differ in what dimensions they choose to depict and how each dimension is visually presented. In this work, we focus on visualization techniques that try to integrate both time and location in one view. We call this type of problem integrated spatial-temporal storyline visualization.

Integrating both time and location in one view is often desirable. It allows users to correlate temporal and spatial information quickly. However, such integration has long been a challenging problem. Existing spatial-temporal storyline visualizations can be classified into four categories: map-based visualizations, timeline-based visualizations, unconventional, and hybrid approaches. Map-based visualization designs center around a map, with temporal information displayed either on top of the map in a 3D view or overlaid on the map in a 2D view. The timeline-based visualizations center around a timeline chart. Hybrid approaches use both map and timeline in a synchronized multiple view configuration. Unconventional approaches use neither a timeline or a map.

In this paper, we present a novel hybrid approach for integrated spatial-temporal storyline visualization. Our method includes a timeline-based visualization and a synchronized map. The timeline-based visualization is built on our previously published work but with significant improvement. A major problem with timeline-based approach is the large number of line crossovers. We have developed a new heuristic algorithm to reduce line crossings for our timeline-based visualization technique. We also show that our timeline-based method maintains regular ordering for both temporal and spatial dimensions.

Through case studies, we demonstrate that this hybrid approach is effective in providing an integrated visualization of the multiple dimensions of a story. A notable benefit is its effectiveness in depicting the context of events.

2 Related Work

Many storyline visualization (or narrative visualization) techniques have been proposed [2,3]. Segel and Heer [2] proposed a high level classification of storyline visualizations.

Most storyline visualizations include a temporal dimension, but many storyline visualizations do not include a spatial dimension [4–8]. Some storyline visualizations, such as Zhao, et al. [9], deal with neither time or location.

We are interested in storyline visualizations that deal with both temporal and spatial dimensions. Visualizing temporal and spatial dimension in a single view is desirable but difficult to achieve. The main reason is that spatial dimension is inherently 2D while temporal dimension is inherently 1D. The existing methods can be classified into four categories: map-based methods, timeline-based methods, hybrid methods, and unconventional methods.

Map-based methods can be further divided into two categories: 2D map-based methods, and 3D map-based methods. In 2D map based methods, the main challenge is how to show temporal information on a 2D map. Liu [10] draw timeline as a circle around the map. The temporal information is not directly overlaid on the map but instead linked through color coding. Sun, et al. [11] and Liu, et al. [12] experimented with different ways of overlaying temporal data on a 2D map: text label, visual symbol, color coding, and animation. In particular, they noted the difficulty of encoding timeline directions on a 2D map.

In 2D map-based methods, the geographical relationship of the events is preserved but their temporal dimension is not presented in natural sequence. For example, it is difficult to differentiate multiple events that happen at the same location but at different times. Bak, et al. [13] attempted to solve this problem by visualizing time as a circle around a location. This deals with event overlapping problem but the sequence of events is still not obvious.

3D map-based methods attempt to solve these issues by drawing timelines above a map. For example, Gatalsky, et al. [14], Andrienko, et al. [15], and Kapler, et al. [16] use 3D maps to integrate spatial and temporal data in one view. However, 3D view has its own drawbacks [17]. One of the main problems is occlusion. It's also difficult to perceive distance or height precisely in 3D views.

Timeline-based methods, location information is encoded on a timeline. In Munroe's hand drawn storyline visualization [18], locations are marked by the distance between lines at each timeframe. Crnovrsanin, et al. [19] adopted a similar approach: location is mapped to distances between entities or certain locations. Tanahashi and Ma [20] and Tanahashi, et al. [21] use a closed contour surrounding the events to depict location. Liu, et al. [22] improved on Tanahashi and Ma's [20]approach by handling location hierarchy with nested contours. The same location may be represented by multiple contours on the chart and they are linked by the same color coding. Both methods require substantial data preparation.

Timeline-based methods preserve the natural sequence of temporal dimension but the geographical relationship is often lost. For example, the same location may appear multiple times in different places on a timeline-based visualization. The location proximity is depicted in color. In some visualizations, only the distance between locations are visualized.

Another potential problem in timeline-based approach is excessive line crossovers [20]. Tanashida and Ma [20] proposed a number of design principles for storyline visualization and one of them is to minimize line crossovers.

Unconventional methods depict temporal and spatial data in more abstract and unfamiliar ways. For example, Cao, et al. [23] proposed a circular layout. Time is presented as nested rings. Locations are presented as circles on the rings. The distance between circles reflects the distance between their locations.

Many visualizations may be classified as hybrid methods with synchronized timeline-based and map-based views. For example, Crnovrsanin, et al. [19], Chu, et al. [24], and Wang and Yuan [25] are hybrid methods.

Our method is a hybrid method with a timeline-based visualization and a synchronized map view. The map view only shows the locations without temporal information but the location markers are synchronized with the timeline view. Our timeline-based visualization is different from previous timeline-based methods because our method maintains both regular ordering for temporal dimension and spatial dimension. This is an attempt to address some of the issues in previous timeline-based approaches where only the ordering of the temporal dimension is maintained. In addition, we address the problem of excessive line crossovers by proposing a new heuristic method for reducing line crossings.

3 Overview

A story is a chain of events and every event consists of four basic elements: time, location, characters, and context. For example, an event is often described as what some characters did at certain time and location, and in certain context. Although an event may contain much more information, such as interactions among many characters, our study focuses on these four essential elements because our storyline visualization is aimed at providing a high level overview of the story, rather than detailed descriptions.

Our visualization system is designed to handle data sets with different combinations of time, location, characters, and context. Fig. 1 show the architecture of our visual storytelling engine.

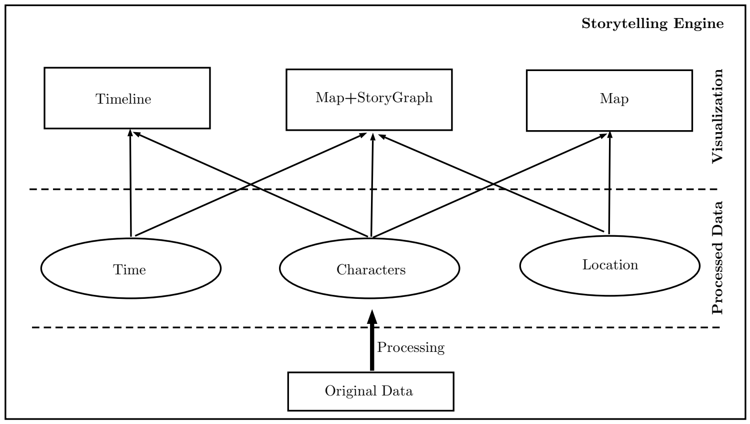


Fig. 1: The Architecture of the Visual Storytelling Engine

The data processing component processes original data sets and extract information about time, location, and characters. For structured data sets such as spreadsheets, such information is usually grouped into separate columns. So the data processing is relatively straightforward. For unstructured data sets, such as news articles or interviews, the task of extracting specific time, location, and characters may be difficult because of the complexity of processing natural language.

Currently our system can process structured data sets (e.g. spreadsheets) and some semi-structured data sets, but not unstructured data sets. A semi-structured data set is a human curated data set. For example, a user may read a series of interviews and enter the events into a spreadsheet. The user may do some light editing and separate the events, but do not have to extract every piece of time, location, or character information. Our program will attempt to extract time, location, and character information using Stanford CoreNLP tools. However, the natural language processing tools are not always accurate and there are ambiguities in extracted time and locations. This problem was dealt with in another publication of ours, and won't be repeated here.

Different data visualizations are created based on user's needs. The three typical use cases of storyline visualization are:

– Time + Character
– Location + Character
– Time + Location + Character

Conventional timeline charts are suitable for "Time + Character" type of visualization tasks. The location information is not presented. This is because users are not interested in the locations of the events, or because the location information is incomplete or unreliable. Such timeline charts are implemented with Google Chart Tools.

Interactive maps are suitable for "Location + Character" type of visualization tasks. The time information is not presented. Again, users may not be interested in time information or the time information is incomplete or unreliable. Such interactive maps are implemented with Google Maps API.

The tasks that require the examination of "Time + Location + Character" is the most challenging. And this is where our main contribution is. We have developed a new type of visualization called Storygraph, which is synchronized with an interactive map. The novelty of Storygraph is that it allows users to examine time, location, and character information in a single view. However, Storygraph has two drawbacks. The first is that the location information is presented as lines in Storygraph, an abstract form that is often difficult to interpret. This issue is handled by adding a synchronized maps to Storygraph. When a location line is selected in the Storygraph, the corresponding location is highlighted in the map, and vice versa.

The second issue with Storygraph, also pointed out by Tanahashi and Ma [20, 21], is excessive line crossings when visualizing large data sets. In this paper, we propose a new heuristic algorithm to reduce point cluttering and line crossings. This new algorithm is the new contribution of this paper.

In the following section, we will describe the interactive map, Storygraph, and the new heuristic algorithm.

4 Interactive Map

An interactive map is a natural choice for visualizing spatial data. Our interactive map is based on Google Map API, which support zooming, panning, and automatic centering of the map. We also implemented additional functions as below.

- A user can select a rectangular region on the map and a corresponding Storygraph for that region is automatically created.
- The map is synchronized with Storygraph.When a location line is selected in the Storygraph, the corresponding location is highlighted in the map. When a location marker is selected on the map, the corresponding location line is highlighted on the Storygraph.
- The map is connected with Google SpreadSheets, where user data sets are stored.

The interactive map and Storygraph provide two synchronized views of the same data set. This is beneficial because any temporal patterns found in the Storygraph can be simultaneously analyzed on the map for their spatial relationship.

5 Storygraph

We have developed a new visualization technique, called Storygraph, to present spatio-temporal data. The details of this design and its benefits have been described in a previous publication. Here we only give an overview of the technique. A Storygraph consists of three axes: two parallel vertical axes for spatial data and one horizontal axis for temporal data. The two vertical axes, similar to parallel coordinates, represent 2D location coordinates such as latitudes and longitudes. Each location (x,y) or (latitude, longitude) is represented as a line segment called location line. Location lines are created by joining the corresponding location coordinates on the two vertical axes. Therefore, a single location on a map becomes a line segment in Storygraph. The benefit is that events occurring at the same location on different times can be plotted as markers along the location line. Figure 2 is an example of a Storygraph. Circles on the line represent the events that happen at the same place at different times.

One of the drawbacks of using location lines is that they are too abstract to be interpreted as locations. This problem is alleviated by adding a synchronized interactive map, which is described in the above section.

Another drawback comes from the location line crossings. Excessive line crossings make the visualization hard to read, and create a false sense of location proximity among crossed lines, causing confusion. Therefore reducing the number of line crossings is a necessary task for Storygraph. In order to look into the problem of line crossings, we need to first treat a Storygraph as a bipartite graph. A bipartite graph can be defined as $G = (V, W, E)$, where V is the first disjoint node set on one vertical axis, W is the second disjoint node set on another vertical axis and E is the edge set connecting nodes in V to nodes in W. The problem of minimizing the number of crossings between edges in a bipartite graph was first introduced in [26]. This problem can be seen as two different problems: the level permutation problem (LPP) where the order of nodes in one node set is fixed, and the bipartite drawing problem (BDP), where both node sets of the bipartite graph can be subject to permutation. LPP and BDP are proved to be NP-hard by Eades and Wormald [27] and by Garey and Johnson [28], respectively. Examples of heuristics for LPP include Barycenter heuristics [29] and median heuristics [30]. Examples of heuristics for BDP encompass MinSort heuristics [31] and Barycenter heuristics [29]. Genetic Algorithm is also applied to solve both LPP and BDP [32, 33].

If we consider latitude points to be V, longitude points to be W, and location lines to be E, then line crossing problem in a Storygraph is equivalent to the edge crossing problem in a bipartite graph. The basic idea for minimizing edge crossing in a bipartite graph is to change the order of points on one axis (level permutation problem or LPP) or both axes (bipartite drawing problem or BDP). Although we can use the same idea to alleviate the problem of line crossings in Storygraph, the rearrangement of points on axes may create other confusion to the readers of Storygraph. So we think running the line crossing reduction algorithm in a user-defined region would be a better idea since it can balance the requirement of reducing the confusion caused by line crossings in the area of interest and maintaining the natural order of latitude or longitude points on axes.

We will explore the level permutation problem (LPP) in the following section, which is a NP-hard problem in a general bipartite graph. We discover that LPP can be simplified with some assumptions in a setting such as Storygraph, and we also present a genetic algorithm that can reduce the number of line crossings while partially maintaining the order of points.

5.1 Storygraph with no identical latitude or longitude pairs

The line crossings in a Storygraph can be eliminated in $O(e)$ time given no identical latitude point or longitude point pairs exist in the Storygraph. This assumptions are inspired from our observation that in the data set of the Afghanistan War Diary [34] with approximately 60K highly structured records, about 70% of the records have unique latitudes or longitudes. It is likely that in a region of the data set this assumption is valid. If that is the case, then the algorithm can be quite straightforward: for each location line, locate its latitude in the vertical axis and plot the location line horizontally. Since location lines are parallel to each other, the number of line crossings is zero.

5.2 Storygraph with only identical latitude point or longitude point pairs

If there are only identical latitude/longitude points in the data set, then the algorithm discussed above is still useful. For each occurrence of identical latitude/longitude points, the number of line crossings it causes is the number of location lines between the current location line and the location line in which that latitude/longitude points firstly occurs. In Fig. 2, location lines (x1, y1) and (x4, y1) share the same longitude point y1. The number of line crossings is equal to the number of location lines between (x1, y1) and (x4, y1).

5.3 Storygraph with identical latitude and longitude point pairs

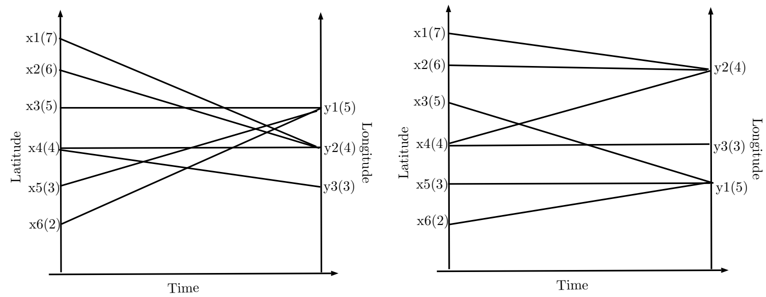
In the general case in which there exist identical latitude pairs and longitude pairs, our solution to this problem is to employ a modified median heuristics algorithm [30]. The heuristics is as follows:

The order of points on one axis is fixed. Points on the other axis are aligned horizontally to the median points of their adjacent points.

We illustrate this heuristics with examples in Fig. 2a and Fig. 2b. Fig. 2a is a Storygraph that does not apply the heuristics. There are points x1∼x6 on the latitude axis, points y1∼y3 on longitude axis and seven location lines. All points are plotted in the numerically ascending order. The number of line crossings is ten. In Fig. 2b, with the modified median heuristics, the position of x1∼x6 on the latitude axis remain unchanged, and on the longitude axis y1∼y3 will be moved so that they are aligned to the median points of their adjacent points on the latitude axis. For example, y1 is adjacent to x3, x5 and x6, and then y1 should be moved to position 5 on the longitude axis which is horizontally aligned to the position of x5 on the latitude axis. With the modified median heuristics, the number of line crossings is reduced to two.

5.4 Storygraph maintaining partial latitude/longitude order

The above algorithm to LPP aims to reduce line crossings in Storygraph, and take little consideration onto the latitude/longitude point order. Yet in reality sometimes people would like to reduce the number of line crossings, perhaps not to the optimal extent while maintaining the partial order of points. In such case, a genetic algorithm can be a convenient solution. A genetic algorithm is based on the principle of Darwinian natural selection. It starts from a usually randomly generated population, which is a set of possible solutions to the problem, measures the quality of the population with the user-defined function called fitness, and determine which of the individuals in the



(a) Layout of the Storygraph without line crossing reduction (b) Layout of the Storygraph after line crossings reduction

Fig. 2: Storygraph before and after line crossing reduction

population can survive. The surviving individuals can reproduce the next generation of the population with mutation and crossover. Then the same process repeat again and again until the fitness value reach the threshold. We discuss the key points in the genetic algorithm as follows assuming we will change the order of latitude points:

- Representation of solutions: each possible solution can be represented as $(x_1, x_2, x_3 ... x_n)$ where n is the number of latitude points and x_i the order of i_{th} latitude points in the original ascending latitude points.
- Population in the initial generation: the population composed of many individual solutions are generated randomly.
- Fitness function: the quality of solutions are measured by two metrics. One is the number of lines crossings, the other is the number of reverse pairs in the solution. If a latitude point A is arranged ahead of another latitude point B, and A is greater than B, then A and B is a reverse pair. The fitness function is described as $F = a1 * (n1/n) + a2 * (r1/r)$, where $n1$ is the number of lines crossing in the current configuration while n is the number of line crossings in the worse case. $r1$ is the number of reverse pairs while r is the number of reverse pairs in the worse case. By computing the ratio of line crossings and reverse paris in the current solution over those in the worse case, we can standardize these two metrics. $a1, a2$ is the weight of two metrics, and can be configured by users.
- Mutation operation: for a solution $(x_1, x_2, x_3 ... x_n)$, randomly generate two integer i and j, and rotate the elements between i_{th} element and j_{th} element in the solution, and generate a new solution.
- Crossover operation: reproduce a new child solution from two parent solution. The child solution inherits elements between two randomly generated crossover points from one parent. The remaining elements are inherited from the alternate parent in the same order as they appear in that parent.

6 Case Study

6.1 Afghanistan War Log (2004-2010)

The Afghanistan War Diary is a publicly available database that comprises significant U.S. military activity reports from Afghanistan during the period 2004-2010 [34]. With approximately 60K highly structured records, the data set includes rigorously categorized records suitable for spatial-temporal visualization.

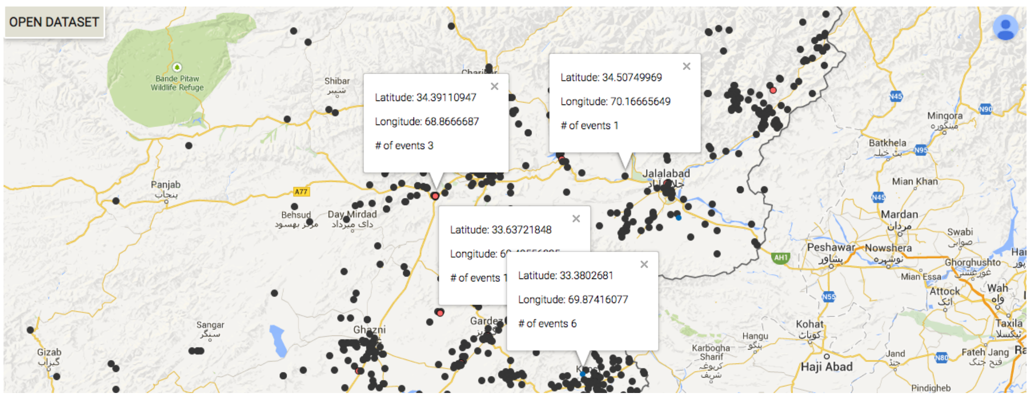


Fig. 3: Interactive Map with Plotted War Data

Figure 3 shows a visualization of the data set on a map. Each war log entry is plotted on the map as a point. Multiple events may happen at the same location, and the color of the points on the map indicates the number of events on the points.

Figure 4 contracts two storygraphs. The left one is the original Storygraph without optimization. It has a number of line crossings and is difficult to read. The one on the right is adjusted with our heuristics. All of the location lines are parallel with no line crossing.

Figure 5 illustrates the synchronization between the Storygraph and map. Locations selected on the map are automatically highlighted on Storygraph. This allows users to quickly relate spatial information with temporal information.

7 Conclusion

We have presented a new storyline visualization method that focuses on presenting the time, location, character, and context dimensions of a narrative. There are two main contributions. The first is a novel visualization technique called Storygraph that presents temporal and spatial information in one view. Storygraph allows users to look for patterns across the temporal and spatial dimensions, something very difficult to do in other

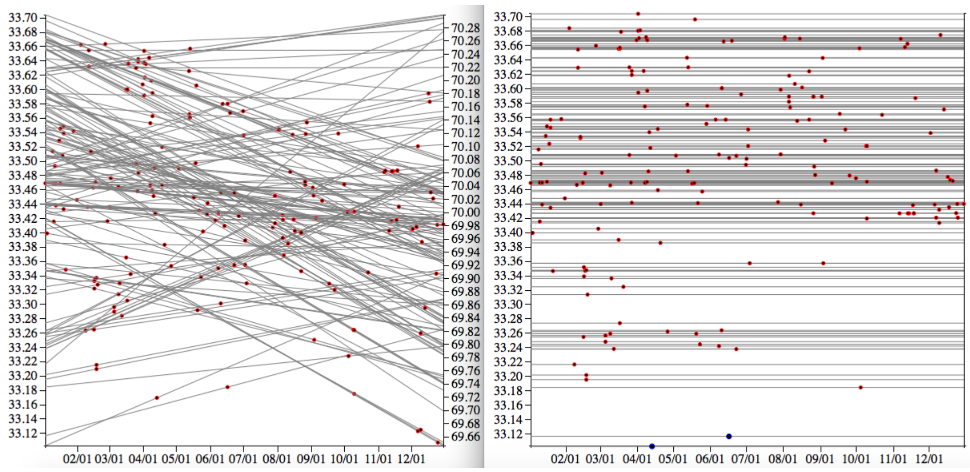


Fig. 4: Storygraph before and after line crossing reduction

types of visualizations. However, Storygraph often suffers from cluttering and excessive line crossings when dealing with big data sets. Our second contribution is a new heuristic method to reduce cluttering and line crossings in Storygraph. This feature, combined with a synchronized map view, makes it much easier for users to conduct spatio-temporal data analysis. The current system works well with structured and semi-structured data. In the future, we plan to continue improving Storygraph with better layout algorithm for spatial data. We are also looking into methods to use unstructured texts as input.

References

1. R. Kosara and J. Mackinlay, "Storytelling: The next step for visualization," *Computer*, no. 5, pp. 44–50, 2013.
2. E. Segel and J. Heer, "Narrative visualization: Telling stories with data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1139–1148, 2010.
3. J. Hullman and N. Diakopoulos, "Visualization rhetoric: Framing effects in narrative visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2231–2240, 2011.
4. S. Havre, B. Hetzler, and L. Nowell, "Themeriver: Visualizing theme changes over time," in *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*. IEEE, 2000, pp. 115–123.
5. D. Fisher, A. Hoff, G. Robertson, and M. Hurst, "Narratives: A visualization to track narrative events as they develop," in *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on*. IEEE, 2008, pp. 115–122.
6. M. Ogawa and K.-L. Ma, "Software evolution storylines," in *Proceedings of the 5th international symposium on Software visualization*. ACM, 2010, pp. 35–42.

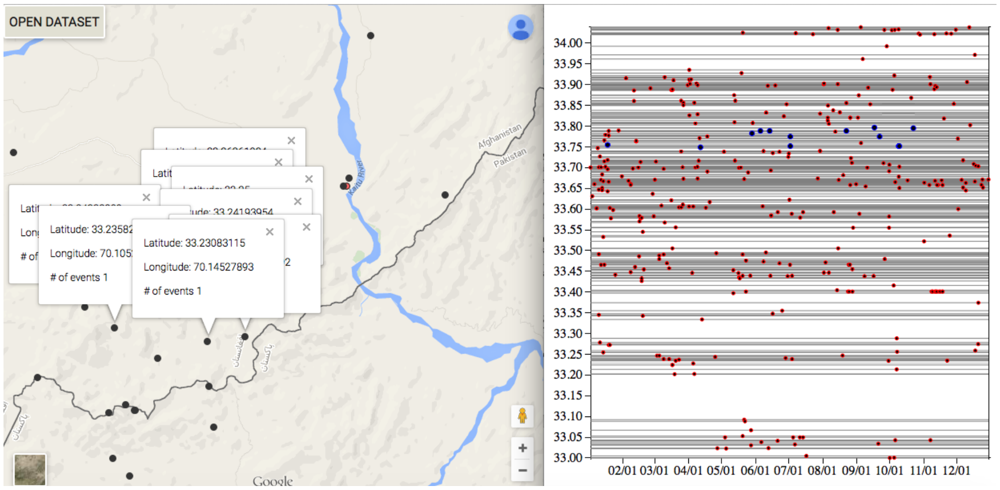


Fig. 5: Interactive Map and Storygraph are sychronized. Users can zoom in or zoom out on the map view and the events shown in the map view are highlighted as blue in Storygraph.

7. N. W. Kim, S. K. Card, and J. Heer, “Tracing genealogical data with timenets,” in *Proceedings of the International Conference on Advanced Visual Interfaces*. ACM, 2010, pp. 241–248.
8. P. H. Nguyen, K. Xu, R. Walker, and B. Wong, “Schemaline: Timeline visualization for sensemaking,” in *Information Visualisation (IV), 2014 18th International Conference on*. IEEE, 2014, pp. 225–233.
9. J. Zhao, F. Chevalier, C. Collins, and R. Balakrishnan, “Facilitating discourse analysis with interactive visualization,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 12, pp. 2639–2648, 2012.
10. S. Liu, J. Pu, Q. Luo, H. Qu, L. M. Ni, and R. Krishnan, “Vait: A visual analytics system for metropolitan transportation,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 1586–1596, 2013.
11. G. Sun, Y. Liu, W. Wu, R. Liang, and H. Qu, “Embedding temporal display into maps for occlusion-free visualization of spatio-temporal data,” in *Pacific Visualization Symposium (PacificVis), 2014 IEEE*. IEEE, 2014, pp. 185–192.
12. H. Liu, Y. Gao, L. Lu, S. Liu, H. Qu, and L. M. Ni, “Visual analysis of route diversity,” in *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*. IEEE, 2011, pp. 171–180.
13. P. Bak, F. Mansmann, H. Janetzko, and D. A. Keim, “Spatiotemporal analysis of sensor logs using growth ring maps,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 913–920, 2009.
14. P. Gatalsky, N. Andrienko, and G. Andrienko, “Interactive analysis of event data using space-time cube,” in *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*. IEEE, 2004, pp. 145–152.
15. G. Andrienko, N. Andrienko, C. Hurter, S. Rinzivillo, and S. Wrobel, “Scalable analysis of movement data for extracting and exploring significant places,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 7, pp. 1078–1094, 2013.

16. T. Kapler and W. Wright, "Geotime information visualization," *Information Visualization*, vol. 4, no. 2, pp. 136–146, 2005.
17. P. O. Kristensson, N. Dahlback, D. Anundi, M. Bjornstad, H. Gillberg, J. Haraldsson, I. Martensson, M. Nordvall, and J. Stahl, "An evaluation of space time cube representation of spatiotemporal patterns," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 4, pp. 696–702, 2009.
18. "Movie narrative charts," http://xkcd.com/657, accessed: 2015-03.
19. T. Crnovrsanin, C. Muelder, C. Correa, and K.-L. Ma, "Proximity-based visualization of movement trace data," in *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. IEEE, 2009, pp. 11–18.
20. Y. Tanahashi and K.-L. Ma, "Design considerations for optimizing storyline visualizations," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 12, pp. 2679–2688, 2012.
21. H. Tanahashi and K.-L. Ma, "An efficient framework for generating storyline visualizations from streaming data," *Visualization and Computer Graphics, IEEE Transactions on*, 2015.
22. S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu, "Storyflow: Tracking the evolution of stories," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 12, pp. 2436–2445, 2013.
23. N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu, "Whisper: Tracing the spatiotemporal process of information diffusion in real time," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 12, pp. 2649–2658, 2012.
24. D. Chu, D. A. Sheets, Y. Zhao, Y. Wu, J. Yang, M. Zheng, and G. Chen, "Visualizing hidden themes of taxi movement with semantic transformation," in *Pacific Visualization Symposium (PacificVis), 2014 IEEE*. IEEE, 2014, pp. 137–144.
25. Z. Wang and X. Yuan, "Urban trajectory timeline visualization," in *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*. IEEE, 2014, pp. 13–18.
26. F. Harary and A. Schwenk, "A new crossing number for bipartite graphs," *Utilitas Math*, vol. 1, no. 203-209, pp. 30–31, 1972.
27. P. Eades and N. C. Wormald, "Edge crossings in drawings of bipartite graphs," *Algorithmica*, vol. 11, no. 4, pp. 379–403, 1994.
28. M. R. Garey and D. S. Johnson, "Crossing number is np-complete," *SIAM Journal on Algebraic Discrete Methods*, vol. 4, no. 3, pp. 312–316, 1983.
29. J. N. Warfield, "Crossing theory and hierarchy mapping," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 7, no. 7, pp. 505–523, 1977.
30. P. Eades and N. C. Wormald, *The median heuristic for drawing 2-layered networks*. University of Queensland, Department of Computer Science, 1986.
31. A. Abdullah, "Data mining using the crossing minimization paradigm," 2007.
32. E. Mäkinen and M. Sieranta, "Genetic algorithms for drawing bipartite graphs," *International Journal of Computer Mathematics*, vol. 53, no. 3-4, pp. 157–166, 1994.
33. S. Khan, M. Bilal, M. Sharif, and F. A. Khan, "A solution to bipartite drawing problem using genetic algorithm," in *Advances in Swarm Intelligence*. Springer, 2011, pp. 530–538.
34. "Afghanistan war logs 2003-2010," http://Guardian.co.uk, accessed: 2015-03.