12-18-2014

Visualizing Spatio-Temporal data

Ayush Shrestha

VISUALIZING SPATIO-TEMPORAL DATA

by

AYUSH SHRESTHA

Under the Direction of Ying Zhu, PhD

ABSTRACT

The amount of spatio-temporal data produced everyday has sky rocketed in the recent years due to the commercial GPS systems and smart devices. Together with this, the need for tools and techniques to analyze this kind of data have also increased. A major task of spatio-temporal data analysis is to discover relationships and patterns among spatially and temporally scattered events. However, most of the existing visualization techniques implement a top-down approach i.e, they require prior knowledge of existing patterns. In this dissertation, I present my novel visualization technique called Storygraph which supports

bottom-up discovery of patterns. Since Storygraph presents and integrated view, analysis of events can be done with losing either of time or spatial contexts. In addition, Storygraph can handle spatio-temporal uncertainty making it ideal for data being extracted from text. In the subsequent chapters, I demonstrate the versatility and the effectiveness of the Storygraph along with case studies from my published works. Finally, I also talk about edge bundling in Storygraph to enhance the aesthetics and improve the readability of Storygraph.

INDEX WORDS: Spatio-temporal data visualizations, Uncertainty visualization, Information visualization

VISUALIZING SPATIO-TEMPORAL DATA

by

AYUSH SHRESTHA

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2014

VISUALIZING SPATIO-TEMPORAL DATA

by

AYUSH SHRESTHA

Committee Chair: Ying Zhu

Committee: Scott Owen

Ben Miller

Yi Zhao

Rajshekhar Sunderraman

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

December 2014

DEDICATION

I would like to dedicate this dissertation to my wife, Kebina, and my family back home, who have all been very supportive throughout these last five years.

ACKNOWLEDGEMENTS

This dissertation work would not have been possible without the support of many people. I want to express my gratitude to my advisor Dr. Zhu for his immense help in providing me with research directions and directing the publications. I would also like to thank Dr. Miller for the datasets and for providing invaluable suggestions on applying Storygraph to other domains like narratives, and Dr Zhao for the brainstorming sessions on the mathematical models. Lastly, I would like to thank my PhD committee for providing me with valuable suggestions on the overall form of this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PART 1

INTRODUCTION

Spatio-temporal datasets include all datasets consisting of location and time. Depending on the source of the data these datasets may or may not contain other attributes. These remaining attributes, in turn, might differ drastically from one dataset to another. In addition, time and location might also differ in precision. For instance, the location could be in form of precise latitude and longitude for real geographic locations (or simply X-Y coordinates for some coordinate space. For simplicity, I will use latitude and longitude in all the subsequent chapters). However, the location can also be expressed as a reference to another location like 'round the corner from Downtown'. These kinds of referential locations are common in datasets extracted from text documents and often times cannot be pinpointed to an exact location. Similar examples can be generated for imprecision in time.

Similar to the diversity in the sources and the precision of these datasets, their use-cases differ as well. A satellite launching system might require data precise to microseconds, while an urban planner may only require data precise to events taking place per day but for a long span of time. Human rights workers trying to analyze patterns of violation may work with even less precise data.

Many domain specific algorithms have been developed for transforming and analyzing this data. However few algorithms, like visualization and visual analytics, spans across these domains. Spatio-temporal data visualizations are specifically geared towards visualizing datasets having location and time. Though these visualization techniques focus on highlighting different aspects of the data, all of these share a common goal of uncovering hidden patterns in the dataset.

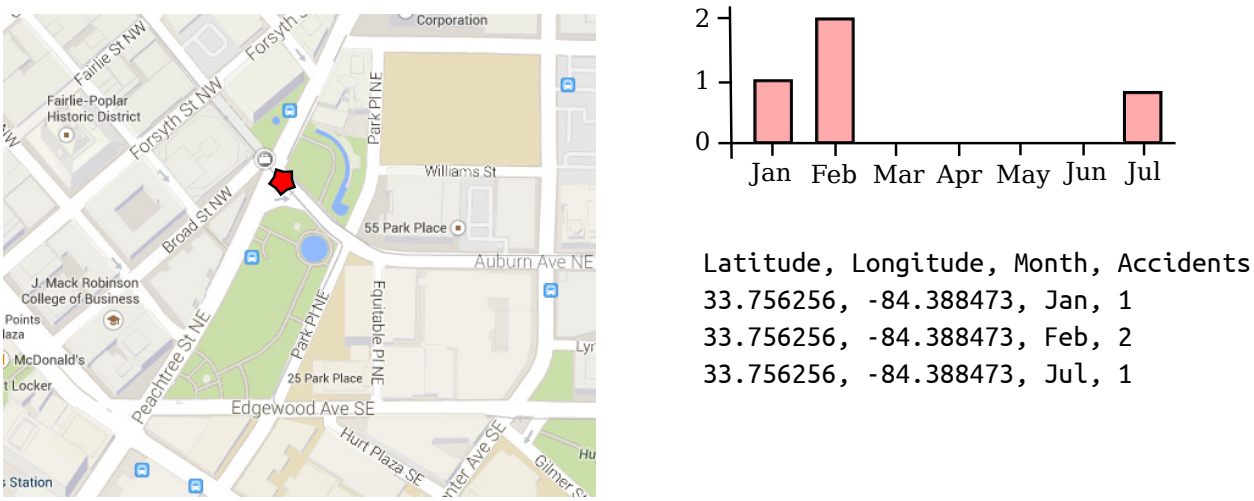


Figure (1.1) Same dataset represented using different visualizations.

Figure 1.1 shows three well established visualizations created using the same spatio-temporal dataset. The left visualization shows a map. The top-right visualization shows a histogram of the data and the bottom-right visualization shows the data in a tabular format. Since maps ignore temporal information, all the events happening at that location during different times and merged into a single point. Similarly the histogram only shows time and the frequency of events but does not show location. The tabular view shows both pieces of the information but the readability of the tables does not scale well with the size of the data. Each of these visualizations have their own strengths and limitations. However, the choice of visualization(s) greatly affect the impression of the viewers regarding the data. It is also important to note that, in this particular figure, three visualizations are presented together. In many real-world scenarios, this might not be possible due to the size of the data or other factors. When one of the dimensions is ignored in the spatio-temporal data, much information is lost resulting in different perspective (imagine just viewing the map). I present state-of-art spatio-temporal visualization techniques in the following Chapter 3 and discuss their advantages and shortcomings.

My work in this dissertation focuses on visualizing time and space using my novel two-dimensional view called *Storygraph*. It is a generic spatio-temporal visualization technique

and is able to visualize both precise and imprecise data.

I begin my presenting a primer on data visualizations in Chapter 2. This chapter gives a brief introduction on what is visualization, and why and how is it useful to present information. Chapter 3 lists the existing work on visualizing spatio-temporal data. In this chapter, I analyze the state of art works visualizing time, space, and both space and time.

I demonstrate the effectiveness of the Storygraph with the case studies in Chapter 5. These case studies are from my published work and include interesting patterns that were previously undocumented in any known formal reports. In the subsequent two chapters, Chapter 6 and 7 I apply Storygraph to other domains like Open Source Softwares and Computer Networks. I also discuss preprocessing steps required for the data arriving from these sources and how they can be transformed and fed into Storygraph. Chapter 9 is another effort to extend storygraph and make it more readable, especially when the size of the data becomes bigger.

Finally I conclude in Chapter 10 with my findings and also state some possible future directions for Storygraph.

PART 2

DATA VISUALIZATION

2.1 What is visualization?

Visualization can be defined as the visual representation of the data using algorithms. It is important to note the term "using algorithms" in the definition since it distinguishes visualizations in computer science from other disciplines. More importantly, visualizations created by using algorithms ensures reproducibility: the same visualization can be obtained irrespective of the user, platform or the choice of programming language as long as the underlying data does not change and the steps in the algorithm are followed precisely.

Often, visualization is seen as a process of generating pretty pictures from the datasets. This is a myth. Good data visualizations provide great insights on the data. One fitting example in this case is Anscombeś Quartet. Anscombeś quartet consists of four datasets having nearly identical properties but they appear differently when visualized. Wikipedia [1] describes the quartet as,

> *"Anscombeś quartet comprises four datasets that have nearly identical statistical properties, yet appear very different when graphed. Each dataset consists of eleven (x, y) points. They were constructed in 1973 by the statistician Francis Anscombe to demonstrate both the importance of graphing data before analyzing it and the effect of outliers on statistical properties. "*

Table 2.1 shows the Anscombe's Quartet with four sets of I - IV each containing 11 x and a y values. In each of these sets,

- mean of x and y, $\mu_x = 9$ and $\mu_y = 7.50$
- variance, $\sigma_x^2 = 11$ and $\sigma_y^2 = 4.1$
- linear regression in each case is $y = 3.00 + 5.00x$

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Table (2.1) Anscombeś Quartet data points

If one were to just look at the statistics, they would probably conclude that the points points must be similar. But as soon as the x-y values are plotted on the graph as shown in Figure 2.1, it reveals a totally different picture of the underlying datasets. Anscombe's Quartet is one of many examples that highlights the importance of visualization.

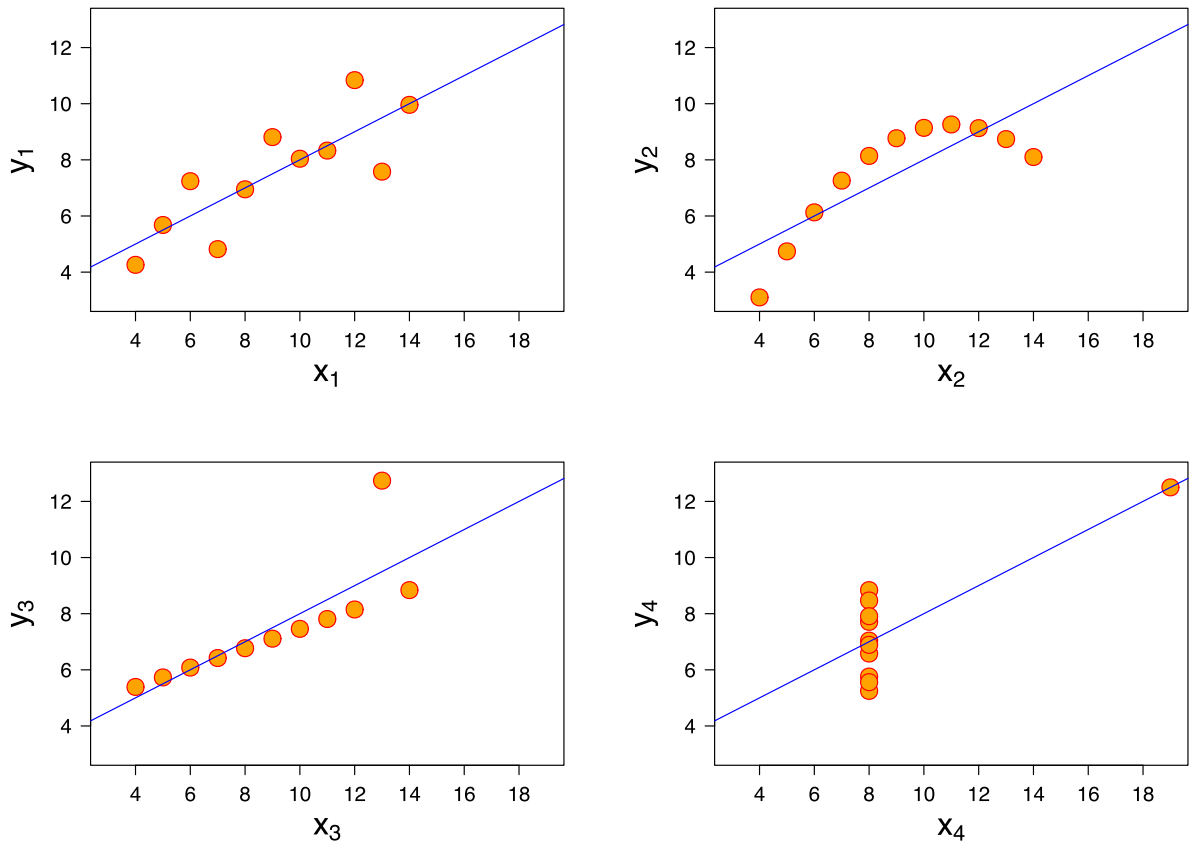


Figure (2.1) Figure adapted from Wikipedia page on Anscombeś Quartet showing the scatter plots for all four datasets. Even though the all four sets are statistically really identical, their graphs really differ from each other.

Besides revealing hidden patterns in the data, visualizations also often aid in accumulating information unconciously from the data or pre-attentive processing [2]. Few pre-attentive attributes include position, orientation, scale, color, brightness, saturation etc. A simple example of pre-attentive processing is shown in Figure 2.2. The subfigure in the left consists of blue squares acting as distractors and on the right, one blue square has been changed to red. Studies [2] have shown that the information that the red square is present in the figure is obtained as soon as one looks at the image as opposed to brain having to process and identify the target. This adds to the value of visualizations.

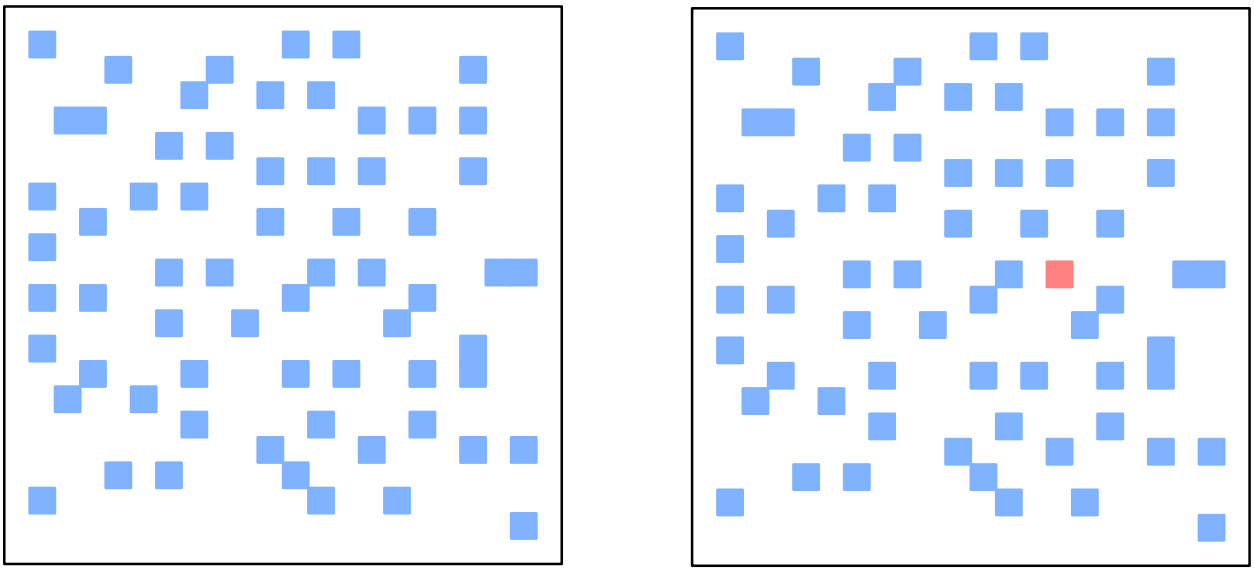


Figure (2.2) Simple example of pre-attentive processing: searching for a target square with a different hue. Left: The target is absent, Right: the target is present.

Figure 2.3 shows a simple example of information visualization involving pre-attentive learning.

1 4 1 5 9 2 6 5 3 5
8 9 7 9 3 2 3 8 4 6
2 6 4 3 3 8 3 2 7 9
5 0 2 8 8 4 1 9 7 1
6 9 3 9 9 3 7 5 1 0

1 4 1 5 9 2 6 5 3 5
8 9 7 9 3 2 3 8 4 6
2 6 4 3 3 8 3 2 7 9
5 0 2 8 8 4 1 9 7 1
6 9 3 9 9 3 7 5 1 0

Figure (2.3) How many 9 are there in 100 Decimal places of π? Left: the 100 decimal places of π listed. Right: 100 decimal places of π with all 9 highlighted so that it assists in pre-attentive learning. There are eight 9 in 100 decimal places of π.

2.2 Information Visualization

Data visualization can be broadly classified into Information visualization and Scientific visualization. Scientific visualization deals with data coming from natural and physical sci-

ences. The visualization work deals more with creating meshes and simulating how different phenomenons work. An example of scientific visualization would be creating a 3D model of a volcano in Geology or creating 3D mechanical model of airplanes to test it in a virtual wind tunnel in Aerospce engineering. Information visualization on the other hand deals more with the data coming from social sciences. A great example of information visualization is the Naepolean's march to Russia in the eighteenth century, drawn by Minard [3] as shown in Figure 2.4. From the figure it can be seen that the army decreased to half when Naepolean reached Russia and continued to decrease while returning. The line chart at the bottom, showing temperature, correlates the death of the soldiers to the temperature. This visualization is impressive since it portrays multiple dimensions on a single chart. The dimensions being size of the army, location and temperature.

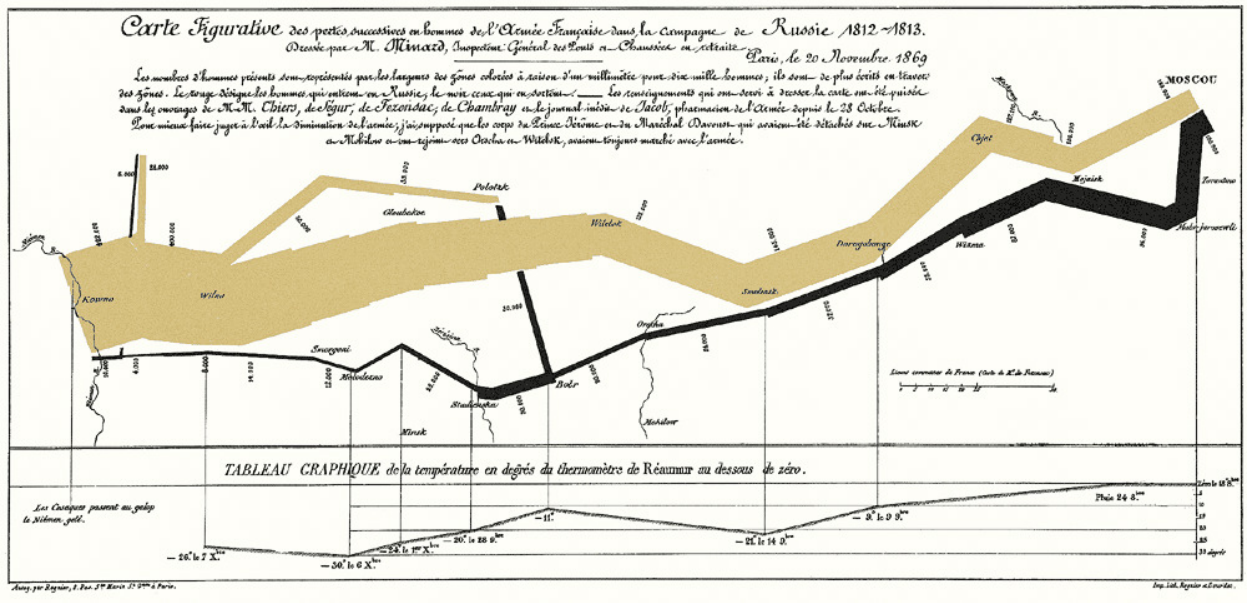


Figure (2.4) Naepolean's march to Russia by Minard (adapted from Tufte[3]. The pale yellow polyline denotes the march towards Russia and the black polyline denotes the march from Russia. The width of the lines corresponds to the size of the army at that point. The bottom line graph shows the temperature (inverse).

Figure (2.5) Examples of 2D visualizations adapted from Google Chart Examples [4].

Information visualization is a broad area which further branches different categories; 2D visualization, 3D visualization, color theory being the top ones. 2D visualizations, as the name states, spans along 2 axes. Hence they are planar. Examples of 2D visualizations include bar charts, pie charts, line charts, etc as shown in Figure 2.5. Maps and tables are other great examples of 2D visualization.

Many times, the dimension of the data exceeds two dimensions. In such cases, either different visual attributes like size, shape and color might be changed or 3D visualizations might be used. The data from the GPS device that we use in the vehicle would be a nice

example to visualize in 3D since it collects the latitude and longitude, which paired with maps provides the altitude information. Although 3D visualizations make it more convenient to plot high dimensional data, it does come with a price. 3D visualizations commonly suffer from occlusion, glyph cluttering and overplotting. These are generic 3D visualization problems which does not limit to information visualization. A clever way to over come these problems include adding user interactivity to the visualizations. These include zooming, panning and filtering. Zooming refers to enlarging or contracting different areas of the visualization. Panning refers to moving the objects within the visualization so that they can be viewed from different angles. Filtering refers to the selection of relevant data. Google Earth [5] is a good example of a 3D visualization which offers all these levels of interactivity. Color theory deals with choosing the proper color so as to enhance the readability or assist the visual analysis of data. It also aids in the pre-attentive learning as in case of Figure 2.3.

PART 3

BACKGROUND AND RELATED WORK

In this chapter, I present a literature review of the existing spatio-temporal data visualization techniques. I begin with discussing visualization techniques that only focus on temporal data, followed by techniques only focusing on spatial data. Later, I move on to techniques that present spatio temporal data classifying them based on their presentation - multiple views, isosurfaces, and hybrid and outlying techniques.

3.1 Visualizing time

Time is generally visualized using two temporal primitives - time point or time interval [6]. Different types of questions can be answered depending on the use of these primitives. Time points are limited to answering questions like whether two events took place at the same time or whether one even took place before the other. Similarly, time intervals are useful for answering questions like whether the events started/ended together, whether events overlapped in time including all the questions that time points could answer. The theoretical and practical approaches to modeling time has been studied in literatures [7][8].

A common goal, irrespective of the choice of primitives, in time visualization is to compare data at different points in time and observe the changes. To achieve this goal, besides the choice of data, other factors that affect the visualizations include whether the data is real or abstract, whether the time is portrayed linearly or non-linearly (including cyclically) and whether the interactivity is required [9]. Figure 3.1 adapted from Aigner's study [9] show different techniques of visualizing time oriented data.

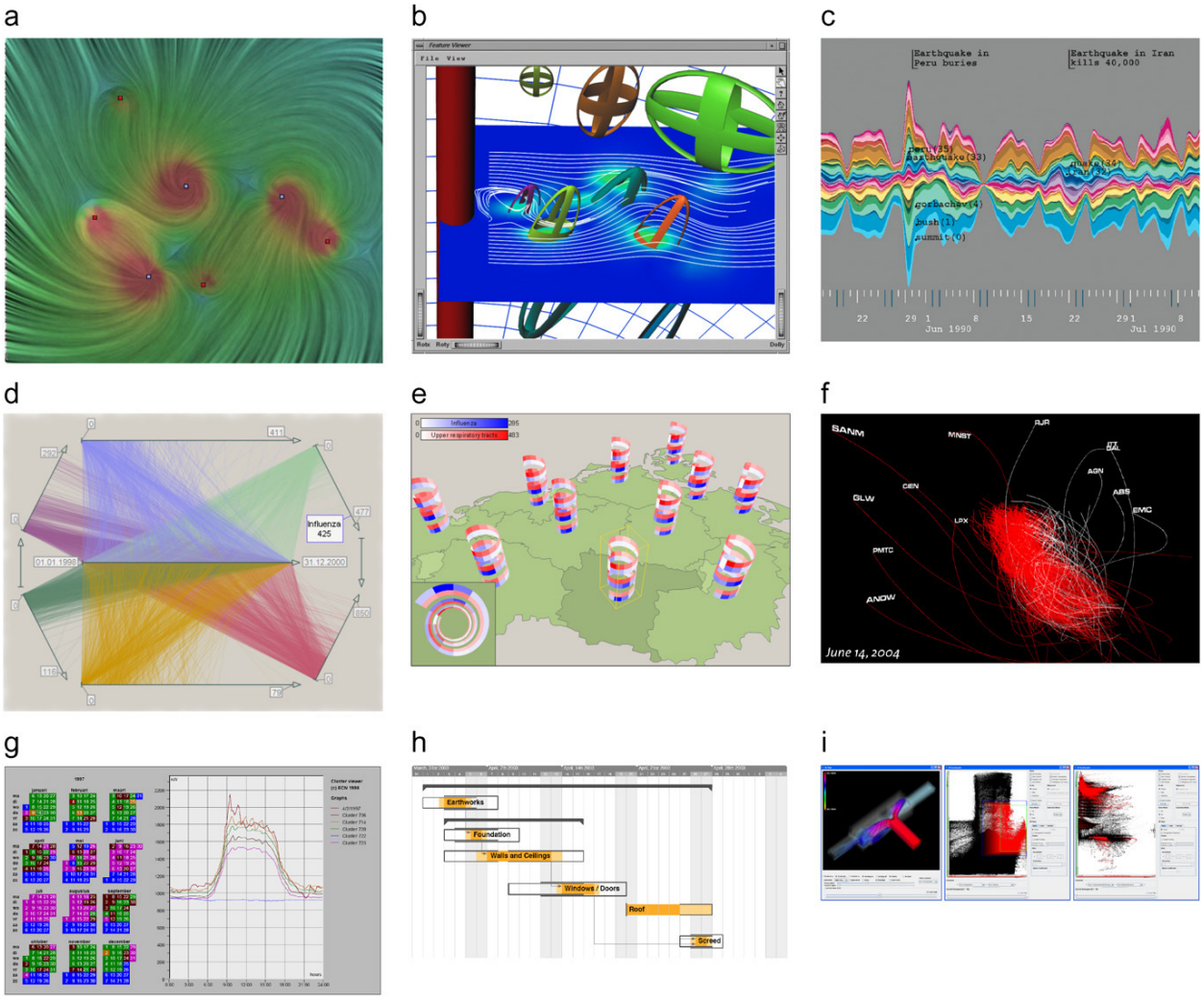


Figure (3.1) Figure adapted from Aigner et al [6] showing different time visualization techniques. The authors have described and categorized the figures (in their own words) as: (a) animated flow visualization[10]: smooth animations created from streamline images; (b) feature and event based flow visualization [11]: animated visualization based on data abstraction and iconic representations; (c) ThemeRiver [12]: static representation of thematic changes in document collections; (d) TimeWheel [13]: axes-based visualization of multivariate data with focus on temporal dependencies; (e) helix glyphs on maps [14] emphasis of cyclic patterns in spatiotemporal human health data; (f) flocking boids [15] stock market visualization based on simulation and animation of flocking behavior; (g) cluster and calendar based visualization [16]: visualization of univariate time series on different levels of aggregation; (h) PlanningLines [17]: visualization of project plans with temporal uncertainty;(i) SimVis [18]: larger system that combines several views to facilitate flow visualization

Similarly, www.timeviz.net [19] by Aigner et al. hosts a comprehensive survey of techniques used for visualizing time as shown in Figure 3.2 including details of some notable visualizations like ThemeRiver[12] and Flocking Boids [15] . Along with the visualizations, the site also allow users to filter the visualizations according to the aforementioned categories.

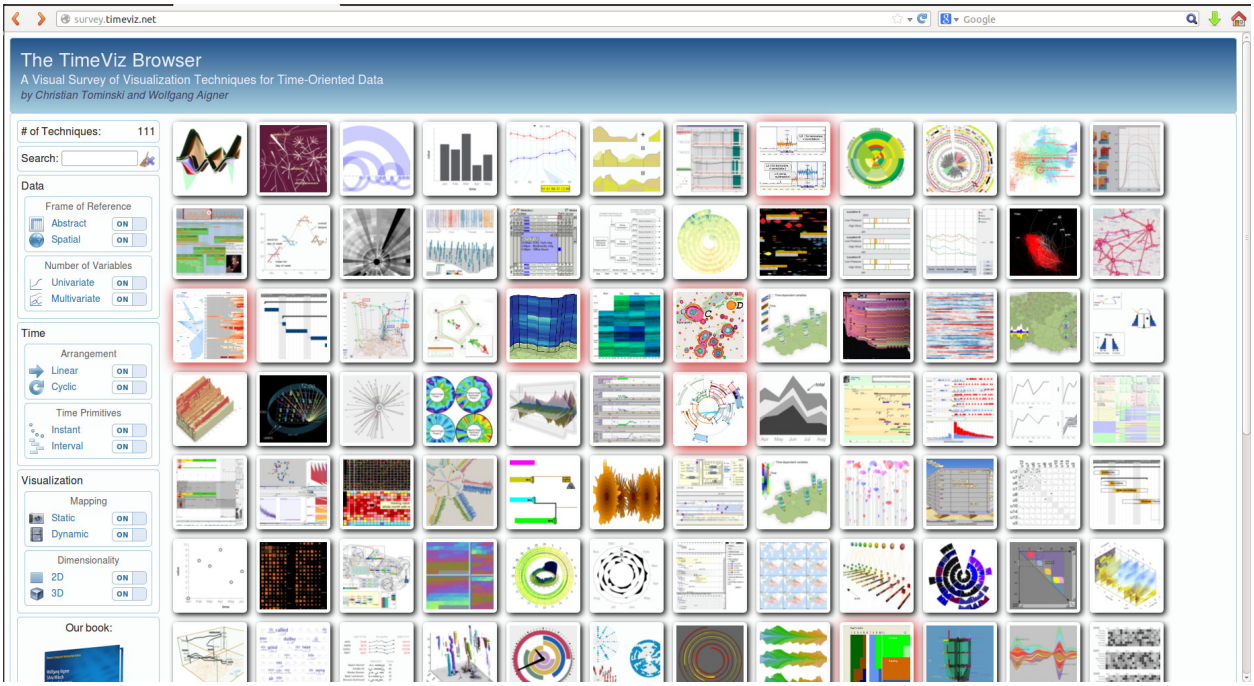


Figure (3.2) Screenshot showing a collection different techniques to visualize time, listed at timeviz.net.

Few other related works not mentioned in the website include Narratives[20], EventRiver[21], Activitree [22], KronoMiner [23, 24], CloudLines[25], SchemaLine [26]. Narratives by Fisher et al. and EventRiver by Luo focus more on how topics in different corpora evolved over time. The output of Narratives is a line chart with the frequency of topics within documents on the vertical axis and time on the horizontal. EventRiver and Cloudlines as shown in Figure 3.3 and Figure 3.4, use similar axes as the Narratives but the each topic is displayed as a bubble with the width corresponding to the time frame span and the height corresponding to the document frequency. However, EventRiver implements a bag of words approach and then predetermines the height, while Cloudlines calculate the point overlaps and increases

the height of the bubble correspondingly.

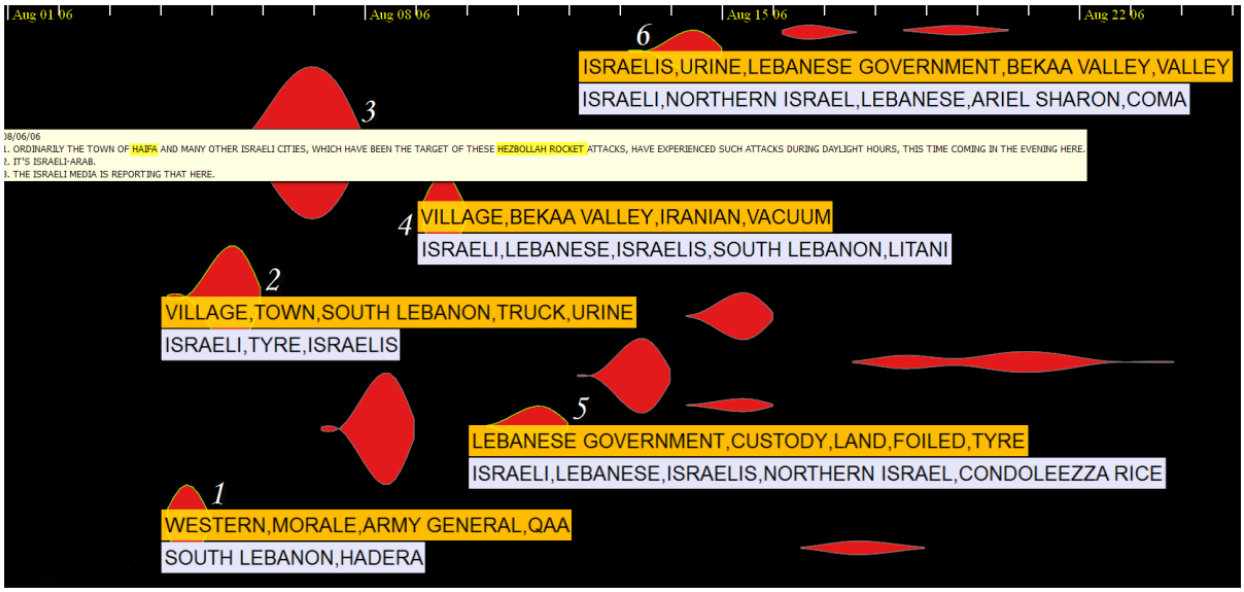


Figure (3.3) EventRiver generated from 2006 Lebanon War (adapted from [21]).

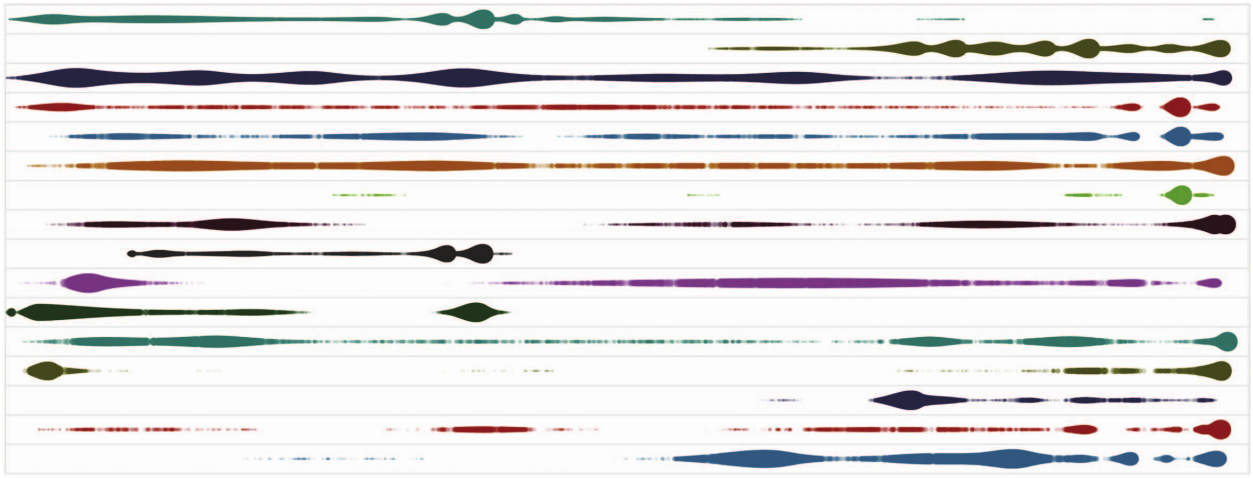
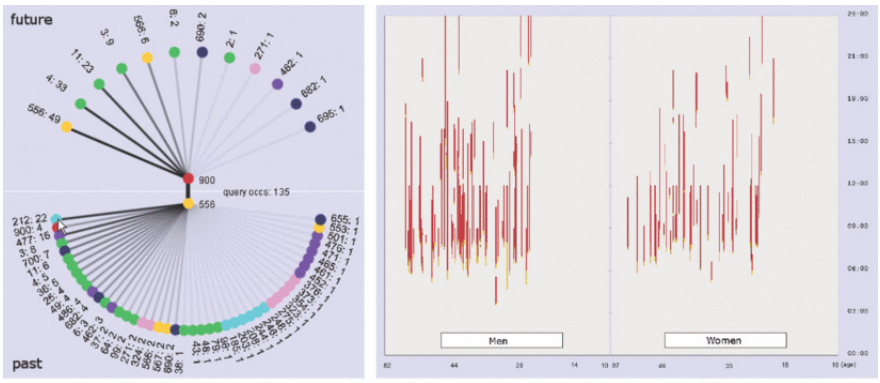


Figure (3.4) CloudLines visualization of key politicians appearing in the news during February 2011 (adapted from [25]).

ActiviTree generates a graph where each event is represented as vertex. Events occurring linearly are placed adjacently and directed edges are drawn between them to show the

sequence as shown in Figure 3.5. Similarly, the opacity of the angles are changed to reflect the frequency in that category.



(a) The 'ActiviTree' visual interface. (b) Linked view showing a sequence in the context of the individuals' daily life.

Figure (3.5) Activity tree of query *Travel by car to work* with a linked view (adapted from [22]).

KronoMiner, is a radial visualization like ActiviTree, that shows events as segments of ring where events taking place at the same time are displayed as concentric segments. Figure 3.6 shows a visualization created from stock market dataset using KronoMiner. Kronominer, also consists of linked views to provide more information regarding the data.

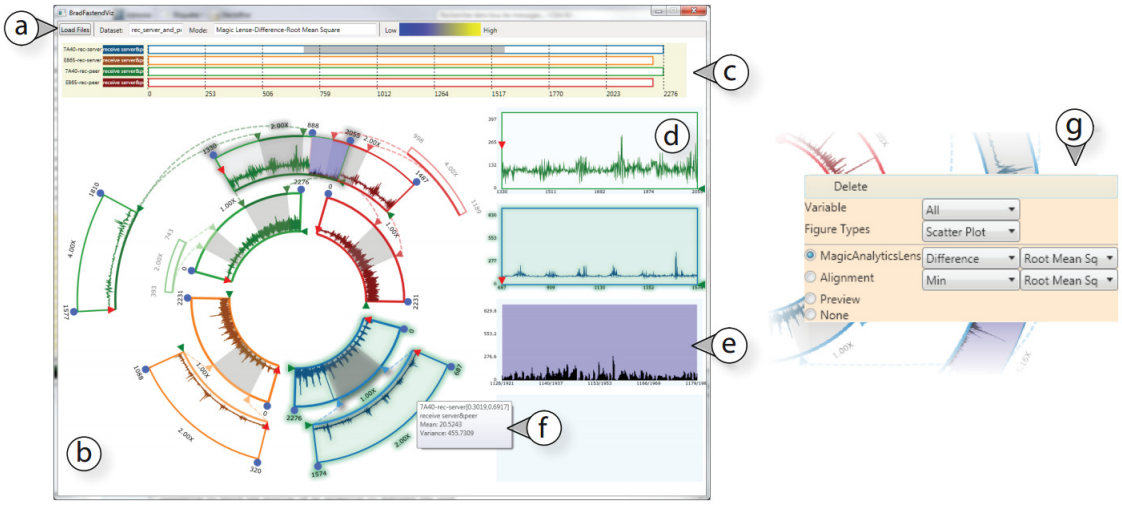


Figure (3.6) Exploration of four stock market datasets using KronoMiner (adapted from [23]).

SchemaLine is a recent work by Phong et al. which generates visualizations close to timelines as shown in Figure 3.7. SchemaLines however, group the tasks into schema and use their layout algorithms to visualize these schema together. Further their layout algorithm also allows dynamic addition of other schema.

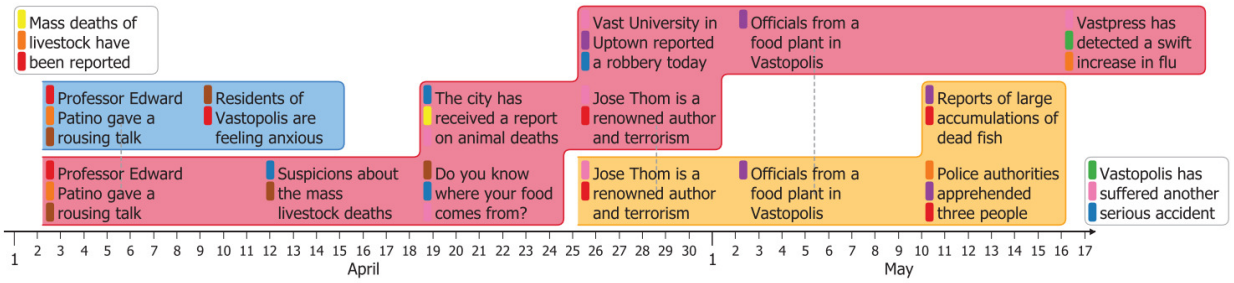


Figure (3.7) SchemaLine (adapted from [26]).

3.2 Visualizing space

Visualizing physical space has always been dominated by maps. Maps are theme based graphic representation of spatial concepts [27]. The earliest known maps in form of Babylonian clay tablets dates back to 2300 B.C. Over the centuries, the developments in cartography continued to transform the representation of maps. During this period, statistics and paper maps were the most prominent tools to study the geospatial data. One of the prominent technique developed during this time is Hagerstrand's time-geography [28]. The development of Geographic Information System (GIS) in 1960's [29] opened up new opportunities for spatial data analysis and also transformed the concept of maps from presentation device to interactive tools to explore the spatial data [30, 31, 32, 33]. Following the advent of the internet, many online mapping tools like Google Maps [34], Google Earth[5], Bing Maps[35], Open Street Maps[36] etc have been developed. Google Maps and Bing maps also provide actual satellite images (not realtime) as layers while Open Street maps are only limited to vector maps with outlines. In addition to satellite maps, Google Earth also provides actual 3D views of the buildings and also support animations via scripts. All these mapping applications allow users to search, zoom, pan and filter locations, out of the box. These

functionalities can be further extended via scripts to fulfill the needs of the user.

3.3 Visualizing spatio-temporal data

The techniques in spatio-temporal data visualization can be classified into Multiple views, isosurfaces, and animations. Multiple views show change in a certain parameter at the same location with time. Generally each view corresponds to the state of that parameter at a given time. Isosurface techniques use map in the X-Y axis and the time in Z-axis. Animations show the change in the parameter at a location with time. In this section, I categorize the works in this same order. However, since Storygraph lies somewhere between multiple views and isosurfaces, I will only discuss techniques implementing multiple views, isosurfaces and hybrid/outlying techniques.

3.3.1 Multiple views

Many visualization methods present time and space in separate views, vast majority of them employ Tufte's 'small multiples' [37] technique - having multiple snapshots of a certain parameter in time. In [38] Powsner and Tufte use this technique to show the medical status of a patient over time as shown in Figure 3.8. In this case, each view represents one parameter changing with time. However, this technique can also be used, as in coplots by Cleveland [39] and its extension, comaps by Brunsdon [40, 41, 42] where the views represent change in one single entity. Figure 3.9 shows the comap generated from the rainfall distribution over time in Scotland.

Other recent extensions and applications of coplot and comap include work by Asgary et al. [43] and Plug et al.[44]. Asgary presents multiple view for both time and space for his case study of fire incidents and causes in Toronto, Canada as shown in Figures 3.10. They use comaps as in Brunsdon's work to display changes in time. Same techniques have been applied by Plug for crash analysis.

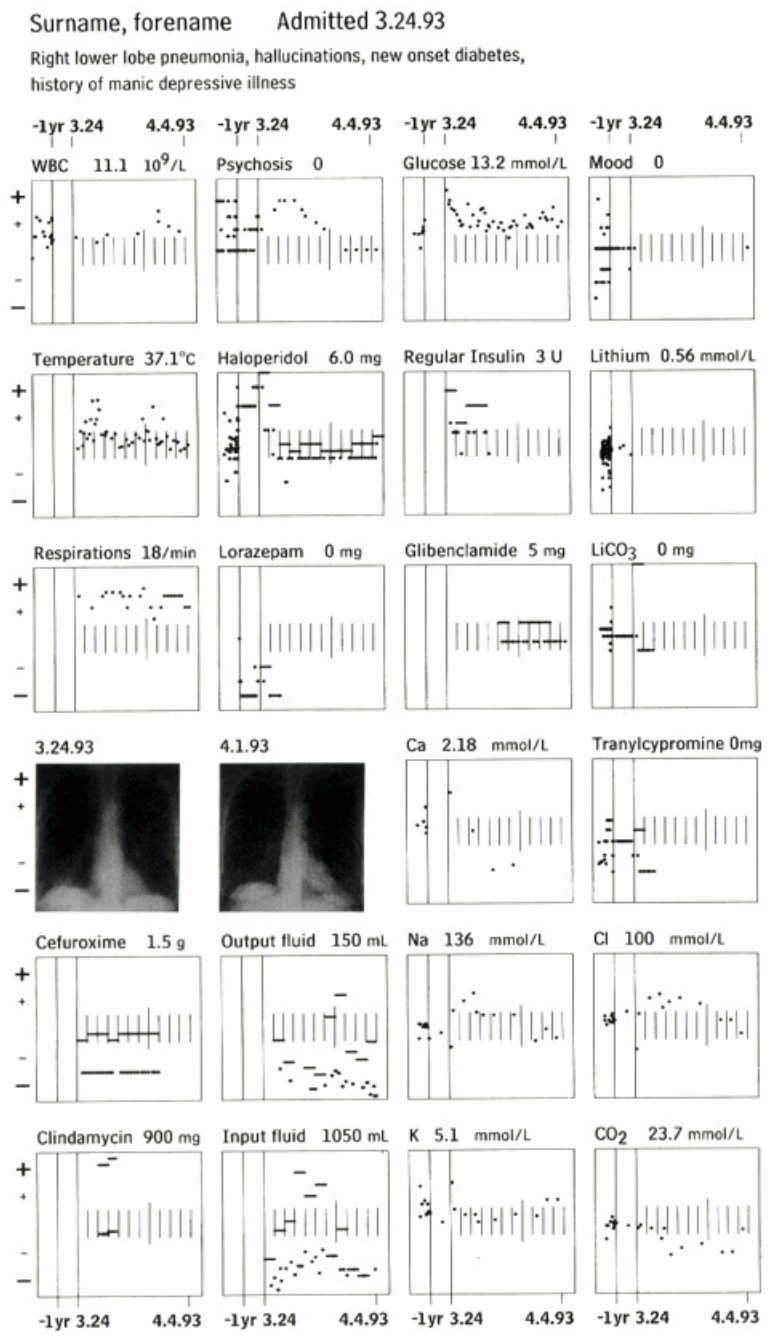


Figure (3.8) Example of Small multiples technique: one chart for each parameter showing changes in time. (adapted from [38]).

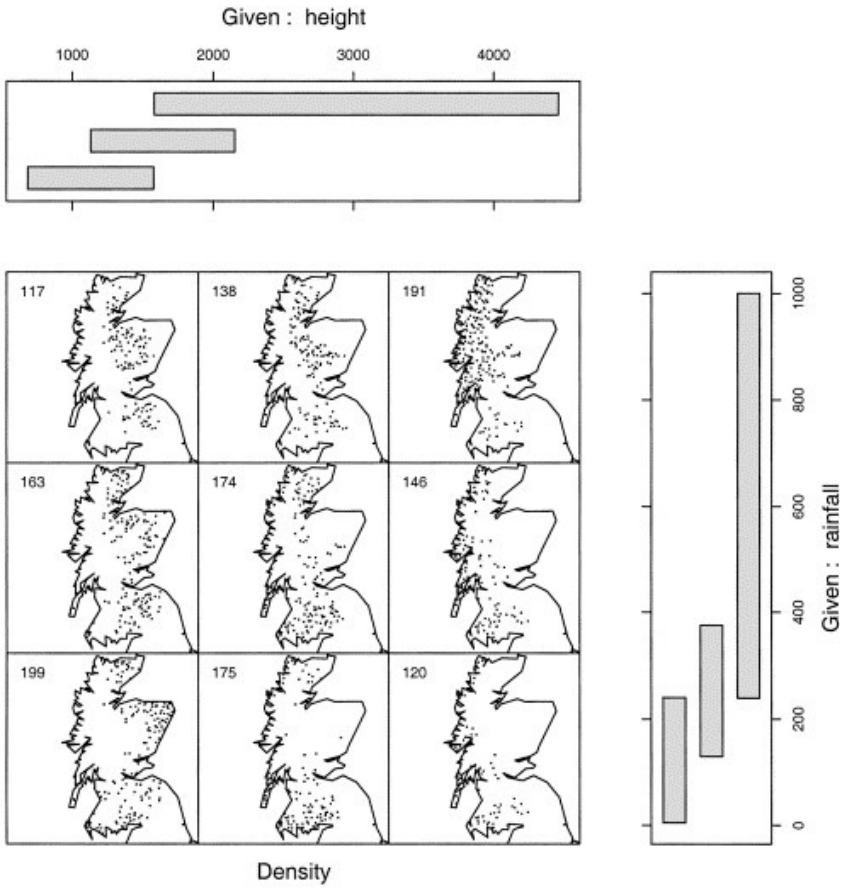


Figure (3.9) Brunsdon's comap showing rainfall in Scotland (adapted from [40]).

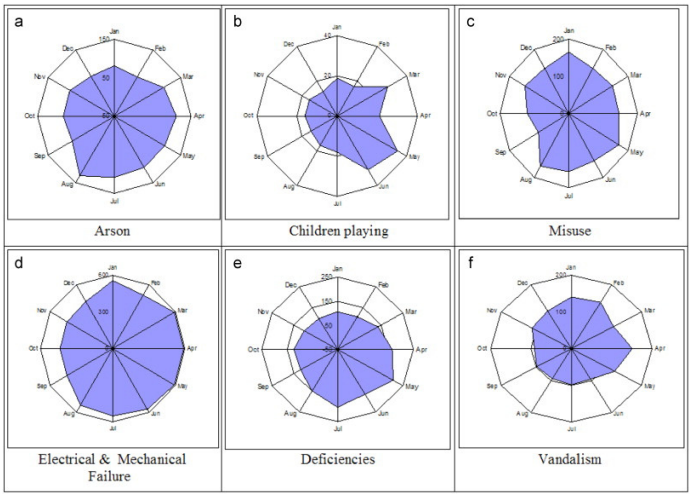


Figure (3.10) Multiple radial views showing the changes in different parameters with time of day (adapted from [43]).

In [45], Jern et al. implement parallel coordinates together with series of heatmaps to show variations over time as shown in Figure 3.12. The parallel coordinates are used to visualize other parameters within the set. Others [46] used a Gantt chart to link temporal data with spatial data. In this case, each small map view corresponds to a bar in the Gantt chart. [46] [47] used an interactive synchronized views. When the user clicks on a data points in the temporal view, the corresponding point in the spatial view is automatically highlighted, and vice versa.

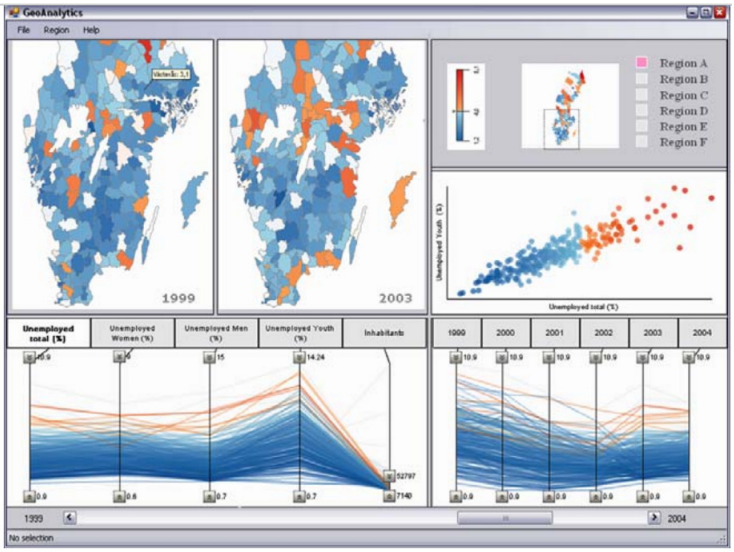


Figure (3.11) Jern's work with parallel coordinates, each axis representing time. (adapted from [45]).

3.3.2 Isosurface

Isosurfaces has also been a popular method of visualizing spatio-temporal data since it inherently supports three dimensions (X-Y to represent Lat-Long and Z to represent time). Literature in this area is dominated by space time cubes and its variants [48, 49, 50]. An example of space time cube is shown in Figure 3.13. The size and the shape of the dots represent additional data attributes. In the space-time cube method by Gatalsky [50], a synchronized 3D time line is plotted above a 2D map. Similarly, Tominski et al. [51] placed 3D icons on a 2D map to visualize events that happen on the same location over a period

of time as shown in Figure 3.14. Survey by Andrienko el al. [49] discusses the existing techniques and the queries these visualizations support.

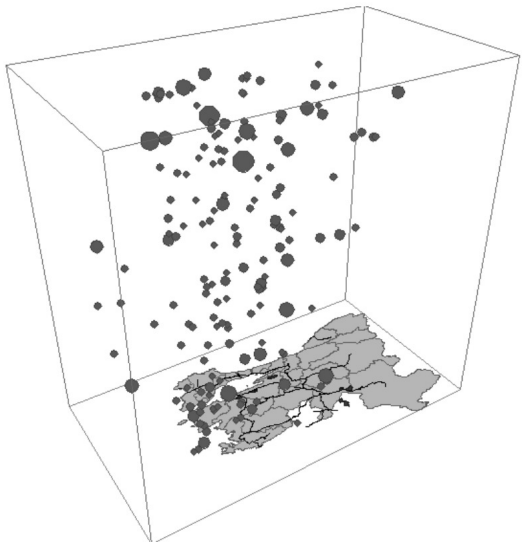


Figure (3.12) Space-time cube. (adapted from [49]).

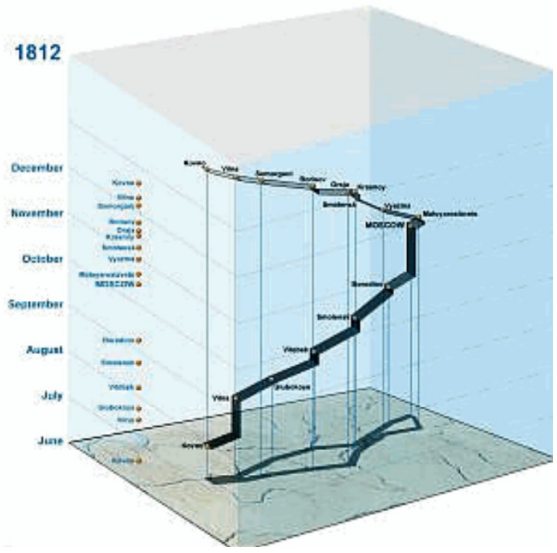


Figure (3.13) Space-time cube. (adapted from [50]).

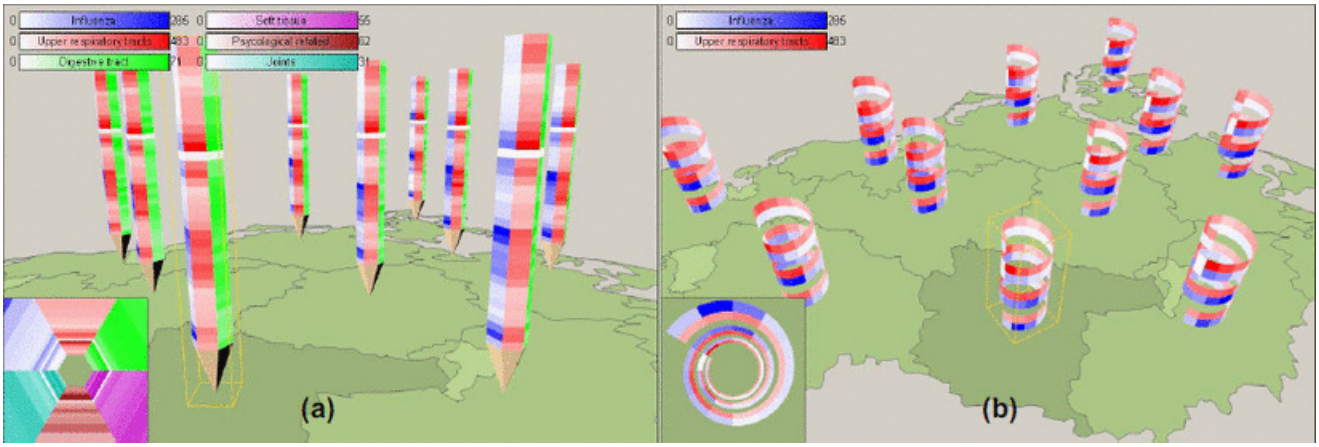
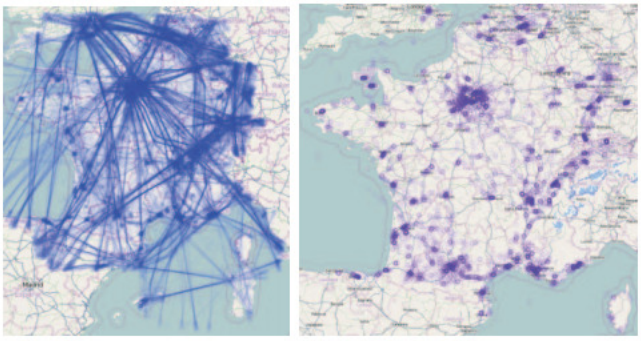


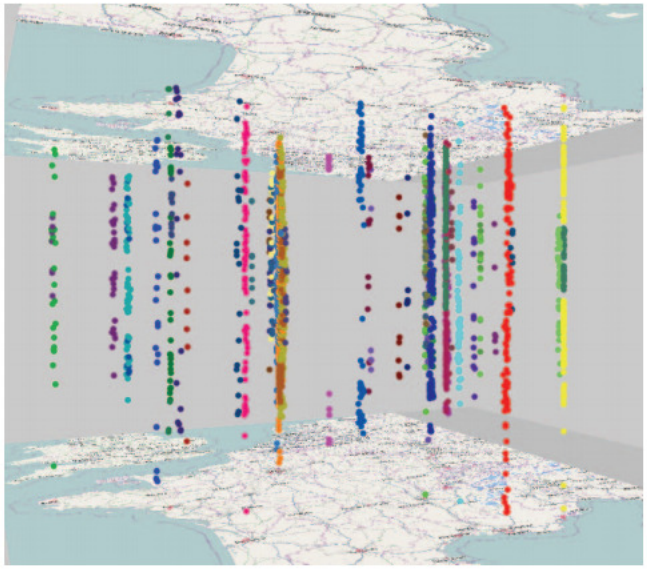
Figure (3.14) Space-time cube. (adapted from [50]).

3.3.3 Hybrid/Outlying techniques

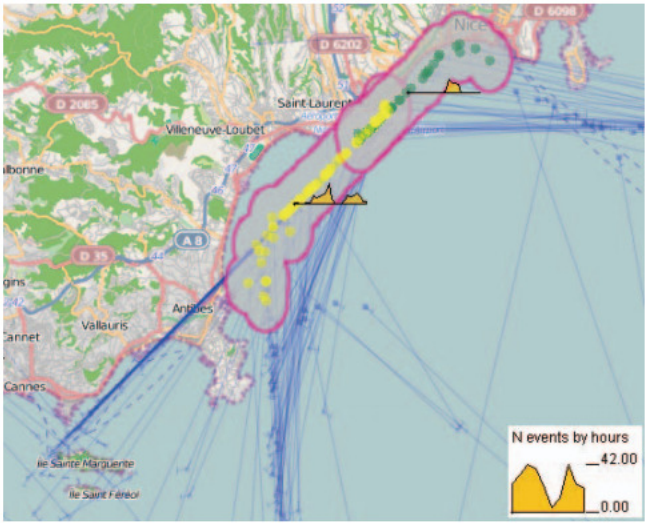
In the recent years, many authors have developed techniques that combine isosurfaces with multiple views to provide supplementary information to the users [52, 53]. Figure 3.15 shows multiple views including isosurfaces from Andrienko's work [52] where the authors are trying to visualize the traffic patterns in the city. Similarly, Figure 3.16 shows the parallel coordinates together with isosurface, adapted from Landesberger's recent work [53]. Landesberger et al. were trying to analyze the cell phone calls using parallel coordinates (top figure) and movement patterns and using the isosurfaces in the bottom figure.



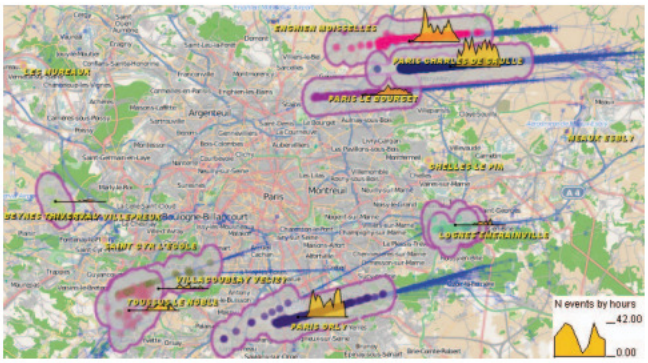
The trajectories are drawn with 1% opacity (left). The positions of the landing events extracted from the flight data are drawn with 50% opacity (right).



The space-time cube shows the landing events clustered by spatial positions and directions.

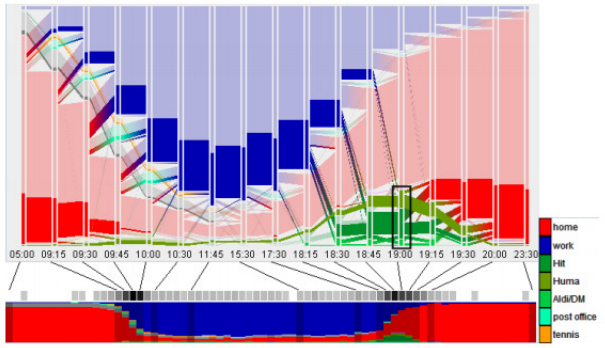


The yellow and green dots represent two SD-clusters of landings in the airport of Nice. The time diagrams show the dynamics of the landings from two directions.

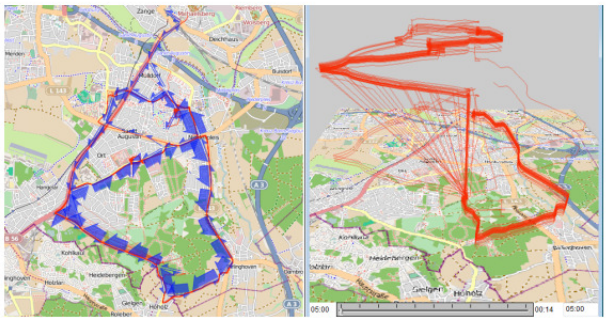


The time diagrams show the dynamics of landings in the airports of Paris.

Figure (3.15) Hybrid views (adapted from [52]).



(a) Dynamic categorical data view with highlighting (DCDV)



(b) Linked geographic views

Figure (3.16) Parallel coordinates and linked isosurface views (adapted from [53]).

Excluding parallel coordinates, there are only few outlying techniques to visualize spatio-temporal data. Recent work by Kristensson [54], which falls in this category, is shown Figure 3.17. The authors use labels to mark the time over a map, different colors representing different people. Another interesting visualization is Growth Ring Maps by Bak et al. [55] as shown in Figure 3.18. Their work is related to analyzing the number of visitors moving in three floors using sensors and visualizing the logs. The size rings in the figure directly correspond to the number of visitors in that place and the color of the visitor correspond to the floors in the building. Thudt et al. present map-timeline views [56] as shown in Figure 3.19, where the timeline shows the amount of spent at a particular place - longer time resulting in a bigger circle.

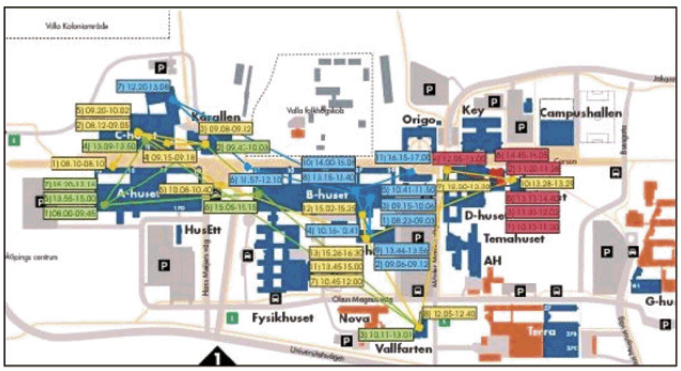


Figure (3.17) Labels over the map, show time (adapted from [54]).

Figure (3.18) Growth ring maps. (adapted from [55]).

Figure (3.19) Map-timeline view. (adapted from [56]).

My work, Storygraph, lies in this area. Compared to multiple views, since Storygraph is a single view chart, it reduces the cognitive load on the viewers. In addition since many statistical estimators are used for preventing the generation of too many views, multiple views often fail to present subtle temporal changes. Many authors have used interactive multiple views, however, they still fail to represent continuous correlation between spatial and temporal data. Similarly, comparing storygraph to isosurfaces, the benefits of using isosurface is that the time graph do not occlude the 2D map. If there are many data points at one location, there is a third dimension to accommodate the icons. Despite of this feature, it's difficult to align time data with location in 3D. It's hard to read and compare the data value in the vertical dimension. These are the inherent problems of 3D data visualization.

PART 4

STORYGRAPH

4.1 Model

Storygraph is a $2D$ diagram consisting of two parallel vertical axes $V_{latitude} \subset \Re$ and $V_{longitude} \subset \Re$ and an orthogonal horizontal axis $T \subset \Re$. All three of the axes, as in Cartesian graphs, are unbounded at both ends. The values in the axes are ordered in ascending order: from left to right in horizontal axis and bottom to top in vertical axes. The vertical axes $V_{latitude}$ and $V_{longitude}$ represent geographic *latitude* and *longitude* or Northing and Easting on a map. They can also be generalized to represent x and y coordinates of a point in a Euclidean plane. The horizontal axis, T, represents time. Thus a point plotted on Storygraph, which shall be referred to as *event* in the rest of the proposal will have at least three dimensions: latitude, longitude and timestamp.

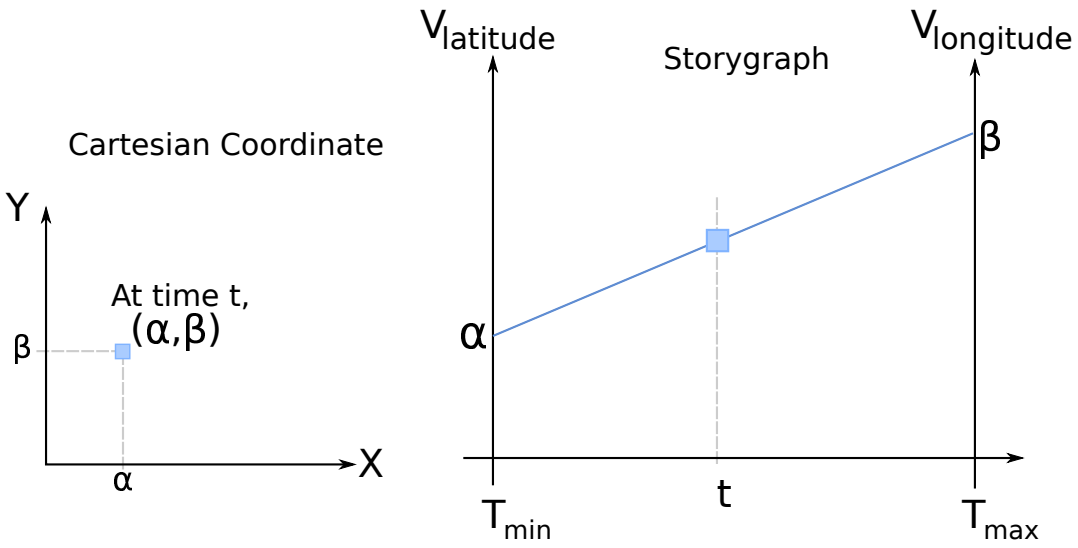


Figure (4.1) Left: A point in the Cartesian coordinate system such as a location on a map at time t, Right: Same point represented in a Storygraph.

For any event occurring at (α, β) in time t as shown in Figure 4.1, our algorithm first draws a *location line* by connecting the points on the two axes, $\alpha \in V_{latitude}$ and $\beta \in V_{longitude}$. The algorithm then returns the point on this line at time t.

The function $f(\alpha, \beta, t) \rightarrow (x_{storygraph}, y_{storygraph})$ which maps an event to the 2D Storygraph plane can be formally written as follows:

$$y_{storygraph} = \frac{(\beta - \alpha)(x_{storygraph} - T_{min})}{T_{max} - T_{min}} + \alpha \tag{4.1}$$

$$x_{storygraph} = t \tag{4.2}$$

where T_{min} and T_{max} are the maximum and minimum timestamps within the dataset.

Figure 4.1 illustrates how a location on a regular 2D map, coded with a Cartesian coordinate, is presented in the Storygraph. Equation 7.1 and 6.2 is used to convert a location on a regular map to a Storygraph plane, and vice versa.

Assuming $T_{min} = 0$ and $T_{max} = T$ without loss of generality, (7.1) simplifies to,

$$y_{storygraph} = \frac{(\beta - \alpha)(x_{storygraph})}{T} + \alpha \tag{4.3}$$

Lemma 4.1.1 *A right rectangular region at time t on a Euclidean plane can be represented by a vertical line segment on the Storygraph at t.*

Proof Given a right rectangular region bounded by four corners $(\alpha 1, \beta 1)$, $(\alpha 1, \beta 2)$, $(\alpha 2, \beta 2)$ and $(\alpha 2, \beta 1)$ at any time, t, from (6.2) we can see that since all the events occur at time t, result in a vertical line segment on the Storygraph. In addition, since $\forall \alpha : \alpha 1 \leq \alpha \leq \alpha 2$ and $\forall \beta : \beta 1 \leq \beta \leq \beta 2$, this line segment consists of all the vertices. ∎

Figure 4.2 illustrates 4.1.1.

Theorem 4.1.2 *An enclosed area of arbitrary shape at time t on a Euclidean plane can be represented by a vertical line segment on a Storygraph at t.*

Proof Construct a right rectangle bounding the shape. From 4.1.1, we get a vertical line on the Storygraph. ∎

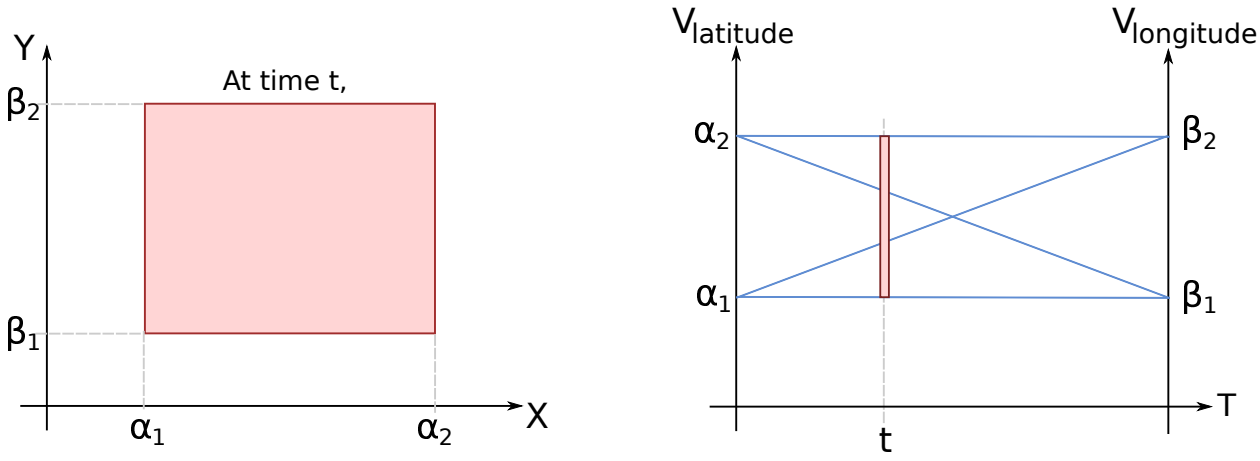


Figure (4.2) Left: An area filled with events in the Cartesian coordinate at time t, Right: Same area represented in a Storygraph.

4.2 Location lines

Location lines are the line segments connecting the two vertical axes in the storygraph. Each location line represents one real world location or a geographic coordinate and vice versa. However in some cases, location lines might result in overplotting. To avoid this, the implementation of storygraph can be made interactive so that users can hide/show location lines. In absence of locations lines, the mapping function f is not one-to-one. Thus, the Storygraph has the following properties:

Lemma 4.2.1 *Without location lines, a point on a Storygraph at time t corresponds to a line segment on a map.*

Proof We can rewrite equation (4.3) as

$$\beta = (1 - \frac{T}{x})\alpha - \frac{yT}{x} \tag{4.4}$$

Thus a fixed point (x, y) on the Storygraph corresponds to many points (α, β) on the Cartesian map at time $t = x$: those $\alpha_{min} \leq \alpha \leq \alpha_{max}$ and $\beta_{min} \leq \beta \leq \beta_{max}$ satisfying (4.4). Plotting these values of (α, β) results in a line segment with non-positive slope since $x \leq T$

as illustrated in Figure 4.3.

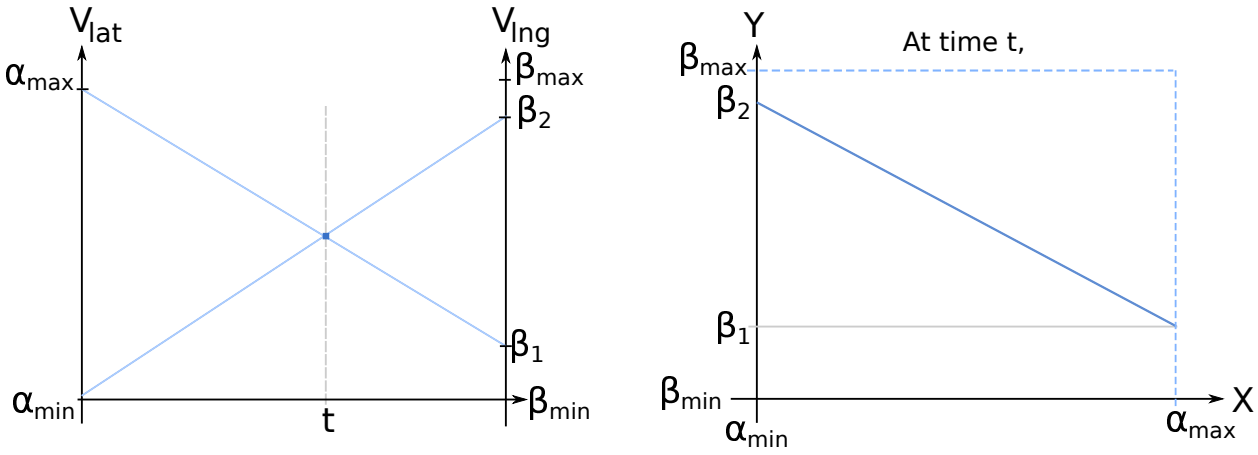


Figure (4.3) Left: A point in the Storygraph at time t and the corresponding location lines the point can belong to shaded, Right: The line segment generated in the Cartesian coordinate by mapping the point.

Lemma 4.2.2 *Without location lines, a vertical line segment at time t on a Storygraph corresponds to an area on a map.*

Proof Consider a vertical line segment, with end coordinates (x, y_1) and (x, y_2), $y_1 \leq y_2$. Using 4.2.2, these extremes of the line segment in (4.4) we get two straight line equations

$$\beta = (1 - \frac{T}{x})\alpha - \frac{y_1 T}{x} \tag{4.5}$$

$$\beta = (1 - \frac{T}{x})\alpha - \frac{y_2 T}{x} \tag{4.6}$$

Hence the vertical line segment between (x, y_1) and (x, y_2) on the Storygraph corresponds to an area between two parallel lines (4.5) to (4.6) on the Cartesian map. As in Lemma 4.2.1, this area is also bounded by the maximum and minimum values of α and β – this results in a polygon as illustrated in Figure 4.4.

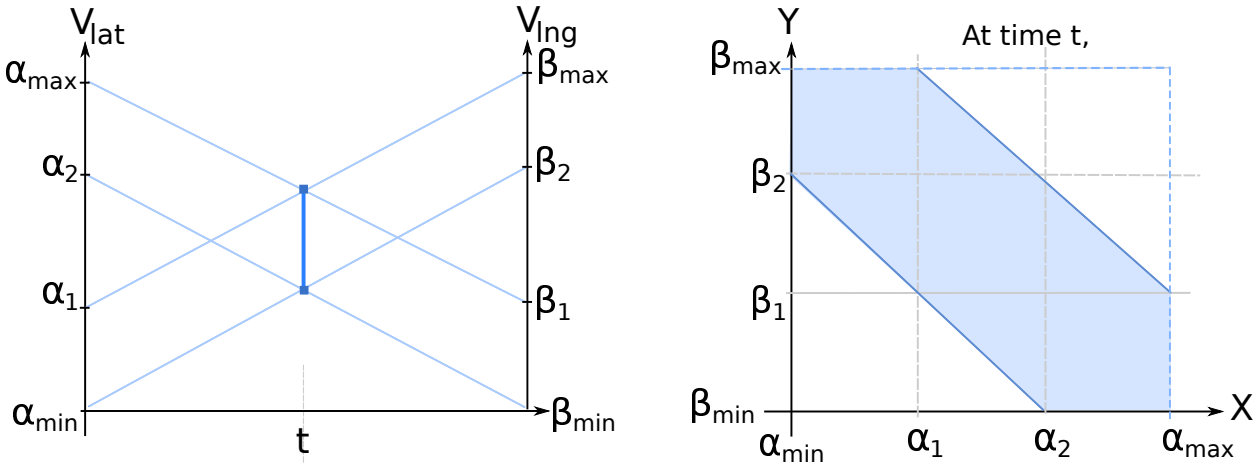


Figure (4.4) Left: A vertical line in Storygraph at time t and the corresponding location lines the line segment can belong to shaded, Right: The bounded region generated in the Cartesian coordinate by mapping the line segment.

Lemma 4.2.3 *Without location lines, a vertical line segment at t on the Storygraph corresponds to a projected area, $A_{storygraph} \geq A_{actual}$, the actual area on a Euclidean plane at t.*

Proof If the area on the plane is bounded by right rectangle, since $\forall\alpha : \alpha 1 \leq \alpha \leq \alpha 2$ and $\forall\beta : \beta 1 \leq \beta \leq \beta 2$, $A_{storygraph} = A_{actual}$. For any other shape, the vertical line segment in the Storygraph represents a rectangular bounding box (from 4.1.2). Thus, $\exists\alpha : \alpha \in A_{storygraph} - A_{actual}$. Hence, $A_{storygraph} > A_{actual}$ ∎

Note: The point of intersection of the location lines do not have any meaning or significance. However, the crossing of the lines denote that the two locations are diagonally apart. In addition especially in cases of interactive Storygraph, the two events may or may not be close. Closeness is a relative measure and its up to the users of the Storygraph to decide when to call two locations close or far apart. Figure 4.5 shows two figures to illustrate this fact. In the left figure, two locations appear to be close in the storygraph but their location lines are far apart relative to the figure on the right.

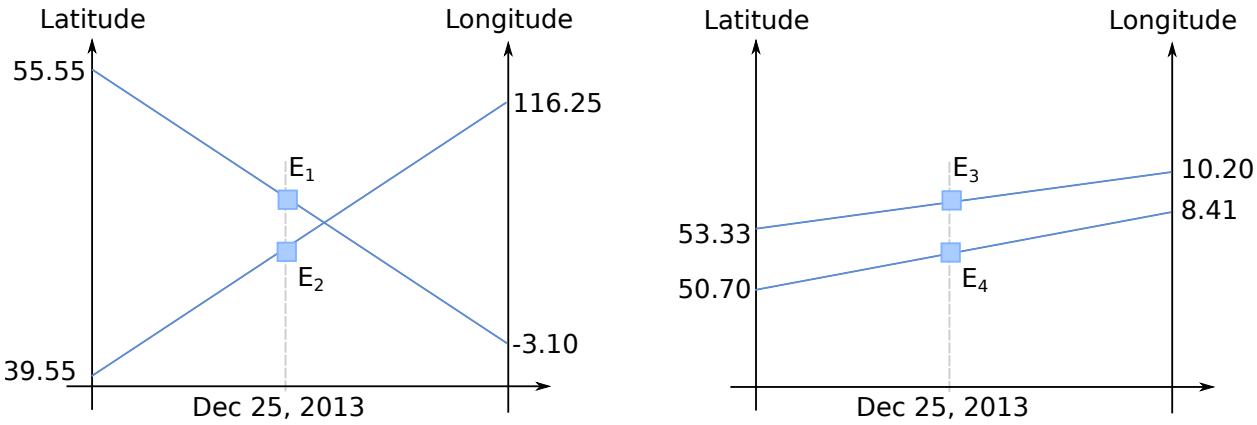


Figure (4.5) Left: Two events seemingly close in the storygraph but far apart due to their location lines, Right: Two events close to each other in the storygraph as well as geographically.

The closeness of the events depend on location lines and closeness of location lines in turn depend upon the scale of the graph. If the latitude axis in the right figure was stretched so that the max and min values in the axis was 53.33 and 50.70, then the generated location lines will have different slopes - one positive and one negative. More, it would seem like the the same two events occurred in far apart locations. Hence it is often important to note the scales in all the axes during interpretation.

4.3 Storyline

Storyline is a series of events associated with a person or a group of people. In the storygraph and rest of the text, people associated with event are called *actors*. Hence, drawing storylines require dataset having actors in addition to latitude, longitude and timestamps. Storylines are drawn on storygraph by connecting all the events associated with an actor sequentially resulting in a polyline. If there is more than one actor, a storyline is created for each of these actors as shown in Figure 4.6.

Figure 4.6 show the storylines of two fictional actors *Jake* and *Amanda*. It also shows a map on the left side with locations A, B, C and D. *Amanda* starts from point A at t_1

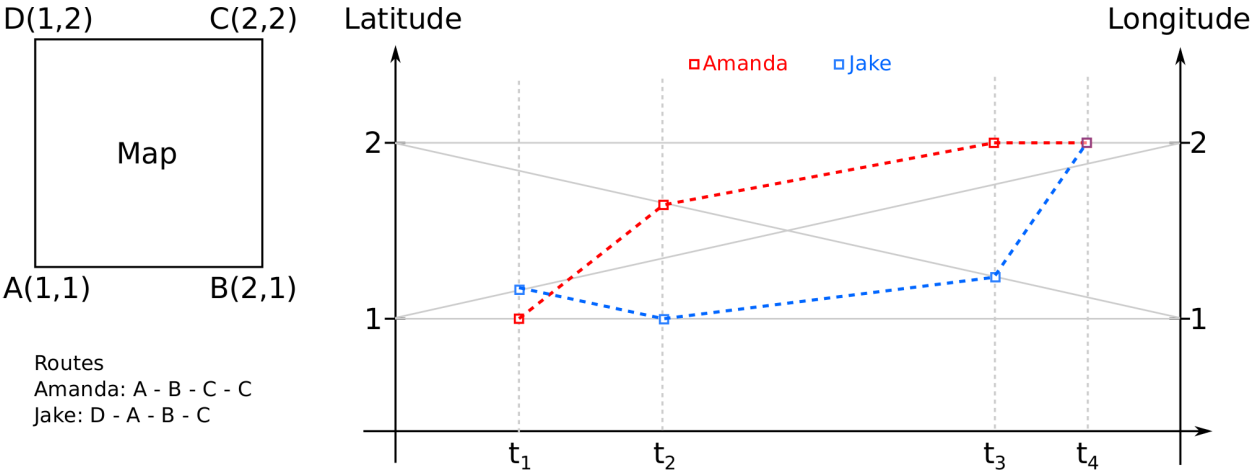


Figure (4.6) Storylines of two characters Jack and Amanda plotted on Storygraph. Amanda travelling from A to C and staying at C, Jake travelling from D to C via A and B.

and then reaches B at t_2, C at t_3 and stays at C. *Jake* starts from D and then reaches C via A and B. The storylines plotted on the storygraph shows the same information. From this figure, users can not only trace the pattern of the movement but also instances where two or more characters might have met; C in this case. Hence it allows users to visualize the relationship and interactions between the actors, which are often difficult to extract from the textual description. Storylines are especially helpful in visualizing movement as in [57][58][59] because a user can see how different actors meet at certain events and then move on to different directions.

Storylines can be compared to well established space-time paths from timegeography [60][48][28]. A direct side by side comparison of storylines to space-time paths is shown in Figure 4.7. The figure shows three co-location phenomenons namely: co-location in time, co-location in space and co-locations in time and space. Co-location in time means that two actors might have been in the different places but both of them were at the same place for a given period of time. Co-location in space means that both actors were at the same spot but during different times. Co-location in time and space means that both actors were at the same location at the same time period.

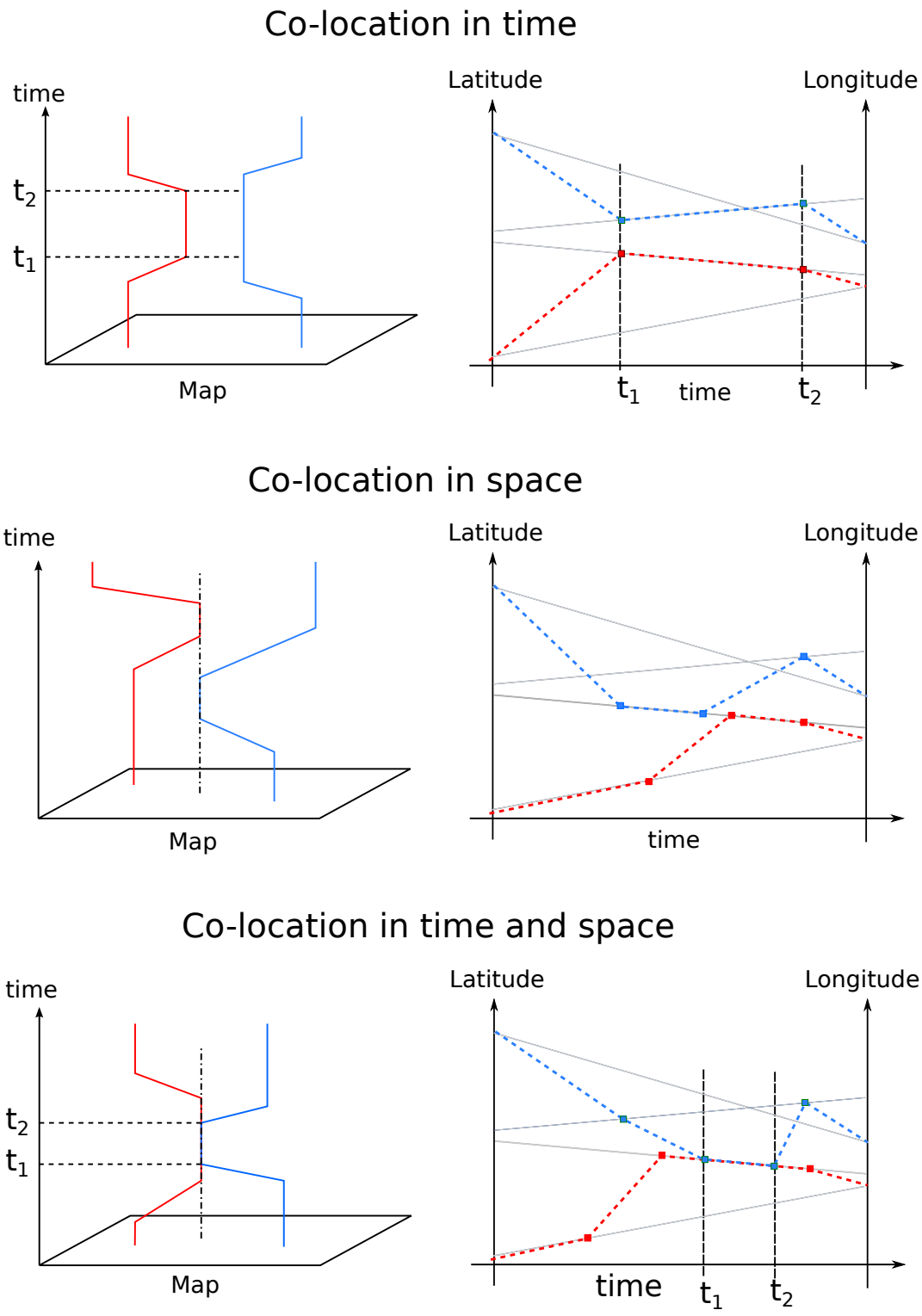


Figure (4.7) Side by side comparision of storylines to space-time paths. All the left figures show different concepts in time geography using space time paths (adapted from [61]). All the figure in right show the information using storylines in Storygraph.

PART 5

CASE STUDIES

In this chapter, I present few case studies visualized using Storygraph, published in various computer science and humanities conferences. I also discuss on previously undocumented patterns that observed in these datasets.

5.1 Afghanistan War Log (2004-2010)

The War Diary comprises U.S. military significant activity reports from Afghanistan during the period 2004-2010 [62]. Consisting of approximately 60K highly structured records, the data provides a rigorously categorized and unprecedented look at the daily conduct of war.

Afghanistan War Logs have been visualized by many organizations and leading news agencies like Guardian.co.uk and CNN as shown in Figure 5.1 and Figure 5.2. While these visualizations are express all the statistics and the evolution of the conflict, I noticed that these visualizations do have a fine grained timeline. Figure 5.1 does give a sense of time using small multiples but then again since each of the mini maps show what happened over a period of an entire year, fine grained information is lost. CNN focuses more on the magnitude of the event rather than what happened when. Another graph by plumegraph.org show in Figure 5.3 presents the information in an interactive 3D view similar to Galatsky's technique[50]. Limitations of this particular visualization that I experienced are: the interactivity is limited to zooming and rotating it along Z-axis (as opposed to free rotation). In addition, the zooming does not change the base map, it just makes the figure bigger. Lastly, events are groups according to some range of date so that they from planes. This does not add any extra information compared to the Guardian visualization; only makes the data harder to read in 3D. These motivated me to apply Storygraph on the Afghanistan war logs.

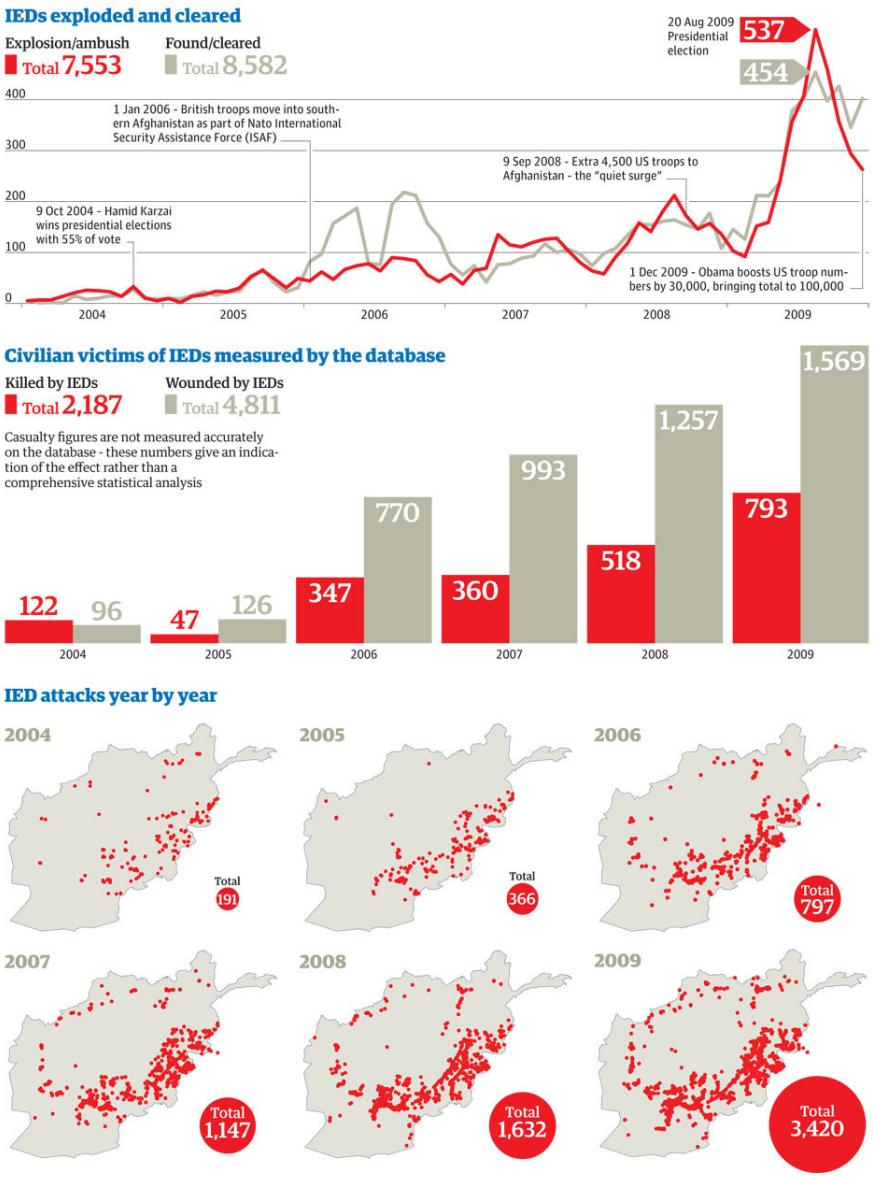


Figure (5.1) Visualization of war in Afghanistan by Guardian.co.uk

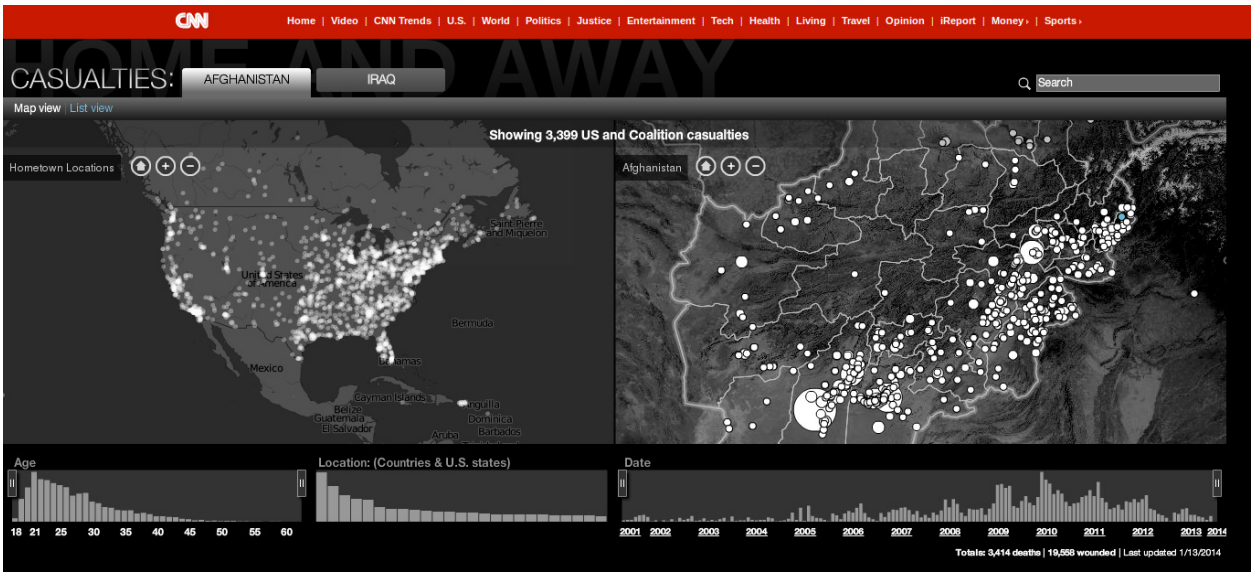


Figure (5.2) Interactive visualization of war in Afghanistan and Iraq by CNN

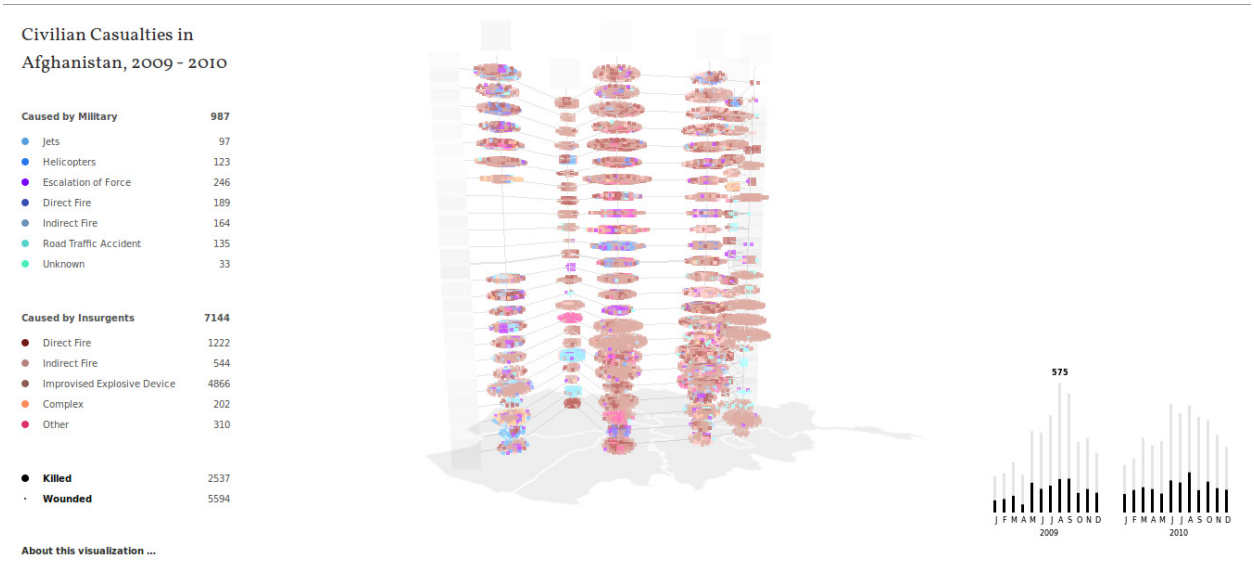


Figure (5.3) Interactive visualization of war in Afghanistan by plumegraph.org

Afghanistan data consisted of approximately 92,000 records and more than 30 dimensions. A detailed description of more all these fields have been provided by Guardian.co.uk[63]. For Storygraph, I wanted to answer 'what happened where' and 'what types of incidents dominated the war period' using visual analytics. Hence, I only used

Latitude,*Longitude*, *Time* and *Type of action*. *Type of action* was represented by the color of the glyph. Any rows with null values for any of these fields were ruled out during the implementation. This included "Air Strikes" under *Type* as it had no latitude and longitude values associated with it. This resulted in the image shown in Figure 5.4.

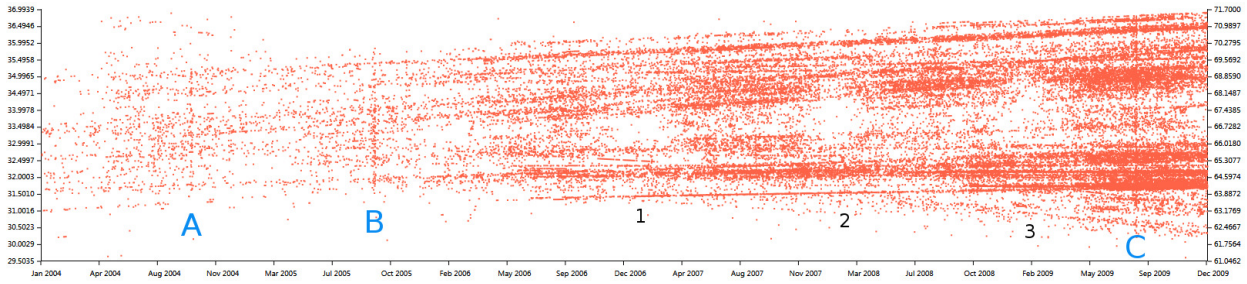


Figure (5.4) Storygraph showing all the events during Afghanistan war from 2004 to 2009 across different regions of the country. Patterns A, B and C show many events taking place within a short time frame. Patterns 1-3 show periodic 'holes'

Figure 5.4 consists all types of events (total number of events totaling to 15). These included "Friendly Action", "Suspicious Incident", "Enemy Action", "Explosive Hazard" etc. Next I wanted to explore how much of these plotted events contribute to the last two: "Enemy Action" and "Explosive Hazard". So, when these were plotted, resulted in Figure 5.5 - both were similar leading to a conclusion a significant amount of these events were due to the type "Enemy Action" and "Explosive Hazard".

From Figure 5.5, two sets of distinct patterns marked by A-C and 1-3 are observed. The vertical bands marked by A-C are formed due to the events clustering between Aug-Nov 2004, Jul-Oct 2005 and near Sept 2009. They indicate widespread and coordinated "Enemy Action" and "Explosive hazards" events at a very short period of time. By correlating these dates to broader events in Afghanistan, we discovered that these clusters were proximal to elections; our hypothesis is that the incidence of violence, and therefore the number of activity reports, increases during elections.

With the discovery of elections, we were also interested to see the whether the frequency

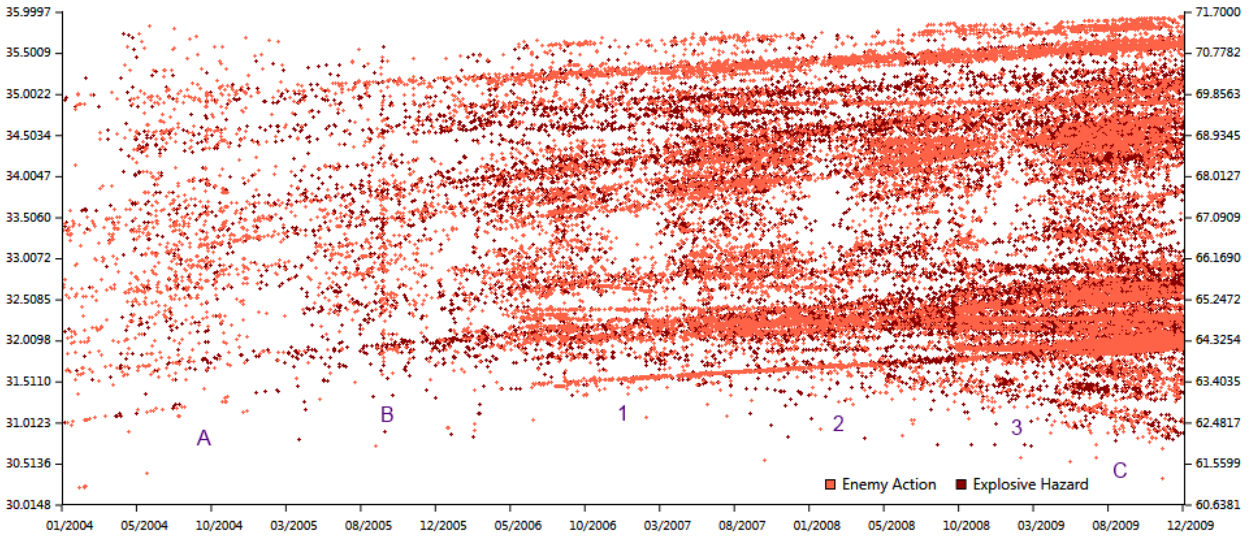


Figure (5.5) Storygraph showing "Enemy Action" and "Explosive Hazards" during Afghanistan war from 2004 to 2009 across different regions of the country. Again, patterns from Figure 5.4 are also present in this figure.

of diplomatic cables incoming and outgoing cables from US Embassy at Kabul change. I downloaded this data from cablegatesearch.net[64]. Since all the cables originated from Kabul, I just plotted a line if there was a cable on that date and change the thickness of the line directly corresponding to the frequency. When I overlayed this layer over the Storygraph and zoomed at the latest election, I obtained Figure 5.6. An interesting find was that with so much events taking place during the election, the frequency of cables were not that high as compared to around July 2009 mark.

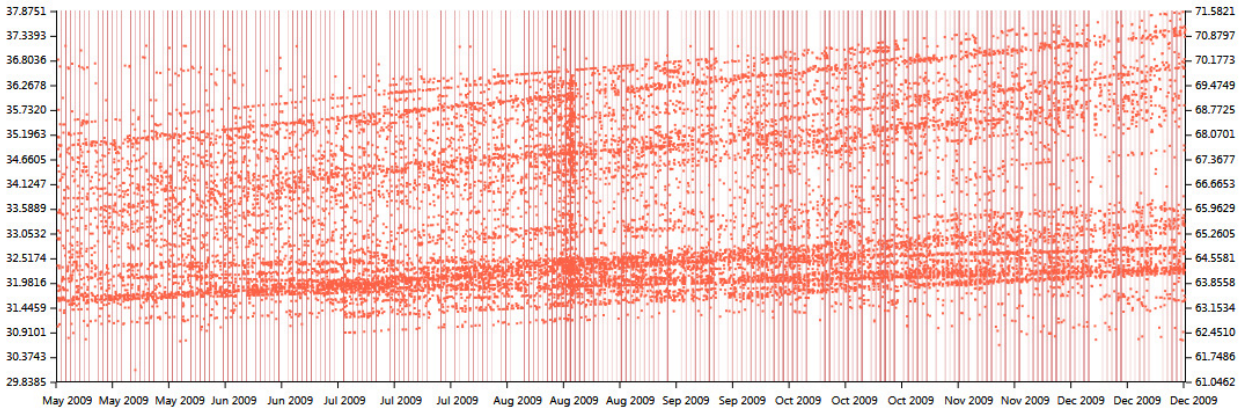


Figure (5.6) Storygraph zoomed with the latest election in 2009 centered. The frequency of cables have been overlayed.

Numbers 1-3 in the figure shows a periodicity of voids, or a lack of reports in those areas. These "holes" seem to appear around the end of the year. From location lines, we found that the geographic location by these holes correspond to a section of the Kabul-Kandahar highway, a key portion of Afghanistans national road system and a main target of attacks. The Storygraph visualization shows that there have been regular quiet periods for that section of the highway around the end of 2005, 2006, 2007, and 2008. Since the data set does not extend beyond December 2009, we are unable to confirm if the pattern repeated in 2009. An interview with an Afghanistan War veteran suggested that these holes were probably because during the winter the sub-zero temperatures resulted in formation of solid layer of ice which was impossible to dig and plant IED's - thus resulting in no action. To our knowledge, this pattern has not been identified or discussed in any published report.

Figure 5.7 shows the death toll as the war progressed. We observed that the frequency of death rose with time. A more interesting observation however, as in case of Figure 5.5, is the periodicity in the number of deaths. With this we can directly correlate the number of deaths with the number of explosive hazards and enemy actions. Although this is not new information, it demonstrates Storygraphs ability to help identify significant event patterns.

Next, I drew storylines of random seven Afghanistan based combat units to see if any patterns can be seen resulting in Figure 5.8. It can be seen that TF Protector and TF

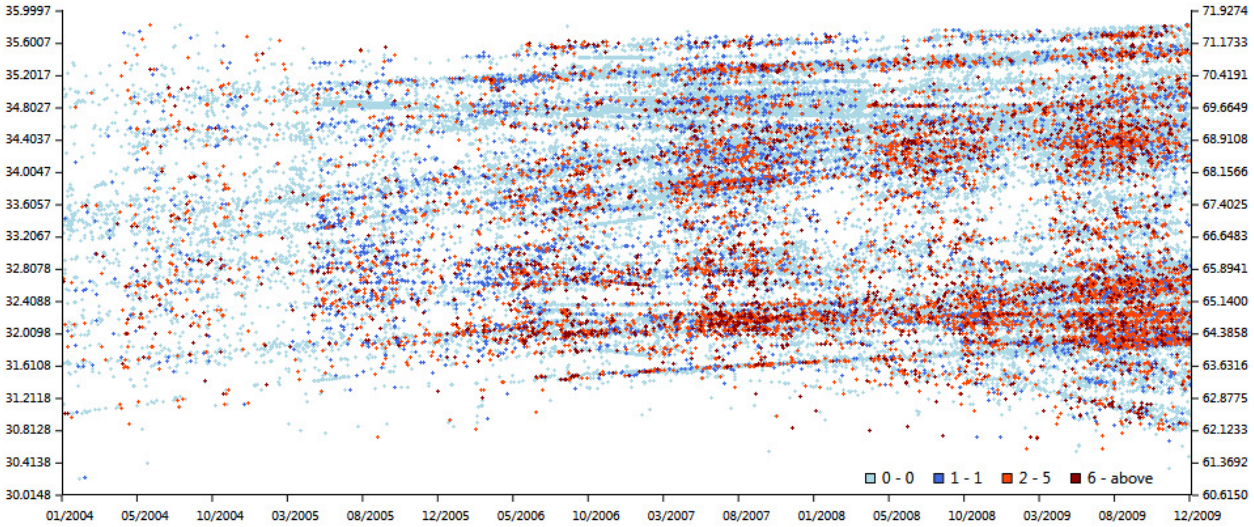


Figure (5.7) Storygraph of Afghanistan war showing periodicity in the total number of deaths. The lighter shades denote events with less than one death reported while darker shades denote events with more than one death.

Cyclone were working close by and they meet several times. Similarly TF White Eagle is continuously moving from one location to another while TF Protector is stationed constantly at the same place. This storylines not only show which two units met but also shows the mobility of units in the scene.

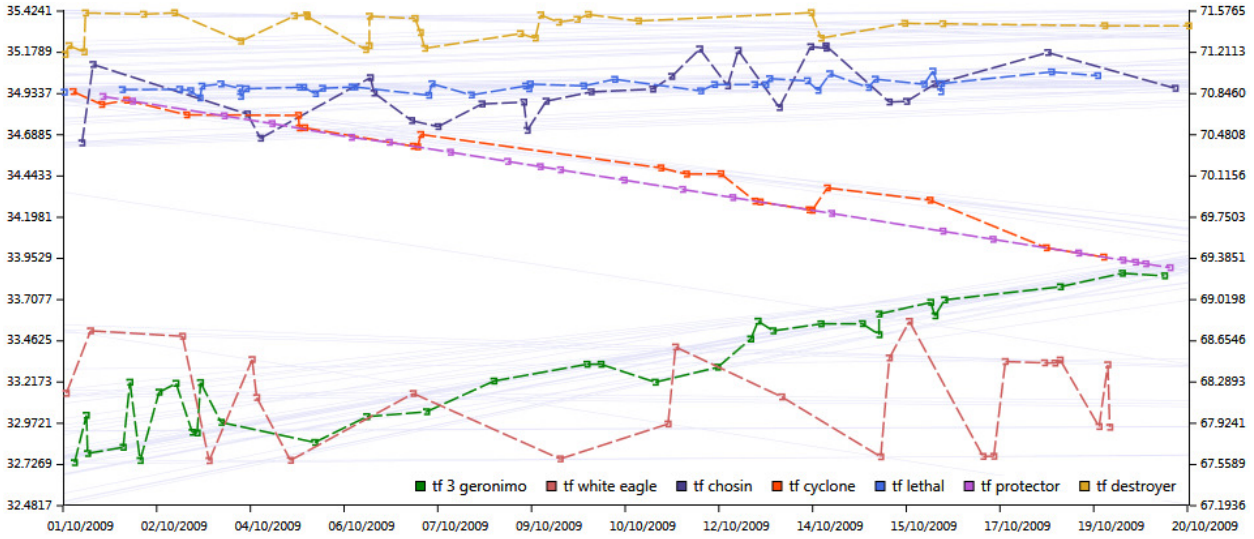


Figure (5.8) Storylines of seven Afghanistan based combat units from October 1 - 20, 2009.

5.2 Unexploded Ordnance Laos (1964-1973)

An ongoing problem in war torn regions is the persistence of unexploded ordnance. Established by the Lao government with support by various NGOs, the Lao National Unexploded Ordnance Programme (UXO Lao) addresses this ongoing problem so as to reduce the number of new casualties and increase the amount of land available for agriculture. This data set details bombings in Laos by the USAF during the period of the war, 1968-1975 and is also a military, descriptive log of the bombings. It consists of 60,000 reports documenting approximately 2 million tons of ordnance [65]. Figure 5.9 shows the graph obtained from this dataset.

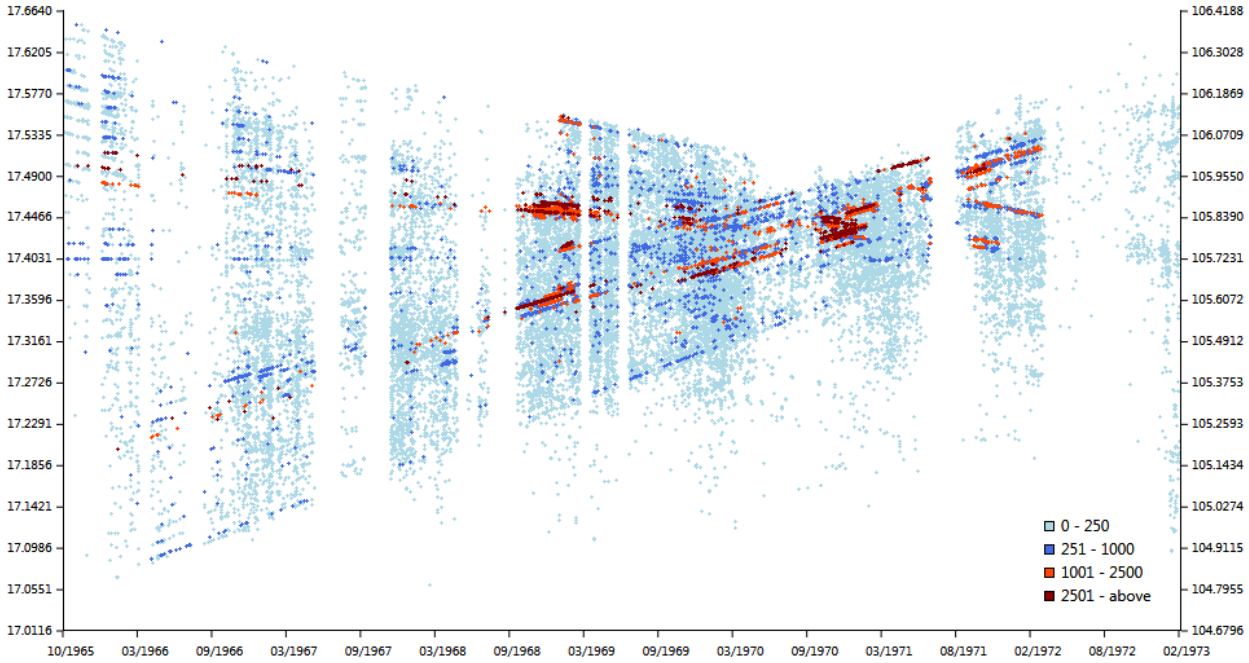


Figure (5.9) Storygraph generated from the Laos dataset. Each event corresponds to a bombing. The color signifies the number of bombs dropped as shown in the legend.

The Storygraph obtained from this dataset shows three distinct patterns.

1. The bombings intensified during October 1969 - Mar 1970 (A).

2. From the start of the data to June, 1966 show cluster of events that form distinct

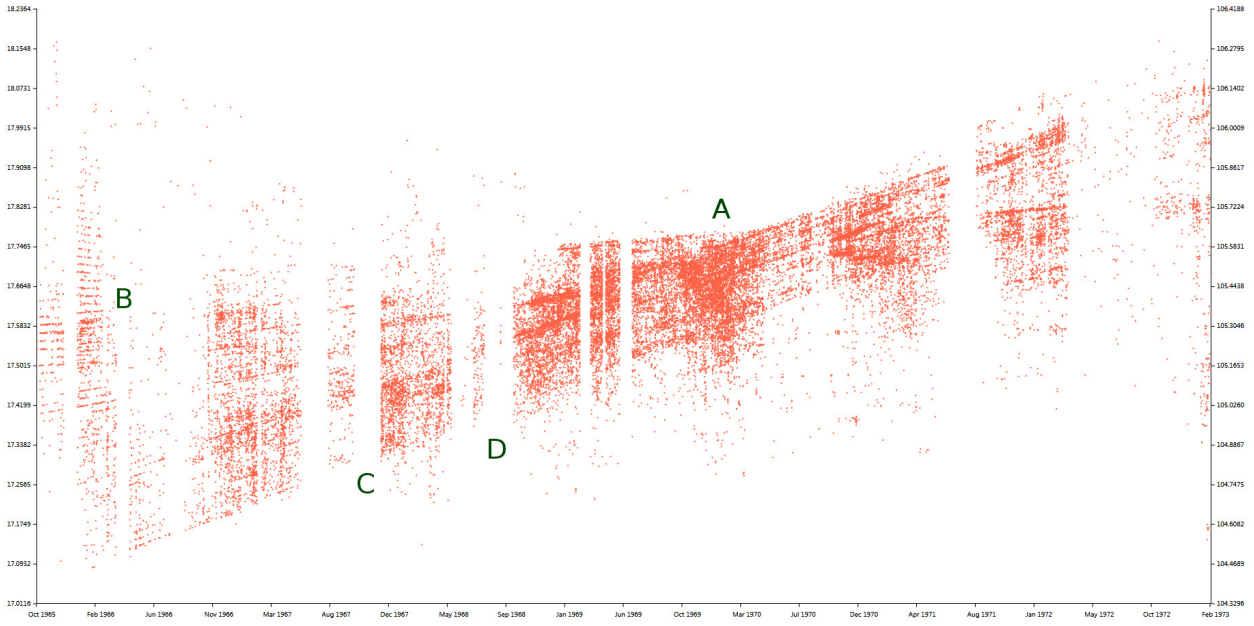


Figure (5.10) All the bombings in Laos (superset of Figure 5.9) annotated by A-D and size of the marker reduced to 1×1 pixel. 'A' shows possible intensification of bombing. 'B' shows few locations were regularly bombed. 'C' and 'D' show voids which could have resulted from data being redacted or pausing of bombing.

lines. These lines signify that the bombings were focused at certain locations at almost regular intervals (B).

3. The bombings reduced drastically after March 1972 when most US troops left Vietnam.

4. The vertical bands above the 03/1970, 03/1971 and 02/1972 show the periodicity in the bombings (C and D).

5. The patterns of bombing data are interspersed with periodic bands of white.

We have two hypothesis for those voids: either they could mean that bombings were paused during that range of time, it could also mean that the data correlated to those raids during that period was redacted from the set. Like much military data, classified operations such as those by special forces are frequently not contained within general operation activity reports. It is beyond the scope of this paper to test these hypotheses. However, it demonstrates the ability of Storygraph to discover patterns and form hypotheses.

Identifying the focal locations of bombing campaigns helps groups like UXO Lao address the areas most in need of remediation. By their estimates, approximately 80 million unexploded bombs remain in Laos as a result of U.S. Air Force (USAF) bombing missions during the Vietnam Conflict. These bombs affect 25% of all villages, all 17 provinces, and have resulted in approximately 25,000 injuries and fatalities in the post-war period, 1974-2008.

5.3 World Trade Center (9/11)

Following the attacks of September 11, 2001, interviews were conducted with first responders to the World Trade Towers. Each of those 503 interviews describes one witness account of the event – skyscrapers collapsing, choking dust, chaotic communications, fatal desperation. Although focused on individual narratives, each interview corresponds to a larger sequence of events: two massive violations of the right to life that took place in New York City. In this particular work [66], we were more interested in visualizing how the events progressed throughout the day. The 'events' here were manually extracted from the documents and the time for all the other events between documented events like the towers falling, were interpolated.

The vertical patterns marked by t1-t4 in the Storygraph in Figure 5.11 shows the key events - planes crashing into towers and the towers collapsing. The horizontal funnel layout of the points, with the mouth to the left axis, indicates that documentation shows people converging from a wider geographic area to the narrow area around Ground Zero. In essence, Figure 5.11 shows first responders converging on the scene of two terrorist attacks.

Figure 5.12 uses the same data to show Storylines of four emergency personnel as they move throughout the spatio-temporal corpus domain. Four features in Figure 4.6 are of particular importance. First, the geographic domain is highly constrained and covers an area from Staten Island to Central Park. Second, with just $10 - 20$ extracted fabula, a sense of the path of these individuals through the event emerges. The chaotic jumble of points in the period from $8:39$ AM to $9:30$ AM corresponds to the event's most chaotic moments. Third, the lines of Firefighters Loutsky and Smith stabilize at two locations as

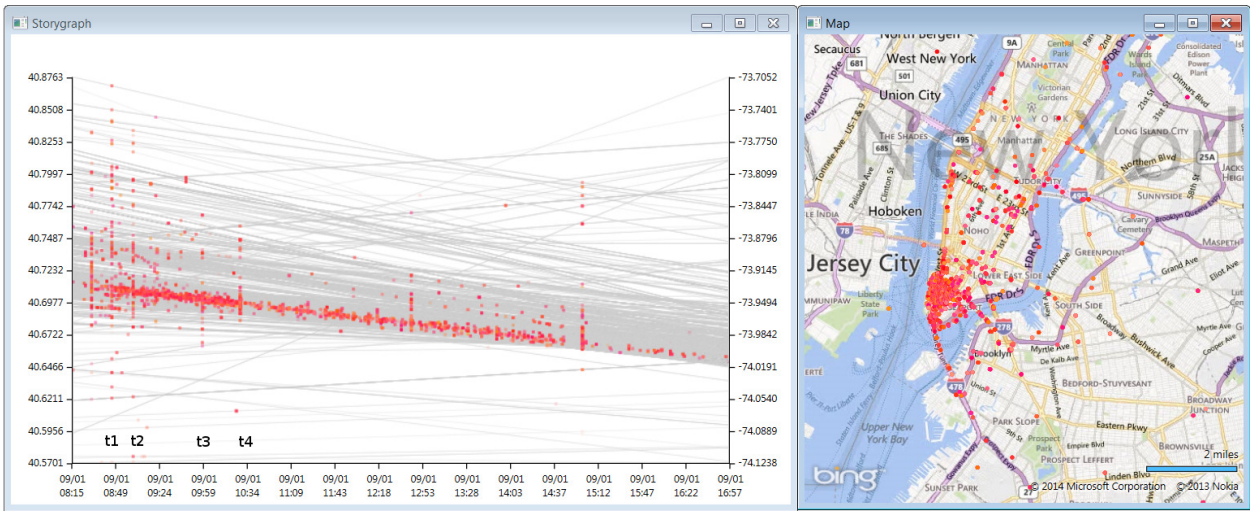


Figure (5.11) Left: Storygraph showing approximately 7000 events within 12 hours during 9/11 attack on WTC. Annotations $t1 - t4$ mark the key events: $t1(8:46)$, first plane crashes into the North Tower; $t2(9:03)$, second plane crashes into South Tower; $t3(9:59)$, South Tower collapses; $t4(10:28)$, North Tower collapses. At each of these times, events occurred simultaneously at multiple locations (marked by vertically aligned events). In addition, it can also be observed that the events clustered around the location $(40.70, -74.00)$. Right: Same set of events plotted on the map. Maps supplement Storygraphs as identifying locations on maps is relatively more intuitive.

they move from emergent crisis to emergency care. And finally, there is the blue storyline of Chief Ganci, which ends at $10:12$ at $40.71, -74.01$, approximately 16 minutes prior to the collapse of WTC Tower 1. Chief Ganci, the highest ranking uniformed fire officer in FDNY, died in that collapse. What Storygraph enables is the identification and organization of fragments from other's statements to reveal the story of what happened to him that day.

5.4 Trans-Atlantic Slave Trade Voyages (1514-1865)

The Voyages dataset obtained from Emory's Slave Trade Voyage database [67] contains the specifics of slave trade voyages in 276 fields of data variables and inputted variables for each voyage, including the ship name, captain name, where the slaves were purchased, and where they were sold. Begun in the late 1960s, and radically enriched by new data from Spanish and Portuguese sources in 2001-2005, this multi source database describes in detail the specifics of what is widely regarded as amongst the most horrendous of mass violations

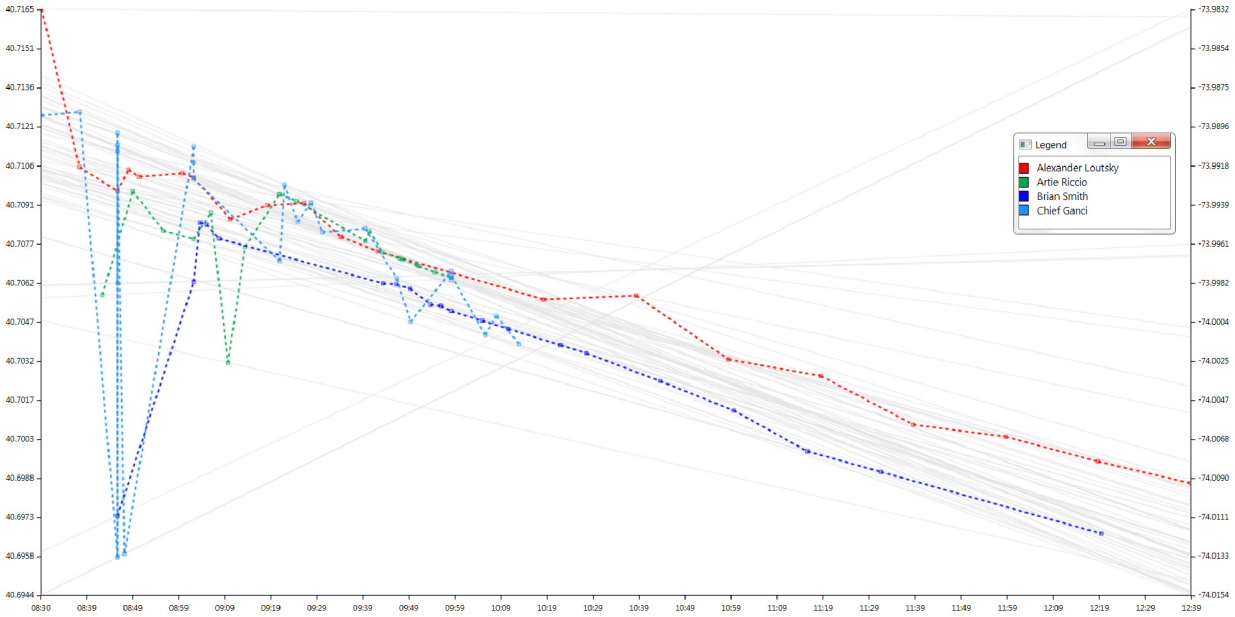


Figure (5.12) Storylines of three firefighters and one EMT.

of human rights in history. The 2010 version of this data describes 34,948 voyages.

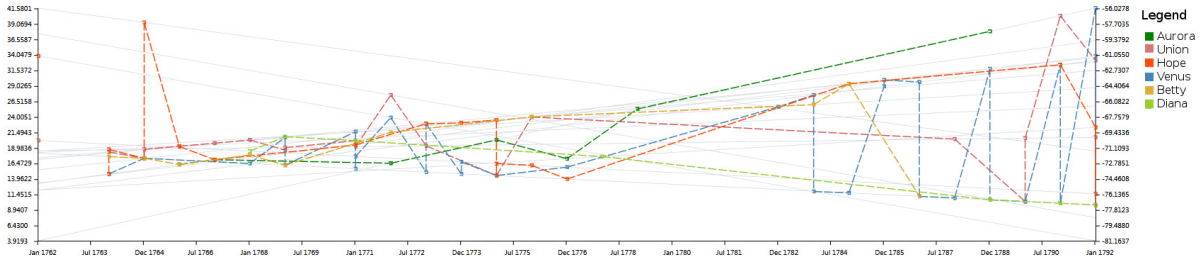


Figure (5.13) Storylines showing the voyages of five different slave trader ships during 1700 A.D. The near vertical lines have been formed as only year was mentioned in the voyages. Hence if a ship went from one place to another within a year, they are lined up vertically.

The generated Storylines for the locations at which six ships conducted their purchases of slaves can be seen in Figure 5.13. Vertical transits in the storyline are caused by missing month and day values for the date; absent more specific temporal data, a ship making two transits in one year would show as being at two locations. From this figure, it can be seen that few locations were significant, and even fewer represented convergence points at which

multiple ships bought simultaneously. These significant locations emerge as the slopes upon which data points fall, and where many Storylines converge. One such point can be seen near the left side of the graph, near Dec 1764. What is suggested by that convergence is an event at which so many individuals were sold into bondage, that it required otherwise usually independent ships to converge. One research question this type of graph could facilitate the answer to is the role of individual ships in the diasporic fragmentation of communities; if a specific ship frequently originated from a port of purchase, and then disembarked slaves at sale points throughout the new world, it would indicate the specific role of that ship and its captain in the social and linguistic fragmenting of a community.

Figure 5.15 shows the events where the slaves were bought across all records from the dataset. Figure 5.15 shows the same information plotted in a our linked map view. Navigating between the map and graph view would allow for a researcher to highlight the routes, ships, captains, and destinations of all ships embarking from a particular point of sale, further enabling the study of the role of these voyages in the fragmentation of communities.

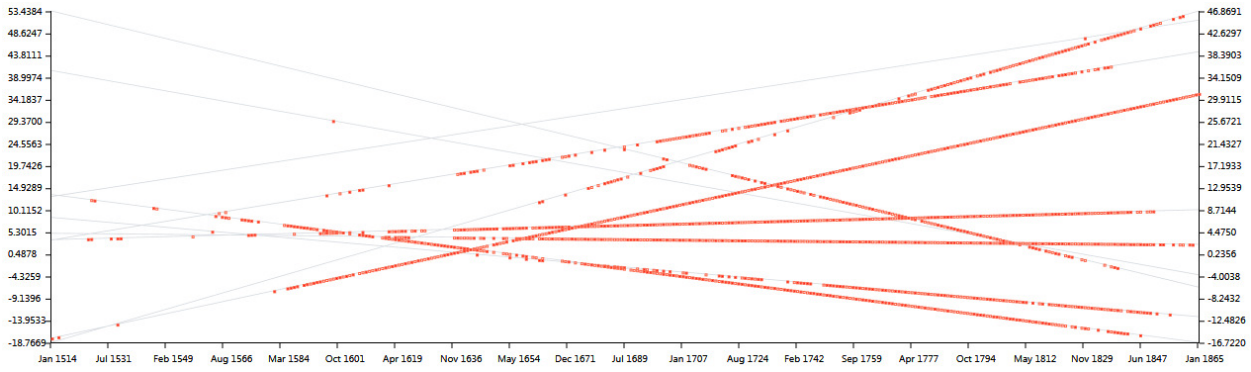


Figure (5.14) Storygraph generated from slave voyage dataset where each event is denotes where the ships got the slaves.

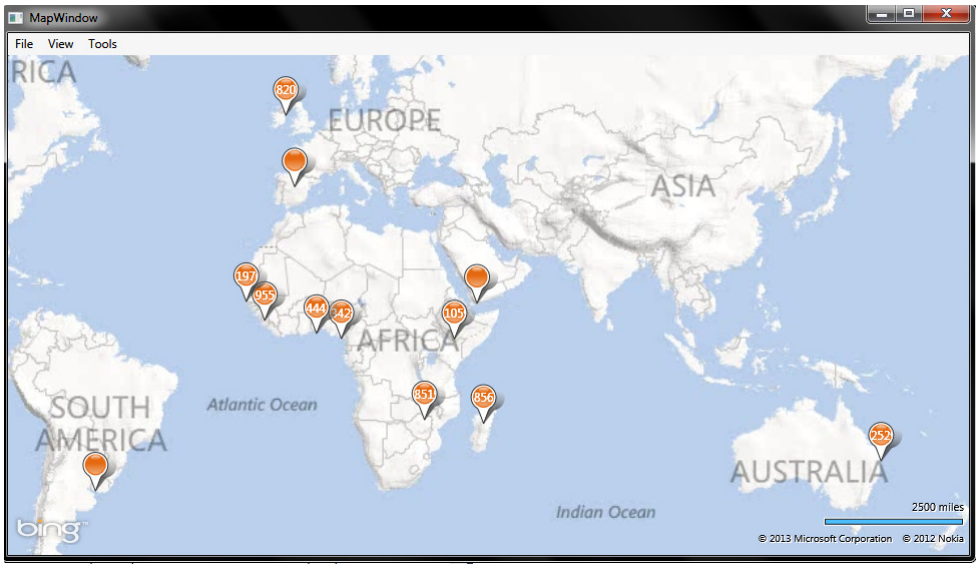


Figure (5.15) Supplement map view for Figure 5.15 showing where the voyages got their slaves from.

PART 6

BEYOND SPATIO-TEMPORAL DATA: VISUALIZING TIME AND GEOGRAPHY OF OPEN SOURCE SOFTWARE WITH STORYGRAPH

6.1 Introduction

Free/Libre and Open Source Software (FLOSS) are in most cases created and maintained by developers voluntarily. Since the collaboration between developers takes place online without any governing institutions, detailed designs, project plan or list of deliverables, developers globally can contribute to the software[68]. This is further facilitated by online open source repositories like GitHub, SourceForge, Google code, CodePlex, etc. Even though open source software projects may seem to involve developers from all around the world, literature show a strong geographic bias of these developers towards North America and Europe [69][70][71][72].

FLOSS provide good platforms for studying distributed software development since most of them are global efforts. Prior work has studied how the difference in geographical locations and time affect the nature of collaboration and the quality of the product. However, most of these analyses did not use data visualization, even though visualization is a valuable technique for analyzing geographical information. Heller, et al. [73] and Xu, et al. [74] developed methods to visualize geographic locations and the collaboration or relationships among the developers. However, these methods do not include time information. Our work improves on the previous work by integrating geographic and temporal information in one visualization.

In this section I demonstrate the application of Storygraph to visualize the geographic location of the developers in FLOSS projects together with their time of commits.

6.2 Concept

The Storygraph is composed of two parallel vertical axes V_{lat} and V_{lng} and an orthogonal horizontal axis H. As in Cartesian coordinates, the values in the axes are ordered in ascending order from top to bottom in the vertical axes and from right to left in the horizontal axis. The vertical axes represent latitude and longitude of a location while the horizontal axis represents time. In order to plot a point on a Storygraph, a precise location and a time stamp is required. Given a commit time stamp together with the location of the developer, a line segment connecting the latitude and longitude on the vertical axes corresponding to the developer's location is drawn in the Storygraph. A marker is then placed on this line at the time of the commit. Multiple commits from the same developer are represented as multiple markers along that location line. We assume that the location of the developers are fixed in this paper.

An example of Storygraph is shown in Figure 6.1. The top figure shows code commits from developers in three locations $(x1, y1), (x2, y2)$ and $(x3, y3)$ at different times $t1 - t5$. The bottom figure shows the same coordinates plotted on the Storygraph. It can be observed that the temporal aspect is more evident in the Storygraph as compared to the traditional map.

6.3 Mathematical model

Given a code commit at time t by a developer in a location with geo-coordinates (α, β), first a line segment connecting latitude and longitude on the two vertical axes is drawn. This line is called a *location line* and represents real geographic location. Second, a marker is placed along this location line at time t.

The function $f(\alpha, \beta, t) \rightarrow (x, y)$ which maps an event to the 2D Storygraph plane can be formally written as follows:

$$y = \frac{(\beta - \alpha)(x - T_{min})}{T_{max} - T_{min}} + \alpha \tag{6.1}$$

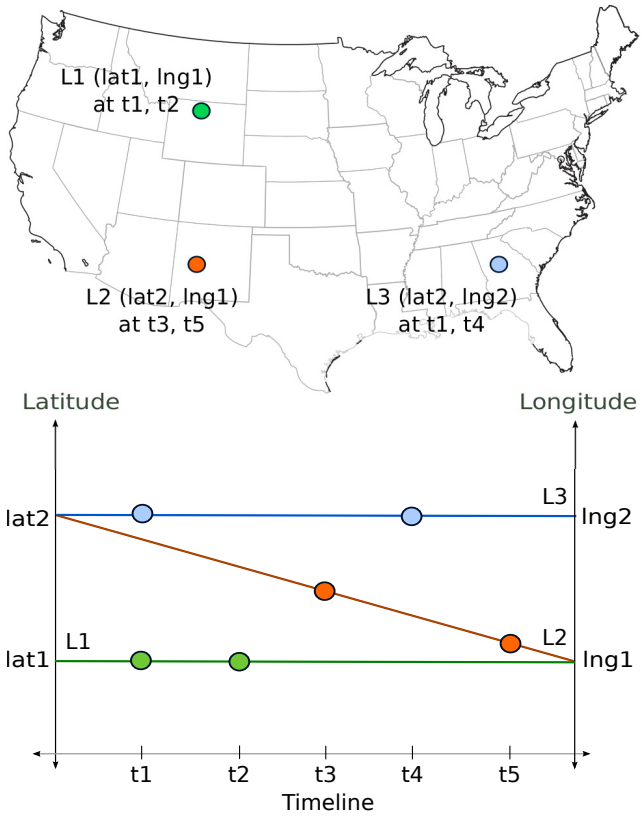


Figure (6.1) Top: A geographic map showing commits from three locations $L1 - L3$ at different times $t1 - t5$. $L1$ and $L2$ share the same longitude, $lng2$ and $L2$ and $L3$ share the same latitude, $lat2$. Bottom: Same set of events plotted on the Storygraph.

$$x = t \tag{6.2}$$

where T_{min} and T_{max} are the maximum and minimum time stamps within the data set.

6.3.1 Interpretation

One of the important properties of Storygraph is its ability to show code commits from multiple locations and times on a single chart. This helps to show where the developers are most active and also when the developers are most active during the software life cycle.

Different locations, same time Multiple code commits at a unit time are represented as a vertical cluster of points at that time. An example of many code commits at one time can be observed in Figure 6.1 (bottom) at $t2$ and the areas pointed by B and C in

Figure 6.2.

Same location, different time Multiple code commits from a certain location are represented as a cluster of points along the location line eg. Figure 6.1(bottom) - all three location lines have two events each. This can also be observed in Figure 6.3 in location lines marked by US and UK.

6.3.2 Location lines

Location lines are a crucial part of Storygraph. Without location lines, a point marker on the graph could belong to multiple locations. By geometry, the location lines of two proximal locations on a map are also close on a Storygraph. Despite this, in situations where data is dense enough to reveal meaningful patterns, drawing location lines may lead to over plotting. In such cases, the location lines may be turned off. Storygraph without location lines only shows the number of commits per unit time.

To prevent cluttering resulting from the location lines, we paint the lines with varying intensity of the same color as shown in Figure 6.2 and Figure 6.3. The intensities in this case directly correspond to the frequency of commits from a location - higher the number of commits, darker the location line. Thus given the maximum and minimum number of commits for a repository, n_{max} and n_{min}, and a tuning parameter p, the color c of the location line is given by:

$$c = (\frac{n - n_{min}}{n_{max} - n_{min}} * 255^{(1/p)})^p \tag{6.3}$$

where n is the number of commits for the current line and p is the tuning parameter.

Increasing the values of p highlights the outliers (locations which have large number of commits). However, Equation 6.3 requires the location data to be noiseless when using large values for p.

6.4 Implementation

Our implementation of the Storygraph consisted of a three stage pipeline.

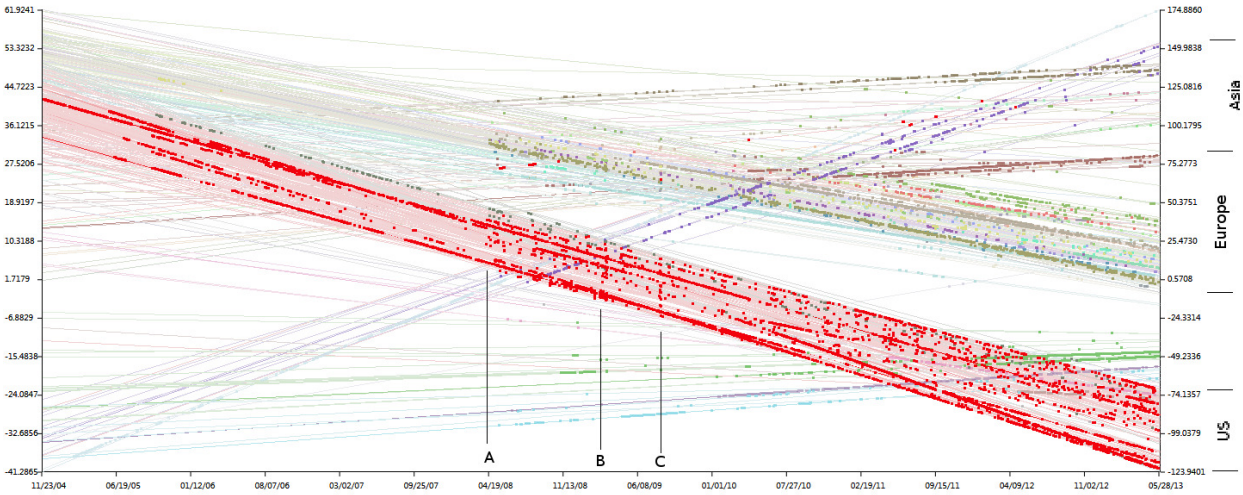


Figure (6.2) Storygraph of *Rails* development based on the GitHub VCS. Up until time marked by A, there were only few developers, majority of them in United States. After A, there were many commits from all over. B and C show many code commits within a short span of time from many locations. $US, Europe$ and $Asia$ show the bands formed due to the clustering of location lines. Location lines have been painted with $p = 0.8$ to subdue locations with smaller number of commits.

Stage I: Getting the log and the location of the developers. For this purpose, we chose GitHub.com since it also had developer profiles with location. We used git to get the list of the commit logs from the server. Obtaining the names of the contributors from the log, we wrote a script to query GitHub for profiles of each user. We used an XML parser to extract the location of the users from the resulting pages. One of the challenges that we faced in this stage was when developers did not have an address. In these cases, we queried the LinkedIn.com database and personal websites if listed.

Stage II: Removing noise and using Google Map API for geocoding. This stage was automated using script which called the geocoding function in Google Map API. The challenges we faced in this stage included multiple locations and vague addresses. Many developers had more than one location listed in their profiles e.g. "London, Tokyo, San Francisco", "Atlanta, New York" etc. Having more than one location resulted in more than one pair of latitude and longitude. For simplicity we only considered the first location and discarded the remaining locations to address this issue. Developers who had vague addresses e.g. "Somewhere in

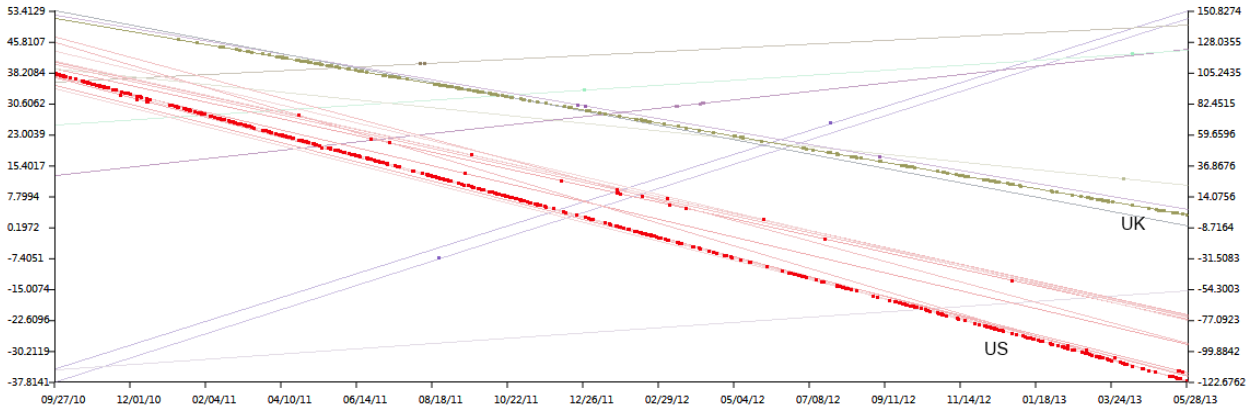


Figure (6.3) Storygraph of $D3.js$ development based on the GitHub VCS. It can be observed that developers from locations marked by US and UK have the highest number of contributions. The location lines have been painted with $p = 0.18$ to include locations with smaller number of commits.

Earth", "Dark side of the moon" etc, were either looked up manually or discarded based on their number of commits.

Stage III: Setting up database and visualizing the Storygraph. We implemented the Storygraph in Microsoft .NET using WPF Framework. The data was stored in MySQL database.

6.5 Application

We plotted the commit history of three projects from *GitHub*: *Rails*, *D3.js* and *Homebrew* on the Storygraph.

Figure 6.2 shows the Storygraph of *Rails*. Also known as Ruby on Rails, Rails is a framework running on Ruby programming language. Rails facilitates the communication between pages/applications and servers. The first version of Rails was released in December 13, 2005 and is still active. From the figure, we can see that until the time marked by A, most of the code commits were centralized in the US with few being in Europe and Asia. After time A, there were many code commits from developers all over the world. Markers B and C show multiple code commits within a short span of time from many different locations. This is generally the case before or after a major version release.

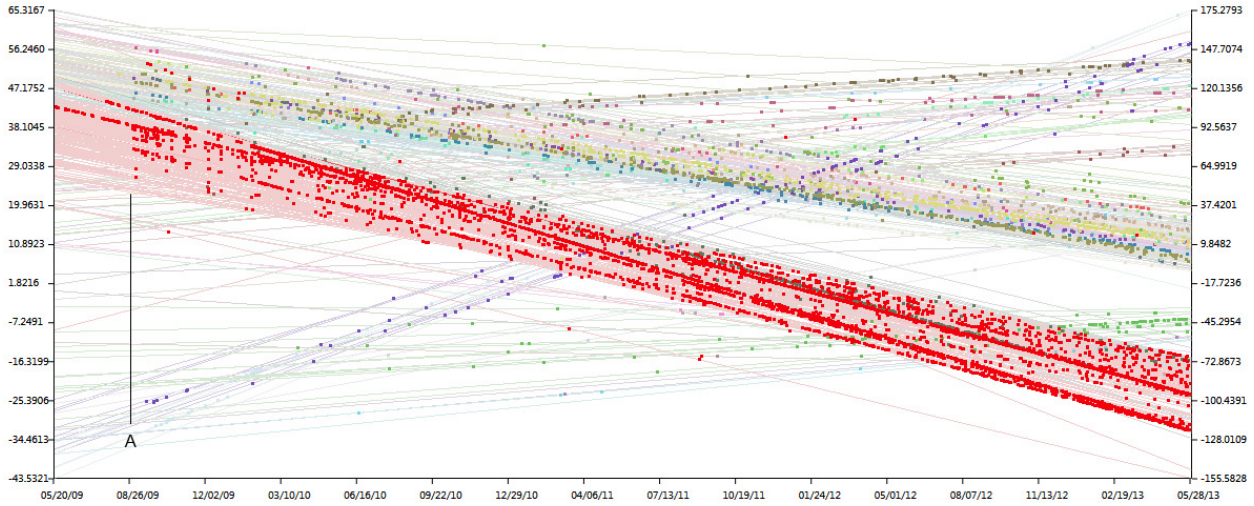


Figure (6.4) Storygraph of *Homebrew* development based on the GitHub VCS. The number of developers starts to increase after time marked by A.

The Storygraph of D3.js is shown in Figure 6.3. D3, standing for Data-Driven Documents is a W3C compliant Javascript visualization library. The first version of D3 was released in 2011 and remains active. Unlike *Rails* or *Homebrew*, two dominating contributors can been seen in the figure marked by US and UK. The set of commits marked by US corresponds to Michael (Mike) Bostock, the founder of the project from United States and UK corresponds to Jason Davies from United Kingdom.

Figure 6.4 shows the Storygraph of the OS X package manager *Homebrew*. Even though Homebrew has yet to release version 1.0 yet, it has been able to attract a lot of developers in US, Europe and Asia. Similar to *Rails*, it can been seen from the figure that after a certain period of time marked by A, a lot of people started contributing to the code from different locations.

Few inferences can be made from Figures 1-3:

- There is a time period in the start of open source projects where there is only a sole developer working on the project. After that point, the contributors tend to increase rapidly.

- In most cases, the developers contributing at the start of the project continue to

contribute to the code actively till the end. This trait is more clearly observed in Figure 6.3.

6.6 Related Work

Some previous works [75][76] have tried to create storylines for open source projects. But they focus on timelines without considering geographical locations of the developers. However, some works [71, 69, 77] have shown that geographical location is important for analyzing the nature of collaborations and the effectiveness of distributed software development. These works, however, have not used visualization for their geographical location analysis. Heller, et al. [73] proposed three techniques (map with links, small multiples, and matrix diagrams) to visualize the geographical locations of the developers as well as their collaborations. However, their methods do not have integrated time information. Xu, et al. [74] proposed a graph chart to visualize the relationships among developers from different regions. That method did not integrate temporal information.

6.7 Discussion and Conclusion

We presented Storygraph and applied our technique to visualize the time of the commits and the location of developers on a single chart. Our visualization shows a narrative of how the developers all over the world collaborated to create FLOSS based on the data from GitHub. The main advantage of Storygraph over maps or timelines is that it can show the temporal aspect of the data without losing the big picture. However, it does not consider the user contribution when highlighting the location lines and also does not show where and which files were modified. We intend to incorporate these in our future design. We also intend to create a web version and collect user feedback to evaluate the effectiveness of our visualization.

PART 7

BEYOND SPATIO-TEMPORAL DATA: DDOS ATTACK ANALYSIS USING STORYGRAPH VARIANT

7.1 Introduction

Denial of Service (DoS) attacks or Distributed Denial of Service (DDoS) attacks are carried out by the adversaries to make the host/target unavailable to the intended users in the internet. DoS attacks are launched using single computer whereas DDoS attacks are launched using multiple computers. The main goal of DoS or DDoS attacks are to render the target computer(s) unable to provide service to the legitimate users by either obstructing communication between users and the target or by exhausting the resources in the target so that it can no longer serve the intended users or by forcing the target to reset/shutdown. DDoS attacks in internet take advantage of the huge number of computers available in the internet by rendering them as bots, and then using these bots to generate 'useless' traffic directed towards the target. This many-to-one attack results in the server being inaccessible to the genuine traffic. Furthermore, the massive amount of traffic generated also makes it impossible to differentiate genuine packets from the compromised ones thus making the defense mechanisms of the target irrelevant.

DDoS has become a major network security threat because the attackers often target high profile Web sites, causing significant disruptions that affect a large population. For example, in the months of September to December 2012 some of the biggest U.S. banks suffered a series of coordinated DDoS attacks. On June 19, 2014, Facebook was shut down for 30 minutes by a DDoS attack. In recent years, the scale of DDoS has also grown significantly. For example, the DDoS attack on the web security company CloudFlare in February 2014 reached a record-breaking 400Gbps, affecting CloudFlare's entire global network.

The key defensive measures against DDoS attacks include detection, traffic classification,

traffic filtering, and post-event analysis. Although much of the defensive measures are carried out automatically by hardware and computer programs, human monitoring and analysis is still crucial because the attacks have grown increasingly more sophisticated. This is particularly true for post-event analysis, where experts are needed to identify patterns in both network traffic and timing. Here data visualization can be a useful tool that convert network data into a more intuitive and readable big-picture view.

A typical network security data visualization includes the display of the source and destination IP, volume of network traffic, and types of network traffic. However, in most network security visualizations the time dimension is not integrated well with the network traffic data. In many cases, the time is presented in unintuitive and unfamiliar methods, such as snapshots, small multiple views, animation, or special glyphs. In some cases, time is not displayed at all. This makes it difficult to study time related attack patterns.

To address this issue, we forked Storygraph to create NetTimeView. NetTimeView provides an integral view of network traffic data and time. It is particularly useful for analyzing time related patterns in post-event analysis. NetTimeView also provides two levels of traffic data visualizations that give users an "overview first, details on demand" type control. We demonstrate the effectiveness of our method through a case study using the CAIDA UCSD "DDoS Attack 2007" dataset.

7.2 Related Work

Network security visualization is an active research area and many methods have been proposed [78]. This review focuses on the visualization methods pertinent to the DDoS data. DDoS related security visualizations can be classified into the following groups based on their techniques:

- Chart based visualizations [79, 80, 81, 82, 83]. Common types of charts used for network security visualization include scatterplots, histograms, bar charts, matrix, etc.
- Graph based visualizations [84, 85, 81, 83]

- Map based visualizations [81, 86]
- Parallel coordinates [81, 87, 88, 80, 89, 90]

NetTimeView is different from previous network security methods in two areas: the handling of the time dimension and the handling of large IP space. In most previous methods, the time dimension is often presented in unnatural and unintuitive ways. For example, some methods present time as an animation [86], small multiple views [91, 79], the radius of rings [85], a vertical axis in a parallel coordinates [90], or special clock-like glyphs [80]. Some visualizations only show data for a specific point in time (i.e. a snapshot) [79, 83]. Some visualizations don't present the time dimension explicitly [81, 84, 89, 88]. The unique strength of NetTimeView is that it presents the time dimension in the most intuitive and familiar way – a horizontal timeline. Therefore, NetTimeView is particularly helpful for time related pattern analysis.

Visualizing large IP space is a common problem in network security visualization. Kintzel, et al. [80] pointed out that many security visualization techniques cannot handle large IP space. Typical methods for handling large IP space include using minimized glyphs [79, 80] or space saving layout, such as a matrix layout [82, 79, 80]. To handle large IP space, the resolution of the time dimension often has to be compromised [81]. In NetTimeView, we take a different approach. We divide each IP address into two parts: the two higher order octets and two lower order octets. First, the two higher order octets are plotted on the two vertical axes in NetTimeView. The user can click on any octet pair or a group of octet pairs to see the corresponding lower order octets in NetTimeView. This creates a multi-resolution visualization of the IP space with no compromise in the time dimension and no minimization of network event glyphs.

7.3 Data Preprocessing

Our visualization technique is based on the data extracted from the *pcap* files. Figure 7.1 shows a portion of row from such a file. Other fields in the file include packet id, segment,

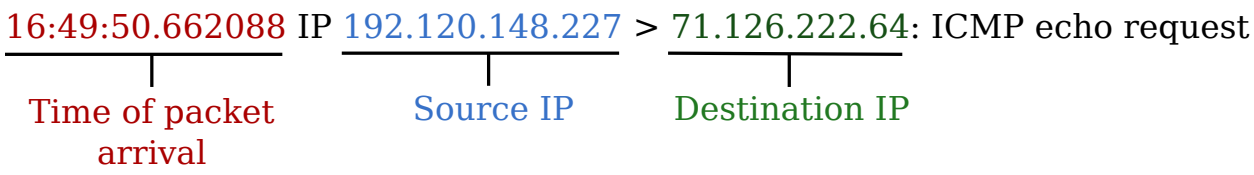


Figure (7.1) A sample row in the extracted *pcap* file showing the time of the packet arrival, source IP address, destination IP address and protocol. Other fields in the row not shown in the figure include packet id, segment, length and the payload.

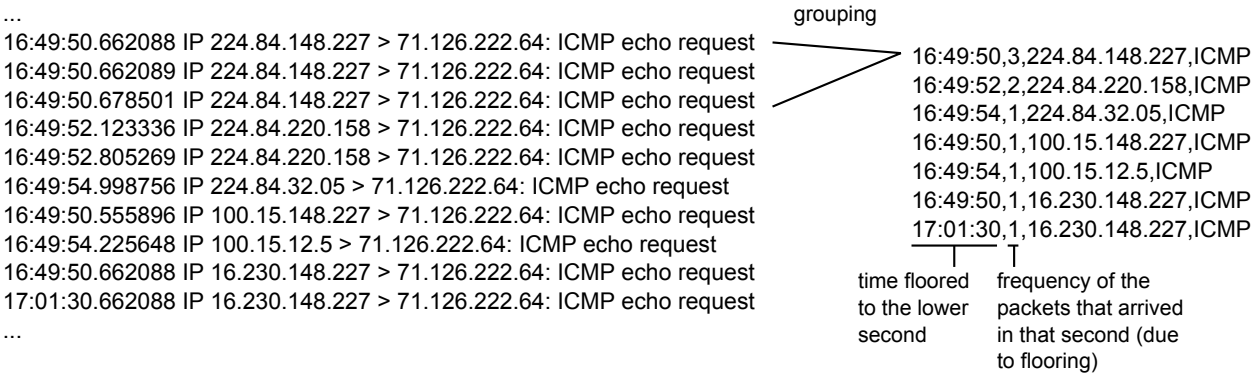


Figure (7.2) Left: The raw data. Right: The processed data with the time floored to the lowest second, number of packets in the group, source IP address and protocol. The destination IP address is truncated and so are other fields like segment, length and payload. The protocol description is also shortened for easier processing.

length and the payload of the packet. In case of general network traffic, there are multiple sources sending packets to multiple destinations, but in case of DOS attacks, all the packets are sent to a single target. Hence our visualization focuses on only the source IP address of the attack. To begin, we first extract the time of arrival, the source IP address and the protocol using *tcpdump*.

Following this, we use the floor function to bin all the milliseconds in the time stamp to the lower second. The program further counts the number of items floored and stores it along with the timestamp. This process is illustrated in Figure 7.2. In the figure, the first three IP addresses from the same source are group together by flooring the milliseconds to the lowest seconds. Similarly, the fourth and the fifth rows are also grouped together. The number of entries grouped together are stored as frequency in an adjacent column.

The cleansed data is then passed on to the visualization engine for rendering.

7.4 Method

The NetTimeView comprises of NetTimeViewU and NetTimeViewL, and is based on the philosophy of presenting 'Overview of first and details on-demand' [92]. The NetTimeViewU provides the users with the overview of the IP region of the attackers along with the time of the packet arrival and number of packets sent from that source. Users can obtain detailed information regarding particular nodes that are acting as bots within that region by clicking on the desired *IP line* in the NetTimeViewU. An IP line is a the line segment joining the coordinates in the two vertical axes. This line also serves as a timeline for the IP region in NetTimeViewU or IP address in the NetTimeViewL. The clicking on the IP line renders NetTimeViewL showing all the nodes within that region. The NetTimeViewL provides the information on the individual nodes in that particular IP region. This concept is similar to 'zooming' in computer graphics.

Both of these graphs consists of two vertical axes y_u and y_l as in case of parallel coordinates with the addition of one horizontal axis. The vertical axes pair in NetTimeViewU display the upper 16 bits, and the NetTimeViewL displays the lower 16 bits; 8 bits on each axis in both the graphs. Likewise, the horizontal axis displays time of the packet arrival.

Given packet, p arriving at time t, with the IP address $u1 : u2 : l1 : l2$, for NetTimeViewU,

The extents $u1_{max}$, $u1_{min}$ and $u2_{max}$,$u2_{min}$ and computed first and mapped to the height of the canvas H such that, $u1_{min}, u2_{min} \rightarrow 0$ and $u1_{max}, u2_{max} \rightarrow H$. Similarly t_{min} and t_{max} is mapped to the width of canvas, W such that $t_{min} \rightarrow 0$ and $t_{max} \rightarrow W$. The mapping for x is straight forward. The y coordinate is given by first computing the slope of the IP line and getting the y value at x as in Equation 7.1.

$$y = \frac{x(y_2 - y_1)}{W} + y1 \tag{7.1}$$

The same set of equations and mappings can be drawn for the NetTimeViewL by substituting $u1$, $u2$ by $l1$, $l2$.

NetTimeViewU NetTimeViewU shows the IP addresses of the attacking nodes grouping them into regions as shown in Figure 7.3. The grouping done by only considering the upper 16 bits of the IP addresses. Thus, in Figure 7.3 three IP lines can be observed for 224.84.x.x, 100.15.x.x and 16.230.x.x. Any packets arriving from these locations are plotted on the IP line. Further, the color saturation of the packets co-relates to the packet count. Higher the frequency, darker the point.

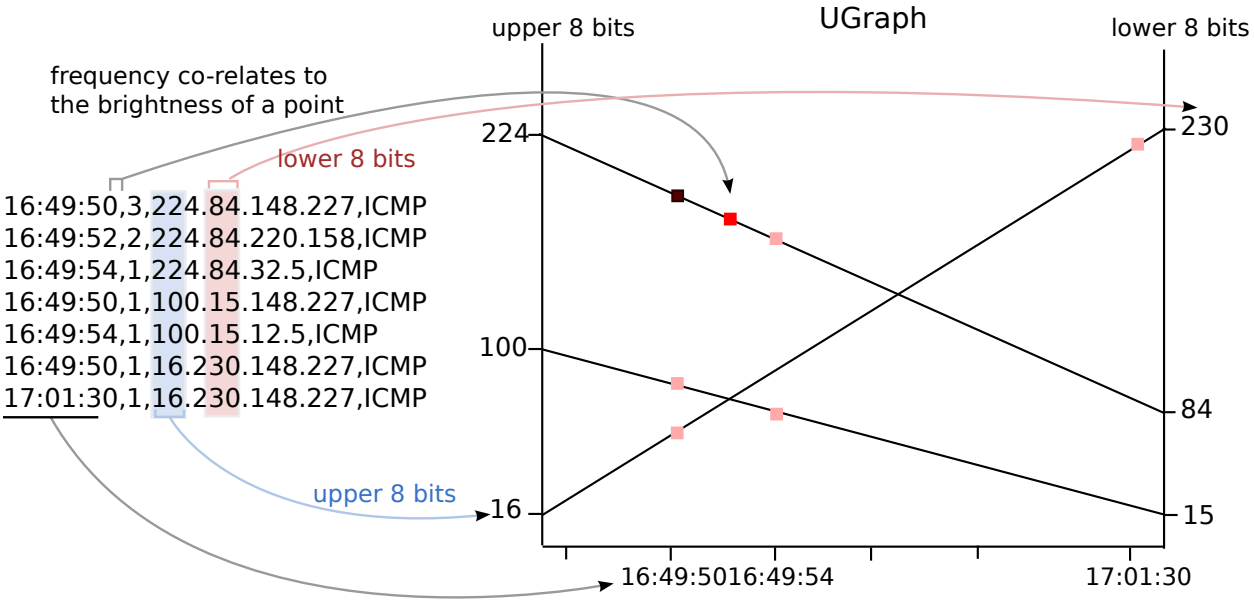


Figure (7.3) Left: The processed data. Right: NetTimeViewU drawn from the processed data. Only the upper 16 bits of the IP address is considered while drawing NetTimeViewU. In addition, the color saturation of the points directly corresponds to the frequency of packets. Higher the frequency, more saturated or darker the point.

NetTimeViewL When a IP line is selected in the NetTimeViewU, an NetTimeViewL is drawn showing all individual nodes within that IP region. Though NetTimeViewL only requires the lower 16 bits of the IP address, the information about the upper 16 bits of the IP address from the NetTimeViewU is mandatory for disambiguating the region. Figure 7.4 shows the traffic within in the region 224.84.x.x from Figure 7.3. The points have been painted with a different color in the NetTimeViewL to make it easier for the users to distinguish between the two graphs. However, the packet count information is still retained using

the color saturation of the points.

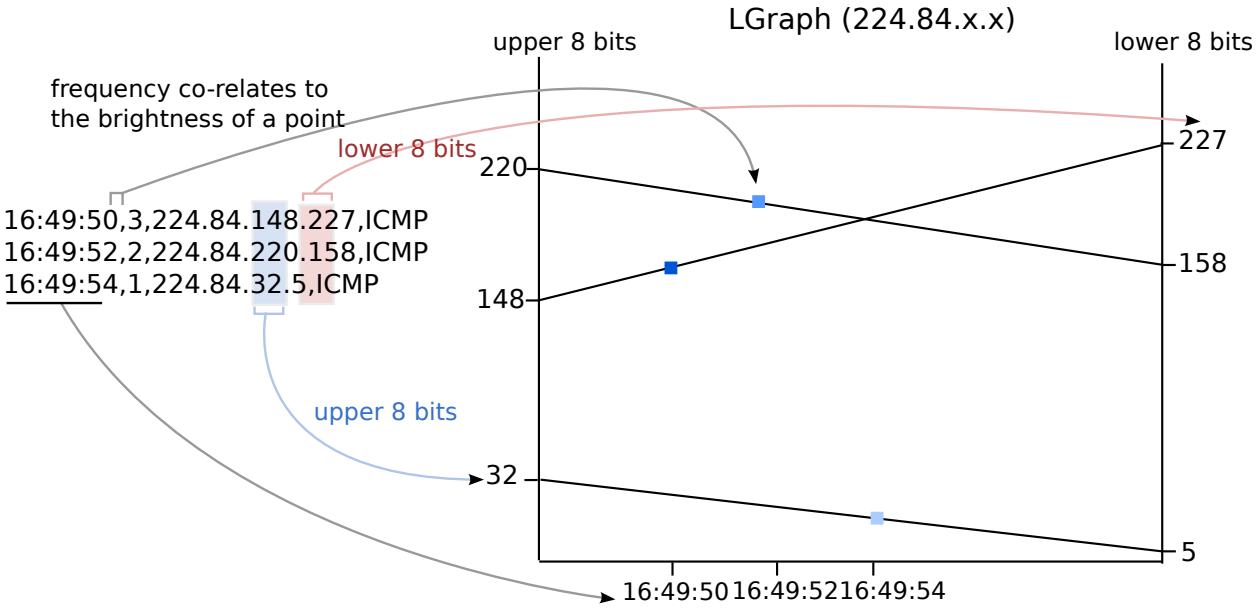


Figure (7.4) left: The processed data. Right: The NetTimeViewL drawn for the IP region 224.84.x.x Only the lower 16 bits are considered for the NetTimeViewL for a given the upper 16 bits. NetTimeViewL can also be interpreted as the zoomed view of a particular IP region.

IP Lines Depending on the the graph, the IP lines denote an IP region or IP address. IP lines play an important role in NetTimeViews as it associates packets to their corresponding regions or addresses. However, in case of DDoS attacks, potentially the number of IP lines, $n \rightarrow (255)^2$. At the same time, the arrival time of the packets are so close to each other that often times give a sense of the IP line. Thus, our visualization tool allows the users to enable/disable the IP lines during runtime. Figure 7.5, 7.6 are two examples where the IP lines have been disabled to reduce cluttering. Figure 7.7 provides and example where the IP lines have been enabled to disambiguate the IP address.

7.5 Case study

To test the effectiveness of the visualization, we implemented it on the the CAIDA DDoS dataset [93]. The contains an hour long traffic traces divided among 5-minute *pcap*

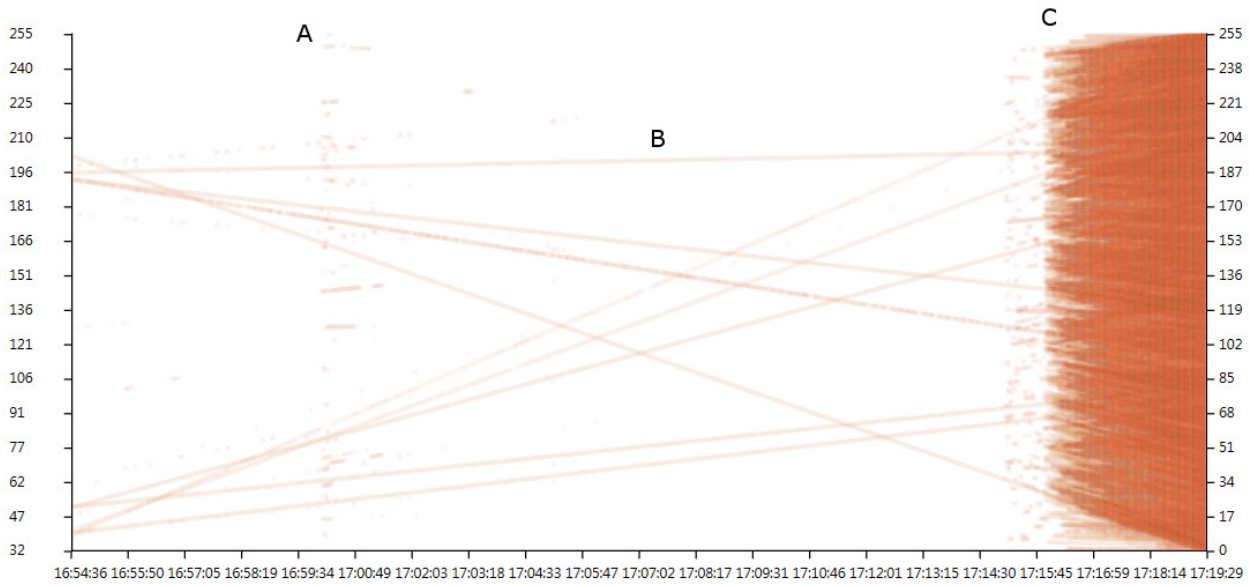


Figure (7.5) NetTimeViewU showing the start of DDoS between $17:15:45$ and $17:16:59$ marked C. normal web traffic before that. Few other patterns include the spike between $16:59:34$ and $17:00:49$ at A. The line segments marked by B denote constant traffic from these sources.

files. The total size of uncompressed dataset amounts approximately to 21GB. The dataset only includes the attack traffic to and from the victim with the payload removed from all of the packets.

Figure 7.5 shows the NetTimeViewU created from the first $500K$ entries. The first pattern that can be immediately observed is the start of DDoS which occurs between $17:15:45$ and $17:16:59$ marked by C. The number of packets all of a sudden rises sharply. Though the IP lines have been disable for cleaner image, few lines can still be seen in the image. It means that these sources were communicating constantly with the server and may be non-compromised nodes. The duration between the packets are smaller than a second, resulting them to form a line when rendered. Few subtler patterns include the spike between $16:59:34$ and $17:00:49$. This means that a number of source sent packets to this computer at the same time marked by A. Similar pattern can be observed between $17:14:30$ and $17:15:45$ before the attack commenced near C. Pattern B shows IP regions that were sending packets continuously to the server.

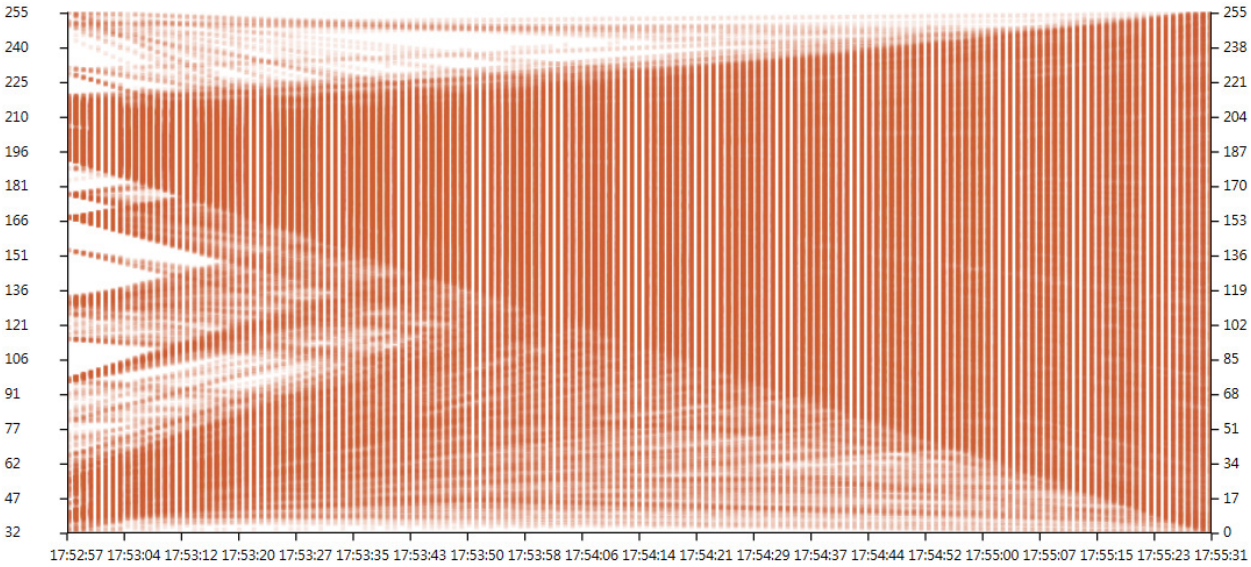


Figure (7.6) NetTimeViewU showing the end of DDoS attack. First pattern that can be observed in this figure is the high volume of traffic coming from nodes within IP regions like 192.x.x.x to 220.x.x.x. Second, there are vertical bands of higher saturation points followed by a band of lower saturation points which denote the amount of packets received per second. The thin white vertical lines are caused due the the flooring of the milliseconds during the cleaning process.

Figure 7.6 shows NetTimeViewU of the last $500K$ entries logged by the server. From the figure, a trapezoidal shape can be observed with the end points 192,220 in y_u and 0,255 in y_l. In addition, the points within this trapezoid have high saturation values. This means that a large part of traffic is coming from nodes within the IP regions, 192.x.x.x to 220.x.x.x.

The second pattern that can be seen in the figure are the alternation of light and dark bands of points. The dark points denote higher number of packets while the light points denote relatively fewer number of packets. One reason for this might be congestion control algorithms. Finding the exact cause is out of the scope of this paper.

The thin white vertical lines are introduced due to the flooring of the milliseconds during the data cleaning process.

Figure 7.7 shows the NetTimeViewL of the the IP region 229.50.x.x with IP lines enabled. It can be observed that not all nodes were active for the entire period of time and the nodes started attacking at different times.

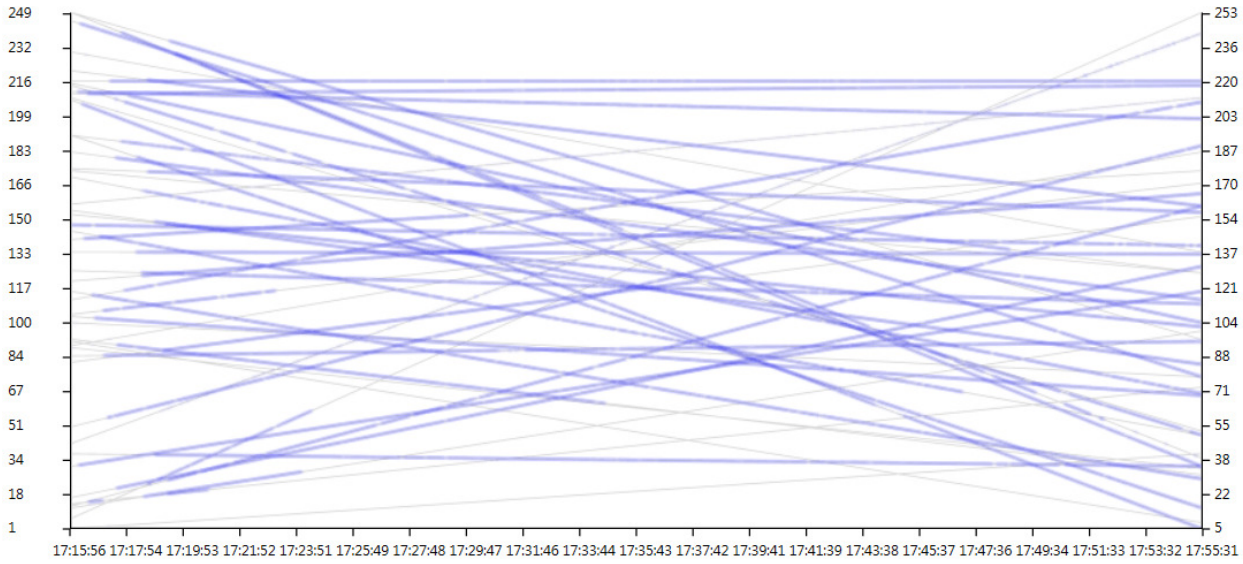


Figure (7.7) NetTimeViewL showing the traffic from the time when attack started to the time when it ended within the IP region 229.50.x.x with IP lines enabled.

7.6 Discussion

Few limitations of NetTimeView include the ambiguity caused due to the absence of IP lines and cluttering due to line crossings. As in Figure 7.5, the each packet at time marked by A could belong to a number of IP regions. In our visualization, we address this issue first by providing the filters for IP addresses and time stamps. Users can implement these filters reducing the IP range resulting in less clutter when the IP lines are turned on. Second, similar to points, we modify the alpha channel of the brushes to paint the IP lines. This results in darker IP lines for IP regions/ addresses with high traffic.

For the issue of overlapping points, our visualization prioritizes IP regions/addresses with large number of packets over the smaller ones. Thus we begin by plotting the IP regions/addresses with fewer number of points first and then plot the regions/addresses with higher number of packets. This results in the points of darker color overlapping the lighter colors.

7.7 Conclusion and Future Work

In this chapter, we presented our technique NetTimeView for post-event analysis of DDoS attacks. NetTimeView better integrates the temporal dimension with the IP views as compared to previous works, making it easier for users to analyze time related attack patterns. In addition, NetTimeView also handles the very large IP address space by providing an "overview first" using NetTimeViewU and then "details on demand" using NetTimeViewL. In the future, we plan to conduct a user evaluation of our technique. Further, we also intend to extend this technique to apply it in real-time scenarios to aid in DDoS prevention.

PART 8

EXTENDING STORYGRAPH: VISUALIZING SPATIO-TEMPORAL UNCERTAINTY

In the previous chapters, I introduced the model of Storygraph and demonstrated the benefits of using it on datasets containing precise geolocations and time. However, when applying this method to spatio-temporal data extracted from witness testimonies and field reports, we encountered problems of uncertainty in space and time. For example, our study of 511 interviews with first responders during the attack on World Trade Center (WTC) on September 11, 2001 showed that the narratives of these interviewees, who were trained to report incidences, still contained a fair amount of uncertainty in their descriptions of locations and times.

To address these issues, we extended Storygraph to visualize uncertainty. We first begin by categorizing uncertainty into two categories: (1) event uncertainty and (2) between-event uncertainty. We designed our method to distinguish and visualize these two types of uncertainty. Event uncertainty is the spatio-temporal uncertainty about the event itself, including events with poorly specified spatial and/or temporal attributes. Between-event uncertainty is the uncertainty between two precisely recorded events, which we call them *key events*. This concept is in part influenced by Hagerstrand's Time Geography [94][28][95]. After specifying the key events, the between-event uncertainties are visualized as space-time prisms between the key events. Through this process, our visualization technique can be used to study the interactions between people (or characters) in both space and time.

8.1 Classification of Uncertainty

Different classifications of uncertainty have been proposed [96][97]; however, most of these classifications are about uncertainties introduced in scientific experiments or proba-

bilistic models. In our case, uncertainties are introduced in narratives. Thus, we classify this kind of uncertainty into three categories:

Uncertainty about time and/or location of the event. This type of uncertainty is characterized by the presence of phrases denoting uncertainty before temporal or spatial description. An example is "I got there maybe around 11 am." The phrase 'maybe around' adds uncertainty to '11 am' in this example. Such uncertainties may also arise from ambiguity in language. For example, in "I was in Brooklyn when the plane hit the building," the word 'Brooklyn' does not give a precise location. We call these types of uncertainties as *event uncertainty* which can be further divided into three sub-categories:

- *Spatial uncertainty.* This category includes events that have precise time stamps but uncertain location.
- *Temporal uncertainty.* In addition to uncertain phrases (e.g. maybe, about), temporal uncertainty may come from the language itself. For example, in "I was at the station in the all day," the phrase "all day" without any modifier can refer to a wide range of time introducing uncertainty.
- *Spatio-temporal uncertainty.* This category includes events that have uncertainty in both time and location. For example, in "It was in the afternoon, I was heading south." The words 'afternoon' and 'south' are uncertain.

Uncertainty between two events. In "It was 8 in the morning I was at home. As soon as I heard about it, I reached the site at 10.", the first event ("at home") and the second event ("reached the site") are both certain. However, what happened between the two events is unknown. We call this type of uncertainty *between-even uncertainty*

Uncertainty about the even taking place. In the WTC corpus, we often encounter sentences like "I think Chief pulled me back". The word 'think' indicates an uncertainty about whether the event has ever happened. Detecting this type of uncertainty is difficult and beyond the scope of this paper. Instead, we focus only on visualizing event uncertainty and between-event uncertainty.

8.2 Event Uncertainty

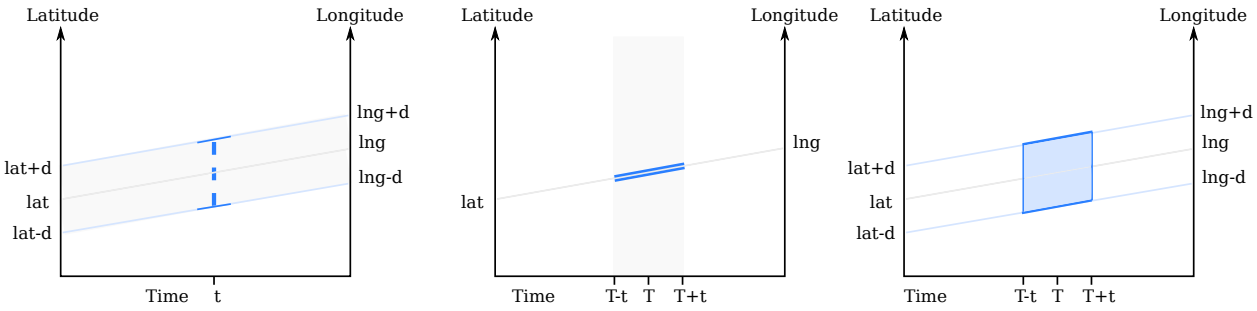


Figure (8.1) Three kinds of glyphs used to represent spatial, temporal and spatio-temporal uncertainty. Left: Dashed I-beam is used to represent spatial uncertainty. The slope of the top and bottom of the beam disambiguates the range of locations in the geographical space. Middle: parallel lines are used to denote the temporal uncertainty. Right: Box showing spatio-temporal uncertainty. The slope of the edges of the box maps to a fixed geographical area within a certain time.

In this section, we discuss the extraction and visualization of event uncertainty. To extract event uncertainty, we compiled a list of English words that may indicate location uncertainty, such as "around," "near," "close to," "maybe," "perhaps," etc. We then gave each word an uncertainty score in the range of $1 - 100$ [98][99][100]. The same process was repeated for temporal information. We extracted the named entities from WTC corpus using Stanford NER [101] and time using SuTime [102]. TARSQI [103] was used to extract the temporal sequence of the events, and locations were geocoded using Google Maps API. The results were then verified and corrected.

In the WTC corpus, we observed all three types of event uncertainties: spatial, temporal, and spatio-temporal. Some key events with precise spatio-temporal information were used as anchor events. These include the first and second plane hitting the tower, and the plane crashing into the pentagon. These events were chosen as key events because all of the interviews described more local events in reference to these global events. Examples include "When the second plane hit the tower, I was running towards Vesey," and "I was at the station when the news about the first explosion was on TV." When considering these key events in the context of the first example, the time is certain but the location is uncertain.

Additionally, in a sentence that references no key events like "When the EMS arrived at the scene, I began heading south", both location and time would be considered uncertain.

For each event, latitude, longitude, date/time, color, spatial uncertainty, and temporal uncertainty were fed to the visualization program, which then visualized the uncertainty information along with other information.

Spatial Uncertainty. Spatial uncertainty is visualized as a vertical dashed I-beam. From Chapter 4, we know that an area on a map corresponds to a line in Storygraph. The length of the I-beam is proportional to the area of possible locations. More importantly, the top and the bottom of the beam disambiguate the range of locations in geographical space. This is shown by the left sub-figure in Figure 8.1.

Temporal Uncertainty. We use sloped double lines to represent temporal uncertainty. Each double line is drawn along the location line for the corresponding event, which can be seen in the middle sub-figure in Figure 8.1. A double line indicates that the event happens at a particular location within a certain time frame. In contrast, a single solid line along the location line means that the character stayed at the specified location for a period of time. Through these representations, the two cases are visually distinct.

Spatio-temporal Uncertainty. We use a semi-transparent box to visualize spatio-temporal uncertainty, which means both location and time are uncertain. The sloped top and bottom sides of the box indicate the range of locations while the vertical sides of the boxes shows the temporal bound. The box is drawn as semi-transparent to prevent glyph occlusion. This is shown by the right sub-figure in Figure 8.1.

Figure 8.6 shows this concept applied to the events extracted from WTC corpus.

8.3 Between-event uncertainty

The purpose of visualizing between-event uncertainty is to display the space-time constraints between two key events. Any activity takes place within a certain span of time and a certain geographical region. Individuals participating in these activities have to trade time for space or vice versa. For example, during a workday lunch hour a person could walk

to a nearby restaurant for a longer meal or drive to distant restaurant for a shorter meal. Visualizing between-event uncertainty can assist planning, scheduling, analyzing possible overlapping in people's activities.

Our between-event visualization technique is partially based on Hagerstrand's Time Geography, a conceptual framework which focuses on constraints and trade-offs in the allocation of time among activities in space [95]. However, Time Geography is a map based 3D visualization. Therefore it suffers from the typical problems associated with 3D visualizations, such as 3D occlusion and difficulty of navigation. Besides, space and time are not well integrated in Time Geography. Our work is an attempt to address these issues.

8.3.1 Space-time paths and space-time prisms

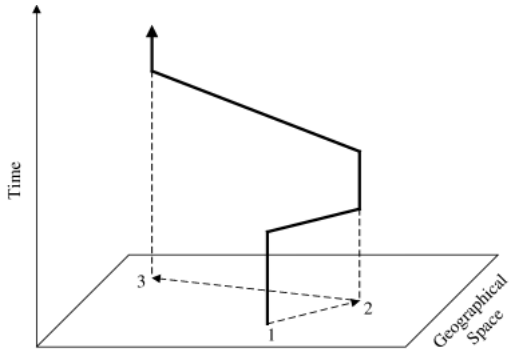


Figure (8.2) An example of space time path adapted from [95]. Space time paths trace the movement of an individual moving from one location to another. Space-time paths also show the amount of time spent at a location by the individual before moving to the next location.

We adapted two important concepts from Time Geography: space-time paths and space-time prisms. Space-time path traces the movement of a character in space and time. Figure 8.2 shows an example of a space time path adapted from [95]. The base plane is the geographical space and the orthogonal axis is time. In this example, an individual travels from location 1 to 2, spends some time at 2 and then moves on to 3. The time and location of the starting or end point are known as *control points* or *key events*. The straight line segments connecting two control points are known as *path segments*. Path segments are represented by straight line segments for simplicity [104][105]. In our earlier work, we adapted

the concept of space-time paths in Storygraph using storylines[106]. Here, Storylines become space-time paths, connecting two consecutive key events via dotted line segment.

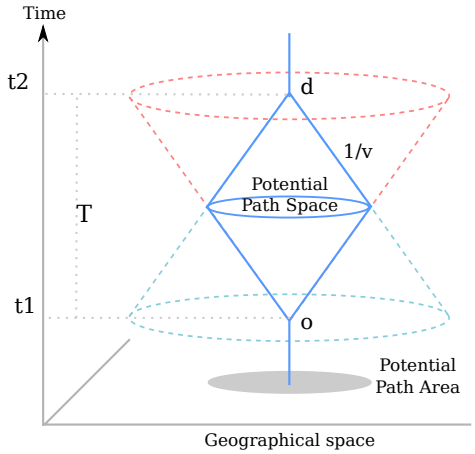


Figure (8.3) Space-time prism. In this figure, the individual is at location o at $t1$ needs to be at the same location d at $t2$ (o origin of travel). (S)he has the time budget of T. The red dotted cone shows the possible path space starting from o with the maximum velocity, v. Similarly the blue dotted cone shows the path space towards d. The intersection of these two cones gives the potential path space under the given time budget T. The potential path area is shown by the gray area on the geographic space.

Space-time prisms extend space time paths to create a 3D space consisting of all the possible routes an individual can take while moving from one point to another. This space is known as the *potential path space*. The prism between $t1$ and $t2$ in Figure 8.3 demonstrates this concept. The slope of the edges of this prism is determined by the inverse of maximum velocity. That is, the possible paths are constrained by the maximum velocity of the individual, a fixed time frame, and fixed destinations. In our implementation, the maximum velocity is set by the user.

If an individual is at origin, o, at t_o and needs to reach destination, d, at t_d, the time budget is $T = t_d - t_o$. The path space from the origin under the time budget is shown by the red inverted dotted cone. This space shows all the possible paths and all the possible locations that can be reached within the time budget with maximum velocity v. Let this region be denoted by $R_o(T)$. Similarly, the blue dotted cone shows the path space towards d under the time budget. This 3D space gives all the locations from where d can be reached

under time T. Let this region be $R_d(T)$. The intersection of these cones give the potential path space for individual traveling from o to d [107]. Hence,

$$R_{od}(T) = R_o(T) \cap R_d(T) \tag{8.1}$$

The projection of the space time prism on the geographical space, as shown by a gray circle in the figure, shows all the possible locations that the user can reach. This area is called the *potential path area*.

Given all the control points within a specific time window, τ, the construction of space-time prism requires the destination d to lie within the $R_o(T)$ and vice versa. Stating it formally,

$$\forall o, d \in \phi_\tau : (o \cap R_d(T) \neg\emptyset) \wedge (d \cap R_o(T) \neg\emptyset) \tag{8.2}$$

In Time Geography, space-time paths and space-time prisms are generally drawn inside a 3D space-time cube [48] (Figure 8.3). In our work, space-time paths and space-time prisms are drawn on Storygraph in a 2D view.

8.3.2 Visualizing between-event uncertainty

Storygraph draws space-time prisms based on Equations 8.1 and 8.2. From Chapter 4, we know that an area in the geographical space is mapped to a line in Storygraph. Thus starting from a location, $o(\alpha, \beta)$, at $t0$ and taking a snapshots of the potential path area at each time step we get a set of areas sequentially increasing at the rate of the velocity. The top sub-figure in Figure 8.4 shows an individual at point (α, β) at $t0$ and his/her possible path area after each time step $t1 - t5$. The figure simplifies the drawing of the potential path areas by representing them with squares rather than circles. The bounding of the actual potential path by squares introduces some uncertainty itself [108] but greatly simplifies the drawing and the calculations.

Hence, if the time step, $\Delta t \rightarrow 0$, then conical region $R_o(T)$ would be reduced to a

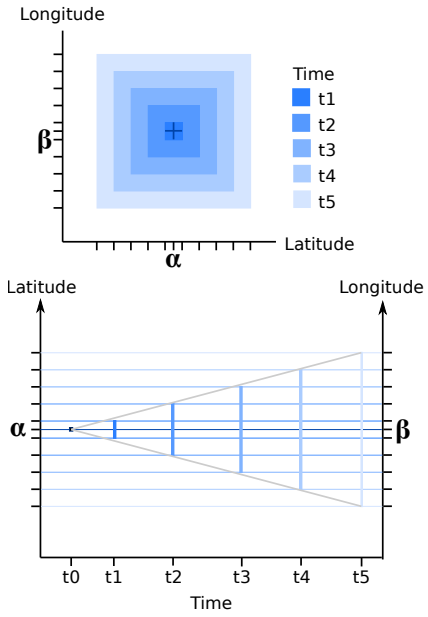


Figure (8.4) Above: Starting from the origin of travel, α, β, at $t0$, the potential path areas in each time step $t1 - t5$ (assuming a certain velocity and regular time intervals). Below: The same data plotted in Storygraph. Each potential path area is mapped to a line segments in Storygraph. For continuous time, this would result in an area enveloped by two gray lines. The slope of the gray lines is equal to the maximum velocity.

triangular region in Storygraph. This region is shown by the area enveloped by two gray lines in the bottom sub-figure in Figure 8.4.

Figure 8.5 shows the result of mapping the space-time prism in Figure 8.3 in Storygraph. The mapping process resembles the drawing of the space-time prism inside the space-time cube. Given two control points, maximum velocity and a time budget, these parameters are plugged into Equation 8.2 to check whether the control points satisfy this criteria. If the criteria is satisfied, we compute the extents (lat_{max}, lng_{max}) and (lat_{min}, lng_{min}) of the $R_o(T)$ with the following sets of equations,

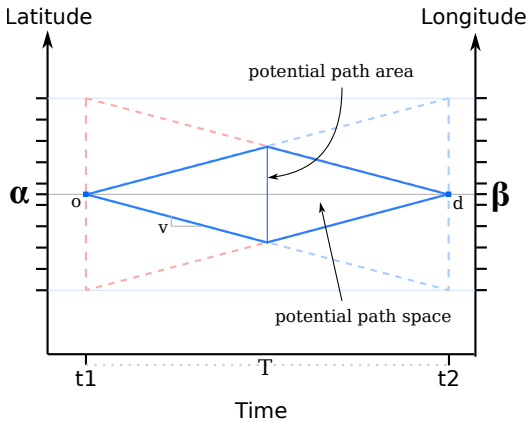


Figure (8.5) The space-time prism shown in Figure 8.3 drawn in Storygraph.

$$lat_{max} = max_{lat_r}[\sqrt{(lat_r - lat_o)^2 + (lng_r - lng_o)^2} = vT] \tag{8.3}$$

$$lng_{max} = max_{lng_r}[\sqrt{(lat_r - lat_o)^2 + (lng_r - lng_o)^2} = vT] \tag{8.4}$$

$$lat_{min} = min_{lat_r}[\sqrt{(lat_r - lat_o)^2 + (lng_r - lng_o)^2} = vT] \tag{8.5}$$

$$lng_{min} = min_{lng_r}[\sqrt{(lat_r - lat_o)^2 + (lng_r - lng_o)^2} = vT] \tag{8.6}$$

Similarly, the extents for the $R_d(T)$ is calculated. Finally, $R_{od}(T)$ is obtained from the intersection of these regions.

8.3.3 Intersections of prisms in Storygraph

Space-time paths and prims are both based on the movement data of characters. Given a dataset containing the movement data of two or more individuals, it is likely that the space-time prisms will overlap. However, since Storygraph does not preserve event proximity (Chapter 4), it is important to note that these overlaps may not necessarily mean that these prisms intersect in geographical space.

Hence, given a point p and a prism $R_a(T)$ in the Storygraph, we first establish the conditions for a valid point-prism intersection. Building on this, we then present the validity of intersection between two prisms.

Let $R_a(T) : R_a(T) = R_o(T) \cup R_d(T)$, be all the possible locations that the individual can travel within the time budget T with a maximum velocity v. Then following cases for point-prism intersection could arise:

1. The point is not inside the prism but the location line is inside $R_a(T)$. This case implies that the event occurred within the geographical bounds but the individual may not have been involved in the event due to the travel constraints.

2. The point is inside the prism but the location line is not inside $R_a(T)$. This implies that the event occurred within the time span, T, but at some other location $\notin R_a(T)$.

3. The point is inside the prism and location line is inside $R_a(T)$. This is the only case where the individual could have been involved in the event.

Theorem 8.3.1 *For a valid point-prism intersection, the point should be inside the prism and the location line should lie inside* $R_a(T)$.

Proof Assume that this is not a valid intersection. It means that the point representing the event is either spatially or temporally incorrect. This is temporally incorrect because for a point to lie inside the prism, it has to occur within the time budget. This is spatially incorrect since $R_a(T)$ defines the maximum distance an individual can travel at within a time T.

Hence, given two prisms, $P1$ and $P2$, the prism-prism intersection is only valid if there exists a point p on location line l such that $l \in R_a^{P1}(T) \wedge l \in R_a^{P2}(T) \wedge p \in P1 \cap P2$.

8.4 Case Study: WTC 9/11

In the immediate aftermath of the attacks in New York on September 11, 2001, the NYC Fire Department convened a task force to interview first responders to the affected areas. These 511 interviews, conducted in the two months following the attacks, were later released by the New York Times. Each interview was conducted by staff from the New York Fire

Department assigned to the task force and ran anywhere from 8-20 minutes with the aim to elicit from first responders their activities on September 11. The language of the reports is typical of event interviews and oral histories. Despite having a population with high area knowledge and normalized reporting practices, locations and times were predominately referred to referentially. Known individuals seen by the interviewee are named, but most are either not named or referred to solely by rank. The primary reason to visualize this data is to enable historians and investigators to identify accurate and inaccurate information and to allow for more ready recognition of corroborating evidence. When viewed as a corpus rather than separate interviews, it becomes possible to identify overlaps in the reported events of the witnesses. The challenge posed to this task by the referential language usage of the witnesses is pervasive in oral history and other investigatory work reliant on interviewing.

Event Uncertainty Visualization. Time, location, and characters (or people) in this corpus were extracted using Java code and the aforementioned natural language processing tools. Each event was given an uncertainty score using the method described in Section 8.2. We first drew a Storygraph without uncertainty information (Figure 5.11). In this figure, key events – such as when the first and second plane hit and when the towers collapsed – are shown by $t1 - t4$. There are many co-occurring events around $15:00$ hrs, but the causes of these patterns are not yet clear.

Next, we plotted the storylines of four fire fighters before the South Tower collapsed with event uncertainty (Figure 8.6). It should be noted that two storylines crossing does not necessarily mean the two characters encounter each other; rather, it only means that two people were moving in directions diagonal to each other. One limitation of using uncertainty glyphs is that they might result in occlusion and ambiguity for large datasets. When the dataset is large, the bigger glyphs (e.g. the ones representing spatio-temporal uncertainty) could occlude the smaller ones.

Between-event Uncertainty Visualization. Figure 8.7 visualizes the between-event uncertainty for two victims: Father Judge and Chief Ganci. The space-time prisms in Storygraph enable users to see the possibilities of individuals encountering each other between key

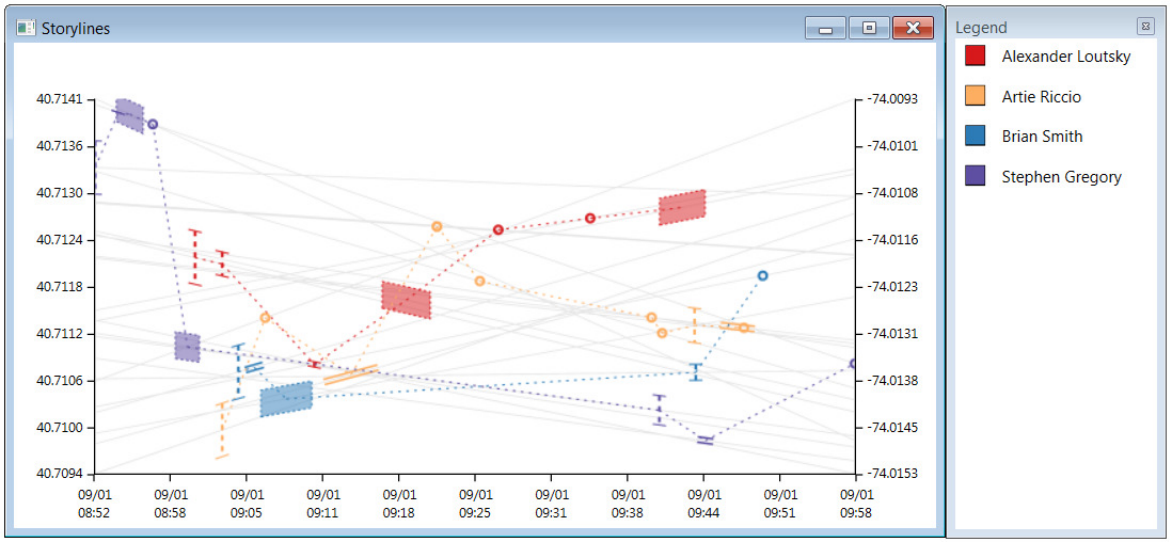


Figure (8.6) Storylines of four firefighters before the second tower collapsed along with event uncertainty. The dashed vertical I-beam shows the spatial uncertainty. The slope of the top and bottom portion of the beam shows the possible range of locations. The parallel lines show temporal uncertainty. The boxes represent spatio-temporal uncertainty and the circles show certain events.

events. There are two patterns in this figure: (1) the prisms are only present between some key events, and (2) some prisms overlap. The first pattern indicates that locations of the two events are too far apart in that it would be impossible for a person to cover that distance at the maximum velocity. It does not necessarily mean that part of the story is false; rather, it may be the result of missing information between two events or uncertainty in the events themselves. From overlapping prisms (from Theorem 8.3.1), we can also deduce that Chief Ganci and Father Judge might have encountered each other within that time frame and region.

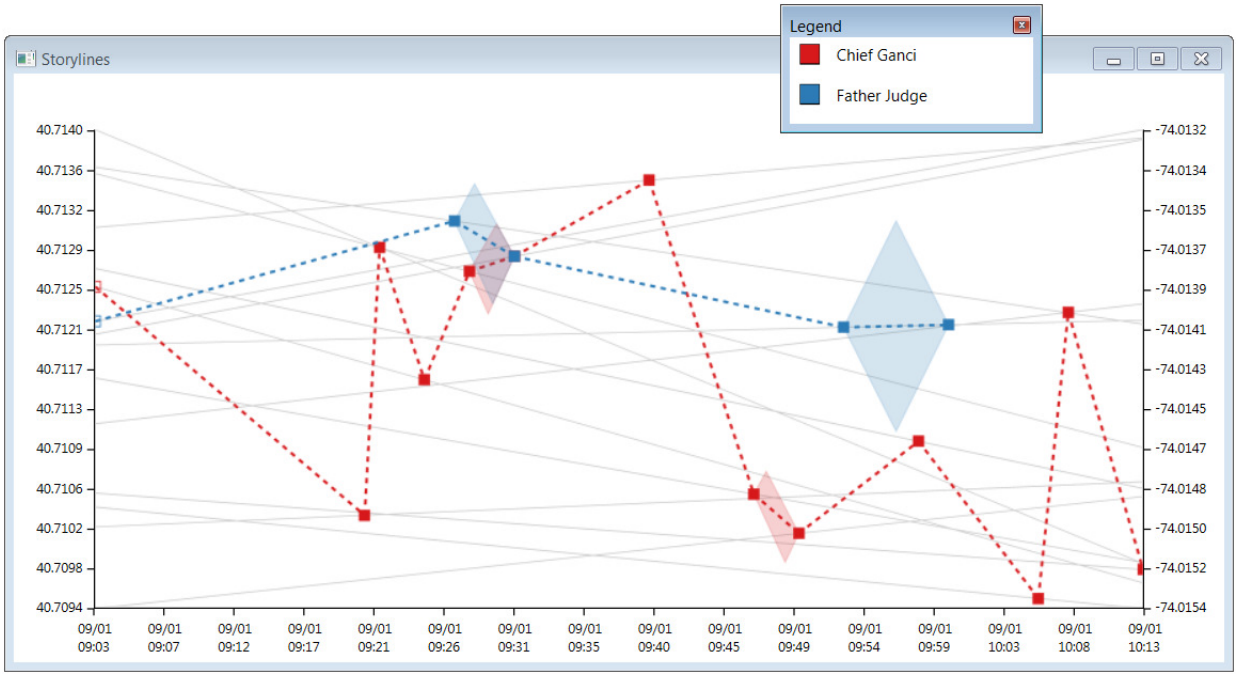


Figure (8.7) Storylines of WTC victims Chief Ganci and Father Judge. Father Judge was officially identified to be the first victim of the incident. Only a few key points have prisms between them. For other key points, the distance between them cannot be travelled within the given time at a velocity set by the user. This could either mean missing data points, change in velocity, or data reporting error.

PART 9

EXTENDING STORYGRAPH: VISUAL CLUTTERING REDUCTION IN STORYGRAPH

Often, many visualization tools that work well with small to medium datasets, fail to produce good results when the size of data increases. Though Storygraph has high tolerance, at some point, the noise generated from visual clutter overshadows the information. To address this issue, in this chapter, I present our work on the application of edge bundling techniques in Storygraph. Our goal of bundling the edges is to to reduce the visual cluttering and help users identify patterns and trends in their data sets. In addition, we also investigate the criteria where visual clustering produces visually superior results.

For big data sets, the number of location lines is very large, which often leads to large number of line crossings. Too many line crossings creates visual cluttering that can significantly reduce the readability of the data visualization. One way to address this issue is to cluster the locations before feeding them into the visualization, using techniques such as K-means clustering. However, finding the optimal data clusters is often computationally expensive. Instead, we propose a visual clustering technique that bundles location lines together using forces. This method reduces visual cluttering by reducing the number of line crossings and grouping location lines by proximity. We demonstrate the benefits of our method using two large datasets: Cablegate data and D3.js data.

9.1 Related Work

A common problem in visualizing big data sets is visual cluttering. Too many visual glyphs will reduce readability of the visualization, making it difficult to identify patterns and relationships. There are two methods to deal with this problem. One method is data clustering, which happens early in the data preparation stage. The other method is visual

clustering, which happens later in the visualization stage.

Data clustering methods attempt to reduce a big data set into data groups (clusters) and then plot the data groups instead of the individual data items. Data clustering has its own drawbacks. First, a data clustering method may introduce errors into the data. Patterns that are visible in the reduced data set may be introduced by the data clustering method itself and may not reflect the relationship in the original data sets. Second, multidimensional data clustering is a very difficult problem and effectiveness methods are often hard to find.

Visual clustering attempts to reduce the number of visual glyphs to improve the readability of the visualization. Our work falls into this category. Since our work is related to visual clustering in parallel coordinates, we first review the visual clustering techniques used in the parallel coordinates, and then review some relevant clutter reduction techniques in graphs (node-link diagrams) using force-based edge bundling techniques.

Parallel coordinates is a technique for visualizing large, multi-dimensional database. It uses vertical axes to represent multiple dimensions and "poly-lines" to represent data entries in the database. The visual clustering methods for parallel coordinates can be roughly divided into two groups: numerical approach and physical approach. Numerical approaches attempt to detect line crossings or closely bundled lines through numerical means. For example, Palmas [109] used density based clustering for parallel coordinates.

Physical approaches attempt to use simulated physical forces to untangle line crossings or closely bundled lines. For example, Guo et al. [110] used attractive as well as repulsive forces on the edges depending on the user criteria. Zhou, et al. [111] also used physical forces and an overall energy function to facilitate line clustering. Our method falls into this category.

Another way to classify the visual clustering methods is based on how they display the "line clusters". Some methods replace the line clusters (line bundles) with different shapes. For example, Palmas [109] rendered the bundles using polygons for faster rendering. Andrienko and Andrienko [112] used envelope and ellipse plots. Other methods maintain the poly-lines by rendering them as curves. For example, in [113] [114], Heinrich et al. clustered

the lines in the parallel coordinates by replacing the poly-lines with piecewise continuous Bezier curves. Zhou, et al. [111] rearranged poly-lines as curves. Our method also falls into this category.

Edge clustering is also widely implemented in graphs (node-link diagrams). A few notable works include hierarchical edge bundling [115], force-directed edge bundling [116], skeleton-based edge bundling [117], winding roads [118], bundling of directional links for network data [119] and image based edge bundles [120].

Out method is different from previous methods in two ways. First, in previous methods, the poly-line itself is the visual glyph that represents data. In our case, both the location lines and event marks on the location lines are the visual glyphs. Therefore our method is more sophisticated than previous works and has higher data density. Second, our method preserves location proximity for the location lines during the clustering. This poses further constraints on how the lines can be reordered vertically. In previous methods, such constraints are not taken into consideration. Our method is the first to address these issues.

9.2 Method

To reduce cluttering, we apply simulated physical forces on the location lines so they naturally form bundles. Let E be the set containing all the location lines. Given two location lines, $P \in E$ and $Q \in E$, and K segments, $K-1$ charged particles are placed evenly on each of these locations lines.

$$x_{p_i} = \frac{T}{|K|} \cdot i \tag{9.1}$$

$$y_{p_i} = \frac{(\beta - \alpha)}{T} x_{p_i} + \alpha \tag{9.2}$$

for $0 \leq i \leq K$

The end points p_0 and p_K remain intact. The forces are then computed for each of these particles and repeated for M iterations.

For any particle p_i in P, the total force exerted is given by the equation,

$$F_{p_i} = F^e_{p_i} + F^m_{p_i} \tag{9.3}$$

where $F^e_{p_i}$ is the electrostatic force and $F^m_{p_i}$ is the mechanical force.

$$F^e_{p_i} = \sum_{e_i \in E} \frac{1}{||p_i - e_i||} \tag{9.4}$$

Similarly the spring forces on the particle is given by

$$F^m_{p_i} = \sum_{i \neq j} ||p_i - p_j|| \tag{9.5}$$

9.2.1 Simplifications

Unlike node-link diagrams in [116], our visualization has a structure similar to parallel coordinates. Taking advantage of this structure, force calculations can be greatly simplified. Graphs generally consider all-to-all interactions for force based layouts and bundling. But in our method, the position of the end points are fixed. Since the distance between the vertical axes is constant, all the end points of the location lines have the same x coordinate in the plane and only y coordinates differ.

Let $p_i(x, y)$ be the coordinate of the i^{th} particle on the location line P. Thus, $\forall Q \in E$ and $Q \neq P$. This simplification greatly speeds up force calculation.

$$p_i(x) = q_i(x) \tag{9.6}$$

where $0 \leq i \leq K$.

Considering only adjacent particles Previous literatures [115][116] have shown that incorporating the effects of particles farther away while computing forces does not produce a drastic difference in the rendered image. Hence, we only consider the adjacent

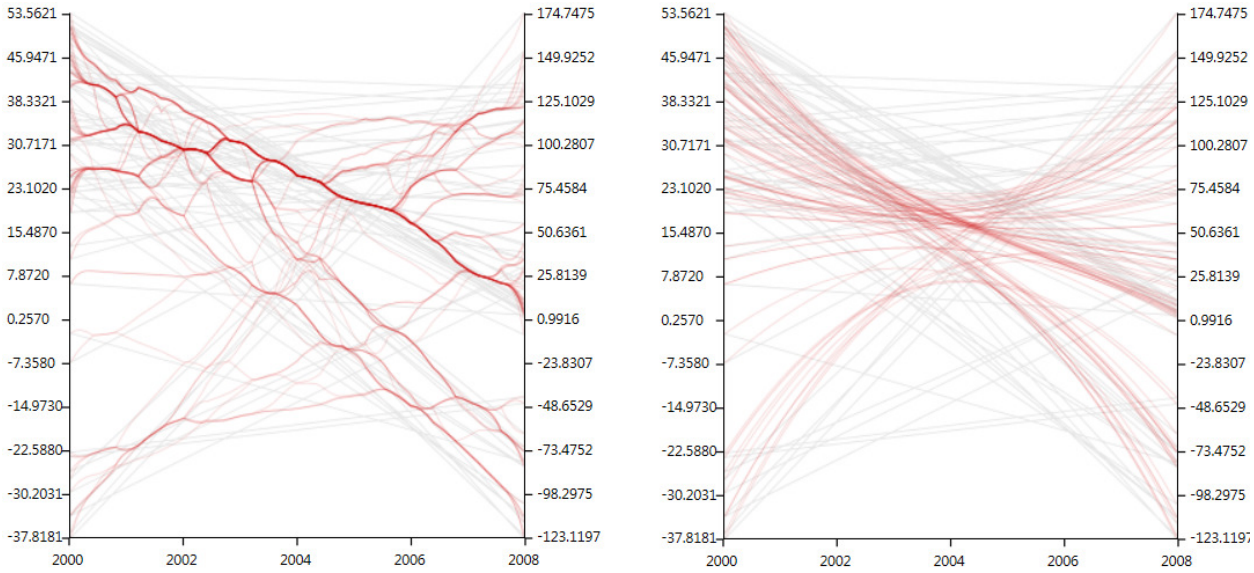


Figure (9.1) Two visualizations of the Cablegate dataset (events hidden) to demonstrate the bundling of the edges without application of any constraints. The curves in red are the bundled location lines while gray straight line segments show the unmodified location lines. In the left sub-figure, each line was divided into 60 segments while in the right, the line was divide into 3 segments i.e. $K = 60$ and $K = 3$ respectively.

particles while calculating the mechanical force. Thus, (9.5) gets reduced to,

$$F^m_{p_i} = (||p_{i-1} - p_i|| + ||p_i - p_{i+1}||) \tag{9.7}$$

Considering only the vertically aligned particles Calculating the effect of forces from all other particles on a particular particle is computationally intensive. In addition, formation of a cluster attracts other particles towards it, skewing the location lines considerably. As a result the events plotted on these location lines also gets cluttered. Therefore, to compute $F^e_{p_i}$, for particle p_i we ignore the effect of all other particles q_j where $i \neq j$. From (9.4) and (9.6), we get

$$F^e_{p_i} = \sum_{q \in E, 0 \leq i \leq K} \frac{1}{||p_i - q_i||} \tag{9.8}$$

9.2.2 Constraints

Figure 9.1 demonstrates the effect of segmentation on the location lines as the value of K increases. Both of the visualizations were drawn from the same dataset and in both cases, events have not been drawn to highlight the bundling. It can be observed that without placing any constraints in the edges and computing the compatibility of the edges, as K increases, so does the branching in the bundled curves. Branching adds to the ambiguity making it hard to trace the origin and the end of location line.

Many parallel coordinates prefer bundling technique as shown in the right sub-figure with $K = 3$ as it tends to make the clusters more prominent in the overall figure [121][122]. However, in our method, it leads to location lines bundled in the center, resulting in unnecessary overlap of the events. This problem can be observed in the two location lines from -22.5880 to -48.6529 which are more readable when not bundled.

To address this issue, we apply a few constraints on the bundling.

Non-uniform stiffness of the edges Since each location line contains a number of events, we set the stiffness of the location lines directly proportional to the frequency of the events in that location. Thus, (9.7) becomes,

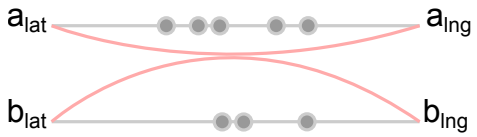


Figure (9.2) The location lines with the higher number of events are more stiff than the location lines with fewer events. The gray straight lines are the original lines and the red curves show the same lines after the forces have been applied. The gray points on the location lines represent events. Since (a_{lat}, a_{lng}) has more number of events than (b_{lat}, b_{lng}), the latter is more curved.

$$F_{p_i}^m = k_p \cdot (||p_{i-1} - p_i|| + ||p_i - p_{i+1}||) \tag{9.9}$$

where $k_p = c \cdot eventFrequency(p)$

Varying the stiffness results in the location lines with lower frequency being attracted to the lines with higher frequency as shown in Figure 9.2.

Distance between the locations Two location lines P and Q can only be bundle if $||P - Q|| < \psi$, where ψ is a cut-off distance set by the user as shown in Figure 9.3.

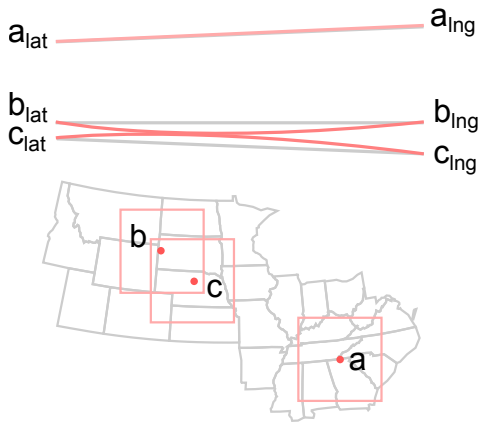


Figure (9.3) Top: The location lines within a certain cut-off distance are allowed to bundle together. This is demonstrated by (b_{lat}, b_{lng}) and (c_{lat}, c_{lng}) above. Since (a_{lat}, a_{lng}) is beyond the cut-off distance from both locations, it does not participate in the bundling. Bottom: Same set of (hypothetical) events plotted on the map. The squares represent the cut-off distance.

In the map, this is equivalent to drawing a square with sides equal to ψ and the location as the center. This is shown in the bottom sub-figure. Only those events within the square contribute the to the force calculation of that location line. In the illustration b and c are within the bounding box of each other. Thus only b influences in the force calculation of c and vice versa.

Any arbitrary shape on the map at time t, with maximum latitude lat_{max} and longitude lng_{max}, and minimum values lat_{min} and lng_{min} is represented by a vertical line segment at time t in our visualization. with the end points of the segment being (lat_{max}, lng_{max}) and (lat_{min}, lng_{min}). This transformation is equivalent to drawing a rectangular bounding box around the shape on the map and then plotting the bounding box in our visualization.

Angle between the location lines Two location lines P and Q with the interior angle between them $\theta = arccos\frac{P.Q}{|P||Q|}$, $\theta < \alpha$ where α is a threshold angle also set by the user. Previous studies have shown that the lines with the intersection angle, α closer to right angle, are easier to read than with smaller angles [123] [124] [122]. Figure 9.4 demonstrates this concept with four location lines $a - d$. The angle between bc, being smaller than the threshold results in these location lines getting bundled while location lines a and d remain unchanged.

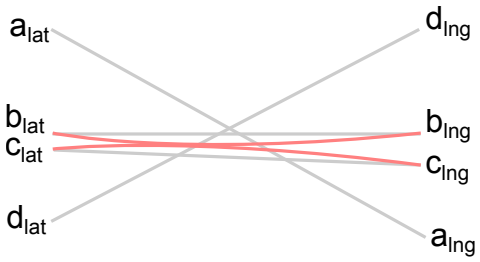


Figure (9.4) Four location lines $a - d$ where the interior angle between $bc < \alpha$ resulting in bc to be bundled.

9.2.3 Colors and Alpha channel

The locations in the Cablegate as well as D3js dataset consisted of places from all over the globe. Since assigning a color to each location would result in a lot of noise, we chose to assign a color for each continent. A qualitative color scheme was picked from Color Brewer for the task [125].

The modification of the spring constant k_p makes the higher frequency edges to be stiffer. However, we observed that when the two location lines are very close (with a small α), brushing alone adds to the confusion on which location had higher number of events. Thus to tackle this problem, we modified the alpha channel of the location lines. Our implementation compares to the work by Artero et al. [126] for parallel coordinates.

Given two alpha values α^u_{min} and α^u_{max} by the user, we first compute the normalized

alpha value of the edge,

$$\alpha_n = (\sum_{e \in Events} e)/max_e \tag{9.10}$$

where $e = 1$ if e is in the location and 0 otherwise and max_e is the largest number of events in one location. We map this value to the user given range as

$$\alpha = (\frac{\alpha_n - \alpha^u_{min}}{\alpha^u_{max} - \alpha^u_{min}})^p \tag{9.11}$$

where p is the tuning parameter. The tuning parameter highlights the high frequency edges over the lower frequencies.

9.3 Case Studies

We demonstrate our method with two big data sets: Cablegate data and D3.js data.

9.3.1 Cablegate

The Cablegate dataset (also known as the United States diplomatic cables leak) consists of cables between US State Department and 274 US embassies or diplomatic missions around the world. The data set has $251,287$ cables that amount to $261,276,536$ words [64]. According to Wikileaks, this is "the largest set of confidential documents ever released into the public domain". Amongst other parameters, the dataset consists of the location of the origin, the location of the destination, and the time the message was sent. This is a good datasets for spatial-temporal data visualization because it contains both location and time information.

In the Cablegate dataset, all the destinations and many of the origins are encoded. The remaining locations are in plain text. We filtered out the encoded locations and obtained a dataset of about $20,000$ entries. The locations were geocoded using the Google Maps API. We only used the fields 'origin' and 'date' for the visualizations.

Figure 9.5 and Figure 9.6 show two visualizations of this dataset. The first one shows the visualization with no edge bundling, and no brushing of the location lines as well as

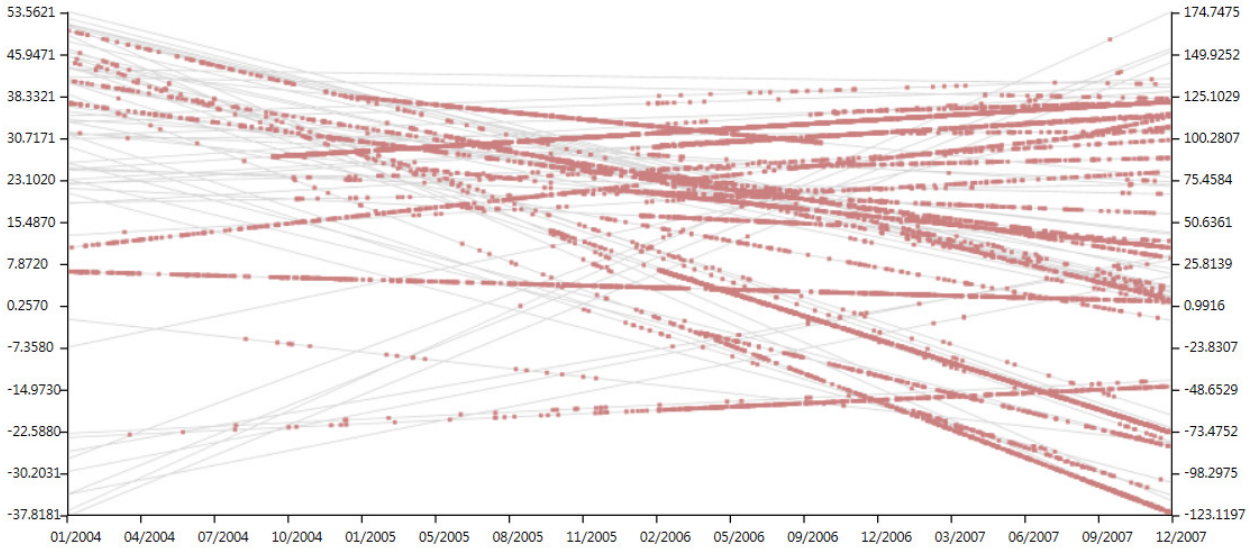


Figure (9.5) Visualization of the Cablegate material without edge bundling. The location lines and the events have been assigned using uniform color.

events. It can be observed that there are many location line crossings in Figure 9.5 than in the improved version. However, they are not as aggressively bundled as in Figure 9.1. Annotations $A - F$ show the location lines that are more prominent than others. These are locations from where more cables are originated. A few lines only have events from the midpoint onwards, such as location lines ending in E and F in Figure 9.6. The main reasons for this kind of patterns are either missing data or inactivity at that location.

9.3.2 D3js

The D3js dataset contains the software commit history of the popular Javascript library called d3.js from 2010 to 2013. The dates and the commits were extracted from the Git based commit history. The locations of these commits are the locations of the developers who committed them, which we extracted from their public GitHub profile. The authors without location information were ignored in the process. The authors without location information typically had less than 5 commits on average. The dataset consisted of $4,200$ data entries.

Figure 9.7 shows a side by side comparison of two visualizations generated from the

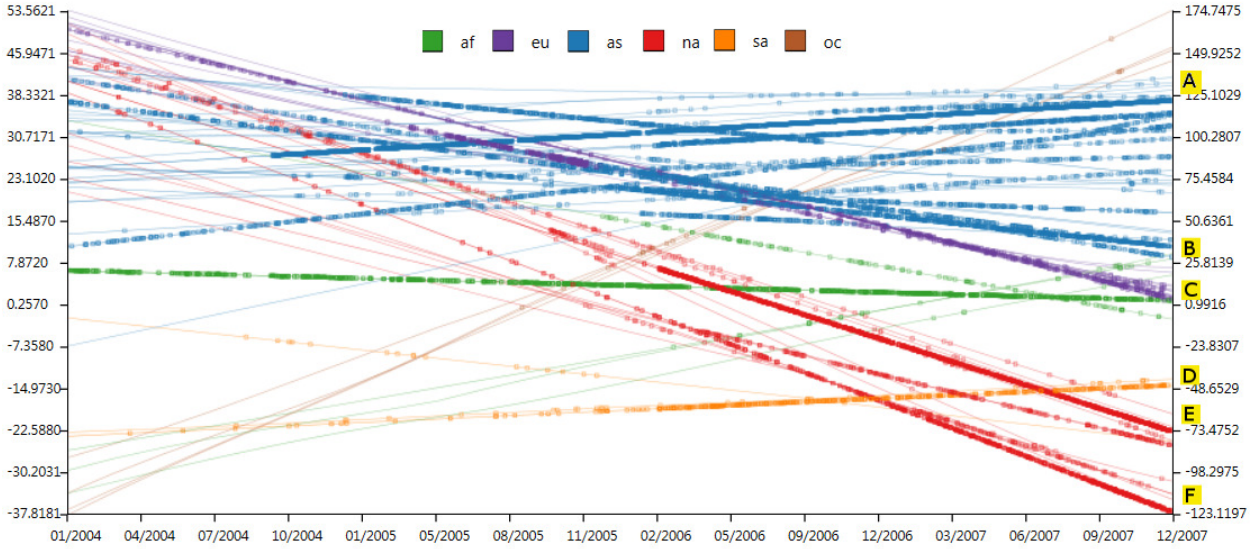


Figure (9.6) Visualization of the Cablegate data with edges bundled and constraints applied. The edges as well as the events on those edges have been assigned the same color according to the continent they belong. Annotations $A - F$ show few locations which were actively sending cables.

D3js dataset. The left image shows the original visualization and the right one shows the improved visualization with edge-bundling. It can be observed that the image on the right has more edges bundled together enhancing the readability.

9.4 Implementation

We implemented our visualization method using the .Net 4.5 framework for the controls and Direct2D for rendering. The curved edges were drawn as Bezier curves. The rendering pipeline consisted of the following stages: edge segmentation, computation compatibility matrix, calculation of forces, and rendering. The main bottlenecks in this pipeline are the computation of the compatibility matrix and the force computation. The computability matrix requires an all-to-all comparison for all the unique edges, resulting in computations in the order of $O(n^2)$. We achieved some speed-up through parallel computing. The force calculation is in the order of $O(n_c^2 \times K \times M)$, where n_c is the number of unique compatible edges, K is the number of segments, and M is the total number of iterations. It should

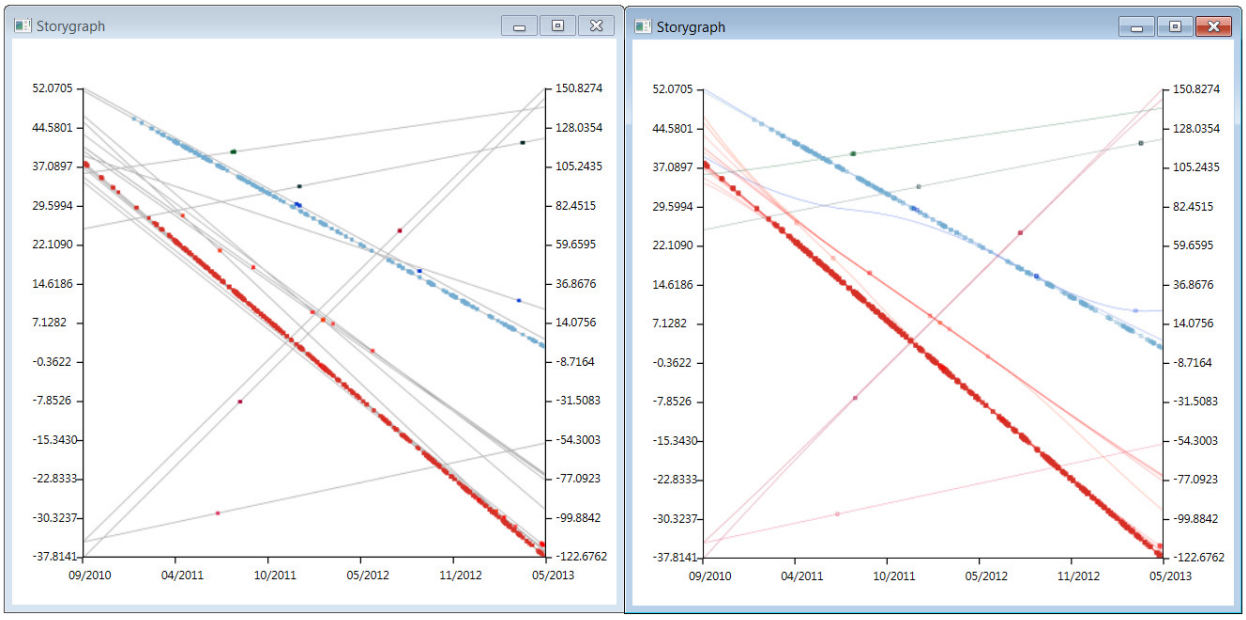


Figure (9.7) (Left) Visualization based on the commit history of the Javascript library D3.js without bundling. (Right) Same graph with the edges bundled.

be noted that as $nc \to n$, the order of the computation will increase to $O(n^2 \times K \times M)$, significantly affecting the performance of the algorithm. Both the segmentation and the rendering are of order $O(N)$. In addition, vector graphics is used to speed up the rendering.

9.5 Discussion

In the process of plotting these two datasets we observed that increasing the bundling also increases the chances of glyphs overlapping like (as location line ending in B and eu line ending in C) in Figure 9.6. Our program addresses this issue by providing users with filtering options so that the users can filter the range of latitude, longitude and date. Each time a user applies a filter, the graph is rendered again with the modified extents, thus minimizing the crossings. The same technique can be used for overlapping glyphs that are not caused by bundling, such as (af location line ending in C and na line ending in D).

We also noticed that edge bundling in our visualization is more effective for large number of events taking place at small clustered regions. This difference can be observed in Figure 9.7 in comparison to Figure 9.6. For the D3js dataset, most commits were either from California,

USA (clusters painted in red) or from London, UK (clusters painted in blue). Though commits were from multiple locations within California, the location lines are clustered to form a single bundle. Similar observations can be made for London. In contrast, the Cablegate dataset consists of distinct locations that are far apart from each other. Therefore the number of bundled location lines is small.

PART 10

CONCLUSION

In this dissertation, I showcased my novel visualization technique called Storygraph. Storygraph addresses one of the most crucial questions in Information visualization research area - 'How do we present time and space together?'. It overcomes the generic difficulties that the 3D diagrams run into like occlusion and cluttering by transforming the 3D space into 2D, consisting of two parallel vertical axes and one horizontal axis. By doing this, although many we were able to discover and publish previously undocumented patterns in the data, deciphering information from Storygraph needs prior training.

Storygraph is not as intuitive as map, which have been around for hundreds of centuries. In some cases it might take longer to decode the information from Storygraph as compared to timeline. However, Storygraph provides unique insight to the data compared to any other visualization, as it preserves both spatial and temporal contexts. Storygraph when supplemented with maps, and timeline (or vice versa) provides a complete picture for better spatio-temporal data analysis.

Like any other visualization techniques, more precise the data, better the visualization. However, many sources produce data with high amount of uncertainty. To address this, I added the uncertainty module in Storygraph. This module, based on Time-Geography, visualizes continuity and uncertainty in time and space. It can also plot space-time prisms thus allowing powerful visual queries like whether two people met at a certain place within a set span of time. A setback of space time prisms on Storygraph is that if the analyst is not careful enough, its easy to generate false positives. As one of my future works, I'm working on minimizing these visual false positives.

As the size of the data grows, so does the cluttering in any visualization. The cluttering at some point generates so much noise that it overshadows the patterns. Storygraph also

suffers from this trend. Therefore, to minimize the cluttering, I also implemented an edge bundling technique on Storygraph. Compared to normal graphs, edge bundling in Storygraph was tricky since more bundling resulted in reduced readability. It also caused ambiguity in which lines did the points belong to. I found that bundling produces better results when there are geographic clusters with a good amount of distance between these clusters. Datasets with just one big cluster in some cases failed to produce visible results.

In the latter chapters, I applied Storygraph to datasets which have an abstract location like IP addresses and was able to produce some good results leading to publications. Storygraph could also be a viable option for visualizing the data from natural sciences like genetics where it is not possible to draw physical maps. I want to explore this further in future.

Lastly, I want to test the Storygraph with actual users and domain experts to observe where does it perform better than the existing spatio-temporal visualization techniques. I want to use this study to improve and extend the current version of Storygraph.

REFERENCES

[1] Wikipedia. (2013) Anscombe's quartet. [Online]. Available: http://en.wikipedia.org/wiki/Anscombe%27s_quartet

[2] A. Treisman, "Preattentive processing in vision," *Computer vision, graphics, and image processing*, vol. 31, no. 2, pp. 156–177, 1985.

[3] E. R. Tufte, *The visual display of quantitative information.* Cheshire, CT, USA: Graphics Press, 1986.

[4] Google.com, "Google chart api." [Online]. Available: https://developers.google.com/chart/interactive/docs/gallery

[5] Google. (2013) Google earth api. [Online]. Available: https://developers.google.com/earth/

[6] W. Aigner, S. Miksch, W. Mller, H. Schumann, and C. Tominski, "Visualizing time-oriented dataa systematic view," *Computers and Graphics*, vol. 31, no. 3, pp. 401 – 409, 2007.

[7] E. Hajnicz, *Time structures: formal description and algorithmic representation.* Springer, 1996, no. 1047.

[8] A. U. Frank, "Different types of" times" in gls," *Spatial and temporal reasoning in geographic information systems*, p. 40, 1998.

[9] W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski, "Visual methods for analyzing time-oriented data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 1, pp. 47 –60, jan.-feb. 2008.

[10] J. J. Van Wijk, "Image based flow visualization," in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 745–754.

[11] F. Reinders, F. H. Post, and H. J. Spoelder, "Visualization of time-dependent data with feature tracking and event detection," *The Visual Computer*, vol. 17, no. 1, pp. 55–71, 2001.

[12] S. Havre, E. Hetzler, P. Whitney, and L. Nowell, "Themeriver: Visualizing thematic changes in large document collections," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 8, no. 1, pp. 9–20, 2002.

[13] C. Tominski, J. Abello, and H. Schumann, "Axes-based visualizations with radial layouts," in *Proceedings of the 2004 ACM symposium on Applied computing*. ACM, 2004, pp. 1242–1247.

[14] C. Tominski, P. Schulze-Wollgast, and H. Schumann, "3d information visualization for time dependent data on maps," in *Information Visualisation, 2005. Proceedings. Ninth International Conference on*. IEEE, 2005, pp. 175–181.

[15] A. V. Moere, "Time-varying data visualization using information flocking boids," in *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*. IEEE, 2004, pp. 97–104.

[16] J. J. Van Wijk and E. R. Van Selow, "Cluster and calendar based visualization of time series data," in *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*. IEEE, 1999, pp. 4–9.

[17] W. Aigner, S. Miksch, B. Thurnher, and S. Biffl, "Planninglines: Novel glyphs for representing temporal uncertainties and their evaluation," in *Information Visualisation, 2005. Proceedings. Ninth International Conference on*. IEEE, 2005, pp. 457–463.

[18] H. Doleisch, M. Mayer, M. Gasser, R. Wanker, and H. Hauser, "Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system," in *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*. Eurographics Association, 2004, pp. 91–96.

[19] C. Tominski and W. Aigner. The timeviz browser: A visual survey of visualization techniques for time-oriented data.

[20] D. Fisher, A. Hoff, G. Robertson, and M. Hurst, "Narratives: A visualization to track narrative events as they develop," in *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, oct. 2008, pp. 115 –122.

[21] D. Luo, J. Yang, M. Krstajic, W. Ribarsky, and D. A. Keim, "Eventriver: Visually exploring text collections with temporal references," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 1, pp. 93–105, 2012.

[22] K. Vrotsou, J. Johansson, and M. Cooper, "Activitree: Interactive visual exploration of sequences in event-based data using graph similarity," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 945 –952, nov.-dec. 2009.

[23] J. Zhao, F. Chevalier, and R. Balakrishnan, "Kronominer: using multi-foci navigation for the visual exploration of time-series data," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 1737–1746.

[24] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan, "Exploratory analysis of time-series with chronolenses," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2422 –2431, dec. 2011.

[25] M. Krstajic, E. Bertini, and D. Keim, "Cloudlines: Compact display of event episodes in multiple time-series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2432 –2439, dec. 2011.

[26] P. Nguyen, K. Xu, R. Walker, and B. Wong, "Schemaline: Timeline visualization for sensemaking," in *Information Visualisation (IV), 2014 18th International Conference on*, July 2014, pp. 225–233.

[27] L. Bagrow, *History of cartography*. Transaction Publishers, 1964.

[28] N. Thrift, "An introduction to time-geography." Geo Abstracts, University of East Anglia, 1977.

[29] J. T. Coppock and D. W. Rhind, "The history of gis," *Geographical information systems: Principles and applications*, vol. 1, no. 1, pp. 21–43, 1991.

[30] M. Goodchild, R. Haining, and S. Wise, "Integrating gis and spatial data analysis: problems and possibilities," *International Journal of Geographical Information Systems*, vol. 6, no. 5, pp. 407–423, 1992.

[31] L. Anselin, "Interactive techniques and exploratory spatial data analysis," *Geographical Information Systems: principles, techniques, management and applications*, vol. 1, pp. 251–264, 1999.

[32] L. Anselin, I. Syabri, and Y. Kho, "Geoda: an introduction to spatial data analysis," *Geographical analysis*, vol. 38, no. 1, pp. 5–22, 2006.

[33] M. S. Rosenberg and C. D. Anderson, "Passage: pattern analysis, spatial statistics and geographic exegesis. version 2," *Methods in Ecology and Evolution*, vol. 2, no. 3, pp. 229–232, 2011.

[34] Google. (2013) Google maps api. [Online]. Available: https://developers.google.com/maps/

[35] Microsoft. (2013) Bing maps api. [Online]. Available: http://www.microsoft.com/maps/

[36] OpenStreetMap.org. (2013) Open street map api. [Online]. Available: http://wiki.openstreetmap.org/wiki/API_v0.6

[37] E. R. Tufte, "Envisioning information," *Optometry & Vision Science*, vol. 68, no. 4, pp. 322–324, 1991.

[38] S. M. Powsner and E. R. Tufte, "Graphical summary of patient status," *The Lancet*, vol. 344, no. 8919, pp. 386–389, 1994.

[39] W. S. Cleveland, "Coplots, nonparametric regression, and conditionally parametric fits," *Lecture Notes-Monograph Series*, pp. 21–36, 1994.

[40] C. Brunsdon, "The comap: exploring spatial pattern via conditional distributions," *Computers, environment and urban systems*, vol. 25, no. 1, pp. 53–68, 2001.

[41] C. Brunsdon, J. Corcoran, and G. Higgs, "Visualising space and time in crime patterns: A comparison of methods," *Computers, Environment and Urban Systems*, vol. 31, no. 1, pp. 52 – 75, 2007.

[42] P. Harris, C. Brunsdon, and M. Charlton, "The comap as a diagnostic tool for non-stationary kriging models," *International Journal of Geographical Information Science*, vol. 27, no. 3, pp. 511–541, 2013.

[43] A. Asgary, A. Ghaffari, and J. Levy, "Spatial and temporal analyses of structural fire incidents and their causes: A case of toronto, canada," *Fire Safety Journal*, vol. 45, no. 1, pp. 44 – 57, 2010.

[44] C. Plug, J. C. Xia, and C. Caulfield, "Spatial and temporal visualisation techniques for crash analysis," *Accident Analysis and Prevention*, vol. 43, no. 6, pp. 1937 – 1946, 2011.

[45] M. Jern and J. Franzen, ""geoanalytics" - exploring spatio-temporal and multivariate data," in *Proceedings of Tenth International Conference on Information Visualization*, july 2006, pp. 25 –31.

[46] A. Fredrikson, C. North, C. Plaisant, and B. Shneiderman, "Temporal, geographical and categorical aggregations viewed through coordinated displays: a case study with highway incident data," in *Proceedings of the 1999 workshop on new paradigms in*

information visualization and manipulation in conjunction with the eighth ACM internation conference on Information and knowledge management, ser. NPIVM '99. New York, NY, USA: ACM, 1999, pp. 26–34.

[47] R. Maciejewski, S. Rudolph, R. Hafen, A. Abusalah, M. Yakout, M. Ouzzani, W. Cleveland, S. Grannis, and D. Ebert, "A visual analytics approach to understanding spatiotemporal hotspots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 2, pp. 205 –220, march-april 2010.

[48] M.-J. Kraak, "The space-time cube revisited from a geovisualization perspective," in *Proc. 21st International Cartographic Conference*, 2003, pp. 1988–1996.

[49] N. Andrienko, G. Andrienko, and P. Gatalsky, "Exploratory spatio-temporal visualization: an analytical review," *Journal of Visual Languages & Computing*, vol. 14, no. 6, pp. 503–541, 2003.

[50] P. Gatalsky, N. Andrienko, and G. Andrienko, "Interactive analysis of event data using space-time cube," in *Proceedings. Eighth International Conference on Information Visualisation*, july 2004, pp. 145 – 152.

[51] C. Tominski, P. Schulze-Wollgast, and H. Schumann, "3d information visualization for time dependent data on maps," in *Proceedings. Ninth International Conference on Information Visualisation*, july 2005, pp. 175 – 181.

[52] G. Andrienko, N. Andrienko, M. Mladenov, M. Mock, and C. Politz, "Identifying place histories from activity traces with an eye to parameter impact," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 5, pp. 675–688, May 2012.

[53] T. V. Landesberger, S. Bremm, N. Andrienko, G. Andrienko, and M. Tekusova, "Visual analytics for categoric spatio-temporal data," in *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST 2012)*, October 2012, pp. 183–192.

[54] P. 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, July 2009.

[55] 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.

[56] A. Thudt, D. Baur, and S. Carpendale, "Visits: A spatiotemporal visualization of location histories," in *EuroVis 2013: Proceedings of the Eurographics Conference on Visualization*, 2013.

[57] U. Demsar and K. Virrantaus, "Spacetime density of trajectories: exploring spatiotemporal patterns in movement data," *International Journal of Geographical Information Science*, vol. 24, no. 10, pp. 1527–1542, 2010.

[58] A. Murray, Y. Liu, S. Rey, and L. Anselin, "Exploring movement object patterns," *The Annals of Regional Science*, pp. 1–14, 2012, 10.1007/s00168-011-0459-z.

[59] D. Orellana, A. K. Bregt, A. Ligtenberg, and M. Wachowicz, "Exploring visitor movement patterns in natural recreational areas," *Tourism Management*, vol. 33, no. 3, pp. 672 – 682, 2012.

[60] B. Lenntorp and P. Hort, *Paths in space-time environments: A timegeographic study of movement possibilities of individuals.* Royal University of Lund, Department of Geography Lund, 1976, vol. 44.

[61] S.-L. Shaw and H. Yu, "A gis-based time-geographic approach of studying individual activities and interactions in a hybrid physicalvirtual space," *Journal of Transport Geography*, vol. 17, no. 2, pp. 141 – 149, 2009, ¡ce:title¿ICT and the Shaping of Access, Mobility and Everyday Life¡/ce:title¿. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0966692308001324

[62] Guardian.co.uk. (2010, July) Afghanistan war logs 2003-2010. [Online]. Available: http://www.guardian.co.uk/world/datablog/2010/jul/25/wikileaks-afghanistan-data

[63] S. Rogers. (2010) Wikileaks afghanistan files: download the key incidents as a spreadsheet. [Online]. Available: http://www.theguardian.com/world/datablog/2010/jul/25/wikileaks-afghanistan-data

[64] Wikileaks. (2010) Cablegate's cables. [Online]. Available: http://www.wikileaks.org/cablegate.html

[65] L. N. U. O. Programme. (2008) Annual report 2007. [Online]. Available: http://www.uxolao.org/Download%20files/UXO%20LAO%202007%20Annual%20Report.pdf

[66] B. Miller, A. Shrestha, J. Derby, J. Olive, K. Umapathy, F. Li, and Y. Zhao, "Digging into human rights violations: Data modelling and collective memory," in *Big Data, 2013 IEEE International Conference on*. IEEE, 2013, pp. 37–45.

[67] E. University. (2012) The trans-atlantic slave trade database. [Online]. Available: http://www.slavevoyages.org/tast/index.faces

[68] S. Weber, *The success of open source*. Cambridge Univ Press, 2004, vol. 368.

[69] J. M. Gonzalez-Barahona, G. Robles, R. Andradas-Izquierdo, and R. A. Ghosh, "Geographic origin of libre software developers," *Information Economics and Policy*, vol. 20, no. 4, pp. 356–363, 2008.

[70] K. Crowston, K. Wei, J. Howison, and A. Wiggins, "Free/libre open-source software development: What we know and what we do not know," *ACM Computing Surveys (CSUR)*, vol. 44, no. 2, p. 7, 2012.

[71] Y. Takhteyev and A. Hilts, "Investigating the geography of open source software through github," 2010. [Online]. Available: http://www.takhteyev.org/

[72] G. Robles and J. M. Gonzalez-Barahona, "Geographic location of developers at sourceforge," in *Proceedings of the 2006 international workshop on Mining software repositories*. ACM, 2006, pp. 144–150.

[73] B. Heller, E. Marschner, E. Rosenfeld, and J. Heer, "Visualizing collaboration and influence in the open-source software community," in *Proceedings of the 8th Working Conference on Mining Software Repositories*, ser. MSR '11. New York, NY, USA: ACM, 2011, pp. 223–226.

[74] X. Ben, S. Beijun, and Y. Weicheng, "Mining developer contribution in open source software using visualization techniques," in *2013 Third International Conference on Intelligent System Design and Engineering Applications (ISDEA)*, 2013, pp. 934–937.

[75] A. Kuhn, D. Erni, P. Loretan, and O. Nierstrasz, "Software cartography: Thematic software visualization with consistent layout," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 22, no. 3, pp. 191–210, 2010.

[76] M. Ogawa and K.-L. Ma, "Software evolution storylines," in *Proceedings of the 5th international symposium on Software visualization*. ACM, 2010, pp. 35–42.

[77] A. Sarma and A. Hoek, "Palantir: Increasing awareness in distributed software development," in *Proceedings of 25th International Conference on Software Engineering (ICSE)*, 2003, pp. 444–454.

[78] H. Shiravi, A. Shiravi, and A. Ghorbani, "A survey of visualization systems for network security," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 8, pp. 1313–1329, Aug 2012.

[79] K. Lakkaraju, W. Yurcik, and A. J. Lee, "Nvisionip: netflow visualizations of system state for security situational awareness," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. ACM, 2004, pp. 65–72.

[80] C. Kintzel, J. Fuchs, and F. Mansmann, "Monitoring large ip spaces with clockview," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, ser. VizSec '11. New York, NY, USA: ACM, 2011, pp. 2:1–2:10. [Online]. Available: http://doi.acm.org/10.1145/2016904.2016906

[81] G. Conti, K. Abdullah, J. Grizzard, J. Stasko, J. Copeland, M. Ahamad, H. L. Owen, and C. Lee, "Countering security information overload through alert and packet visualization," *Computer Graphics and Applications, IEEE*, vol. 26, no. 2, pp. 60–70, March 2006.

[82] H. Koike and K. Ohno, "Snortview: visualization system of snort logs," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. ACM, 2004, pp. 143–147.

[83] P. Ren, J. Kristoff, and B. Gooch, "Visualizing dns traffic," in *Proceedings of the 3rd international workshop on Visualization for computer security*. ACM, 2006, pp. 23–30.

[84] J. Zhang, G. Yang, L. Lu, M. Huang, and M. Che, "A novel visualization method for detecting ddos network attacks," in *Visual Information Communication*, M. L. Huang, Q. V. Nguyen, and K. Zhang, Eds. Springer US, 2010, pp. 185–194. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-0312-9_12

[85] J. Pearlman and P. Rheingans, "Visualizing network security events using compound glyphs from a service-oriented perspective," in *VizSEC 2007*. Springer, 2008, pp. 131–146.

[86] Google, "Digital attack map," 2014. [Online]. Available: http://www.google.com/ideas/projects/digital-attack-map/

[87] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen, "Real-time and forensic network data analysis using animated and coordinated visualization," in *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*. IEEE, 2005, pp. 42–49.

[88] T. Nunnally, P. Chi, K. Abdullah, A. S. Uluagac, J. A. Copeland, and R. Beyah, "P3d: A parallel 3d coordinate visualization for advanced network scans," in *Communications (ICC), 2013 IEEE International Conference on.* IEEE, 2013, pp. 2052–2057.

[89] H. Choi, H. Lee, and H. Kim, "Fast detection and visualization of network attacks on parallel coordinates," *computers & security*, vol. 28, no. 5, pp. 276–288, 2009.

[90] S. Tricaud and P. Saadé, "Applied parallel coordinates for logs and network traffic attack analysis," *Journal in computer virology*, vol. 6, no. 1, pp. 1–29, 2010.

[91] J. R. Goodall and M. Sowul, "Viassist: Visual analytics for cyber defense," in *Technologies for Homeland Security, 2009. HST'09. IEEE Conference on.* IEEE, 2009, pp. 143–150.

[92] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on.* IEEE, 1996, pp. 336–343.

[93] CAIDA, "The caida ucsd "ddos attack 2007" dataset," 2007. [Online]. Available: http://www.caida.org/data/passive/ddos-20070804_dataset.xml

[94] T. Hägerstrand, "Reflections on "what about people in regional science?"," in *Papers of the Regional Science Association*, vol. 66, no. 1. Springer, 1989, pp. 1–6.

[95] H. J. Miller, "A measurement theory for time geography," *Geographical analysis*, vol. 37, no. 1, pp. 17–45, 2005.

[96] K. Potter, P. Rosen, and C. R. Johnson, "From quantification to visualization: A taxonomy of uncertainty visualization approaches," in *Uncertainty Quantification in Scientific Computing.* Springer, 2012, pp. 226–249.

[97] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on.* Ieee, 2008, pp. 376–385.

[98] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, "The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty," *ACM Computing Surveys (CSUR)*, vol. 12, no. 2, pp. 213–253, 1980.

[99] M.-C. de Marneffe, C. D. Manning, and C. Potts, "Veridicality and utterance understanding," in *Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing*, ser. ICSC '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 430–437. [Online]. Available: http://dx.doi.org/10.1109/ICSC.2011.10

[100] R. Sauri and J. Pustejovsky, "Factbank: a corpus annotated with event factuality," *Language Resources and Evaluation*, vol. 43, no. 3, pp. 227–268, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10579-009-9089-9

[101] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 363–370.

[102] A. X. Chang and C. Manning, "Sutime: A library for recognizing and normalizing time expressions." in *LREC*, 2012, pp. 3735–3740.

[103] M. Verhagen, I. Mani, R. Sauri, R. Knippen, S. B. Jang, J. Littman, A. Rumshisky, J. Phillips, and J. Pustejovsky, "Automating temporal annotation with tarsqi," in *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2005, pp. 81–84.

[104] D. Pfoser and C. S. Jensen, "Capturing the uncertainty of moving-object representations," in *Advances in Spatial Databases*. Springer, 1999, pp. 111–131.

[105] J. Moreira, C. Ribeiro, and J.-M. Saglio, "Representation and manipulation of moving points: an extended data model for location estimation," *Cartography and Geographic Information Science*, vol. 26, no. 2, pp. 109–124, 1999.

[106] A. Shrestha, Y. Zhu, B. Miller, and Y. Zhao, "Storygraph: Telling stories from spatio-temporal data," in *Advances in Visual Computing*. Springer, 2013, pp. 693–702.

[107] T. Neutens, N. Weghe, F. Witlox, and P. Maeyer, "A three-dimensional network-based spacetime prism," *Journal of Geographical Systems*, vol. 10, no. 1, pp. 89–107, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10109-007-0057-x

[108] K. Brodlie, R. A. Osorio, and A. Lopes, "A review of uncertainty in data visualization," in *Expanding the Frontiers of Visual Analytics and Visualization*. London: Springer, 2012.

[109] G. Palmas, M. Bachynskyi, A. Oulasvirta, H. P. Seidel, and T. Weinkauf, "An edge-bundling layout for interactive parallel coordinates," in *Pacific Visualization Symposium (PacificVis), 2014 IEEE*. IEEE, 2014, pp. 57–64.

[110] P. Guo, H. Xiao, Z. Wang, and X. Yuan, "Interactive local clustering operations for high dimensional data in parallel coordinates," in *Pacific Visualization Symposium (PacificVis), 2010 IEEE*. IEEE, 2010, pp. 97–104.

[111] H. Zhou, X. Yuan, H. Qu, C. Weiwei, and B. Chen, "Visual clustering in parallel coordinates," in *Eurographics/IEEE-VGTC Symposium on Visualization*. IEEE, 2008.

[112] G. Andrienko and N. Andrienko, "Parallel coordinates for exploring properties of subsets," in *Coordinated and Multiple Views in Exploratory Visualization, 2004. Proceedings. Second International Conference on*. IEEE, 2004, pp. 93–104.

[113] J. Heinrich, Y. Luo, A. E. Kirkpatrick, H. Zhang, and D. Weiskopf, "Evaluation of a bundling technique for parallel coordinates," *arXiv preprint arXiv:1109.6073*, 2011.

[114] J. Heinrich and D. Weiskopf, "State of the art of parallel coordinates," in *Eurographics 2013-State of the Art Reports*. The Eurographics Association, 2012, pp. 95–116.

[115] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 741–748, 2006.

[116] D. Holten and J. J. Van Wijk, "Force-directed edge bundling for graph visualization," in *Computer Graphics Forum*, vol. 28, no. 3. Wiley Online Library, 2009, pp. 983–990.

[117] O. Ersoy, C. Hurter, F. V. Paulovich, G. Cantareiro, and A. Telea, "Skeleton-based edge bundling for graph visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2364–2373, 2011.

[118] A. Lambert, R. Bourqui, and D. Auber, "Winding roads: Routing edges into bundles," in *Computer Graphics Forum*, vol. 29, no. 3. Wiley Online Library, 2010, pp. 853–862.

[119] D. Selassie, B. Heller, and J. Heer, "Divided edge bundling for directional network data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2354–2363, 2011.

[120] A. Telea and O. Ersoy, "Image-based edge bundles: Simplified visualization of large graphs," in *Computer Graphics Forum*, vol. 29, no. 3. Wiley Online Library, 2010, pp. 843–852.

[121] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen, "Visual clustering in parallel coordinates," in *Computer Graphics Forum*, vol. 27, no. 3. Wiley Online Library, 2008, pp. 1047–1054.

[122] A. Dasgupta and R. Kosara, "Pargnostics: Screen-space metrics for parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1017–1026, Nov. 2010. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2010.184

[123] W. Huang, S.-H. Hong, and P. Eades, "Effects of crossing angles," in *Visualization Symposium, 2008. PacificVIS '08. IEEE Pacific*, March 2008, pp. 41–46.

[124] C. Ware, H. Purchase, L. Colpoys, and M. McGill, "Cognitive measurements of graph aesthetics," *Information Visualization*, vol. 1, no. 2, pp. 103–110, Jun. 2002. [Online]. Available: http://dx.doi.org/10.1057/palgrave.ivs.9500013

[125] M. Harrower and C. A. Brewer, "Colorbrewer. org: An online tool for selecting colour schemes for maps," *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.

[126] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz, "Uncovering clusters in crowded parallel coordinates visualizations," in *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on.* IEEE, 2004, pp. 81–88.

Appendix A

SOFTWARE PROTOTYPE

Working prototype of Storygraph is available as Free and Open Source Software at http://www.github.com/sayush/E2. I wrote this software using in C#. The UI is created using WPF framework and I used Direct2D (DirectX v9) for rendering the charts. The current state of the prototype supports CSV files and SQLite databases. A minor limitation at this moment is that, at the very least, there needs to be four columns labelled Lat, Lng, Color and Label. Other additional columns are optional. The prototype generates Storygraph, timeline and map from the same data. The Storygraph also supports edge bundling and uncertainty visualization. Figure A.1 shows a screenshot of the prototype.

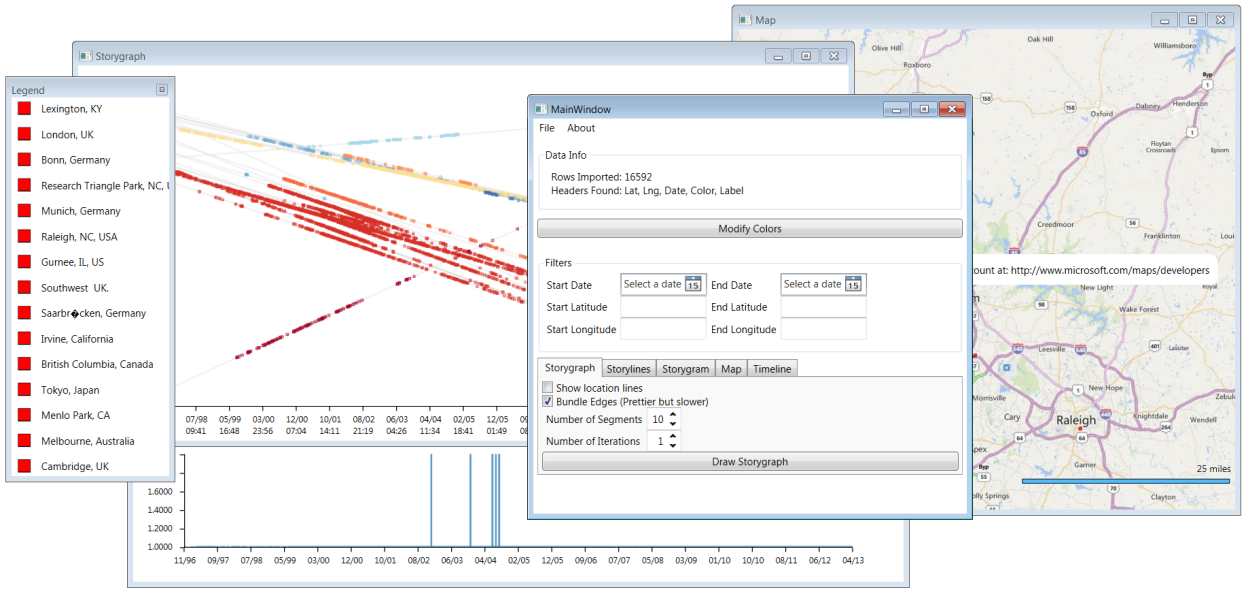


Figure (A.1) Screenshot of the software prototype.