An Interactive Web Based Spatio-Temporal Visualization System

Anil Ramakrishna, Yu-Han Chang, and Rajiv Maheswaran

Department of Computer Science,
University of Southern California,
Los Angeles, CA
{akramakr,maheswar}@usc.edu,ychang@isi.edu

Abstract. Exploratory data analysis can be used to uncover several hidden but interesting patterns that may not be obvious otherwise. Spatio-temporal data offer numerous challenges for such analyses and several approaches have been proposed to address them. In this paper, we present an interactive web based system that can visualize large spatio-temporal data set on a standard web browser. This approach is applied to visualize a large spatio-temporal data set with 2.5 million records and several key patterns are revealed in this process.

1 Introduction

The presence of compact global positioning systems (GPS) in modern smart phones and other hand-held devices has resulted in the collection of huge amounts of location aware data. In [1], the authors estimate that nearly 80 % of all digitally generated data contain a spatial element. Several data sources also include a time stamp to signify the occurrence of important events. Visualizing such **Spatio-Temporal data** could reveal several interesting patterns [2] that are not obvious otherwise or may require extensive analysis to be uncovered. However, owing to complications arising due to their sheer size, such data are not fully utilized in data analysis [3]. Geographic Information Systems (GIS) provide a few means to analyze such data but they are not fully optimized towards large scale visualization of temporal data.

In this paper, we present a *software mashup* [4] [5] based approach to visualize Spatio-Temporal data by combining modern web frameworks to create a system that provides interactive visualization using a web browser.

The remainder of the paper is organized as follows: In section 2, we discuss related work and publications. In section 3, the technologies used with the system are introduced and in Section 4, the system design is discussed. Section 5 offers concluding remarks on the utility of our approach.

2 Related Work

Spatio-Temporal visualization has been a subject of interest for researchers for several years. Andrienko et al. [2] discussed several spatio-temporal animation techniques such as map animation, filtering, visual querying, etc. giving real world examples that implement these. Guo et al. [3] presented a geo-visual analytic system that combined and leveraged well known visualization techniques in multivariate spatio-temporal data visualization. Wood et al. [5] explored the use of software mashups in data visualization and identified several key issues with this approach. Mashups are created by combining existing software technologies for rapid prototyping of various techniques. Slingsby et al. [4] also apply the mashup technique to combine tag clouds and tag maps, and evaluate them on Yahoo's tag map applet and Google Earth.

[6] discussed several java applet based online visualization tools along three themes of visualization: visualization of instant events, visualization of spatial movement and visualization of changing thematic data. Buja et al. [7] presented a taxonomy of visualization techniques in higher dimensions and implemented a few of them on their own home grown X-based visualization system called X-Gobi. Dykes et al. [8] discussed the applicability of a visualization tool called Location Trends Extractor(LTE) for spatio-temporal data. MacEachren et al. [9] combined technologies from Geo-Visualization and Knowledge Discovery methods and applied them on climate data visualization. Keim et al. [10] discuss the generic Visual Data Exploration paradigm: Overview first, zoom and filter, and then details-on-demand along with their own classification of visualization techniques.

3 Design

We present an online visualization system that uses modern web frameworks such as jQuery, D3, etc. to effectively visualize large scale data on a web browser. Though there are a few existing systems that leverage these frameworks for data visualization, they limit user interactions to a minimum and provide no means of navigation along the temporal attributes of the data set. One such example is the recently published article from Los Angeles times that reported the average time taken by the Los Angeles Fire Department (LAFD) units to arrive at locations of incidence. Associated with this report was a map based web page [11] that highlighted the average arrival time for various geographic regions within the Los Angeles city.

The work we present here extends the info-graphic from LA times by providing user interactions to control the visualization and to navigate along the temporal element of the data. We do this by providing the user a way to narrow down the visualization to various time resolutions such as weekday, month and year. The approach we present has many re-configurable elements that are generic enough to be used with any spatio-temporal data set.

3.1 Technologies

In this section we briefly describe the individual components of the system. Fig. 1 shows the data flow diagram among these components.

Geocoder Geocoding is the process of converting addresses that are in human readable format such as street names, zip codes, etc. into geographic coordinates such as latitude and longitude. Several geocoders exist that provide an API for online geocoding or provide rules for offline geocoding [12]. We use the open source Perl module Geo::Coder::US available from the Perl archive cpan.org [13]. This module uses the Tiger/Line data set [14] released by the US census bureau to generate rules for the geocoding process. The module also provides several functions that use these rules to parse the input address and return the corresponding geographic coordinates.

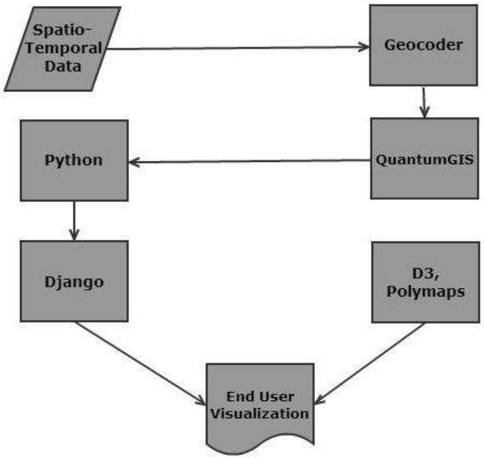


Fig. 1. Data flow diagram

QuantumGIS QuantumGIS [15] is an open source Geographic Information System (GIS) developed by the Open Source Geospatial Foundation (OSGeo). It provides capabilities for geospatial data analysis, editing and visualization, among other functionalities. We use QGIS to *bin* the data into geographic hexagonal cells as shown in Fig. 2. Binning is the process of grouping geospatial data into adjacent cells. QGIS supports binning through the MMQGIS plugin [16].

Python Python [17] is a general purpose programming language that supports functional, object-oriented and imperative programming styles. We use python to parse the data files and generate various statistics related to the data at various time resolutions.

Django Django [18] is an open source web framework written in Python. It emphasizes the Model-View-Controller (MVC) software architectural pattern

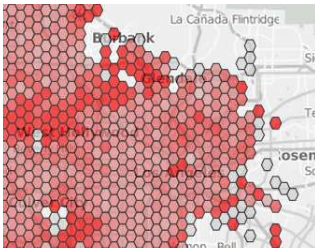


Fig. 2. Hexagonal grids overlayed on the map of Los Angeles city

and promotes rapid development and code re-usability. We use Django to develop the front end web application.

D3 D3 [19] is a JavaScript library that supports a data-driven approach to manipulating elements of the Document Object Model (DOM). It provides efficient methods for data-driven manipulation of the DOM with an emphasis on Web Standards. We use D3 to draw the geographic grid to manipulate the individual cells in the visualization.

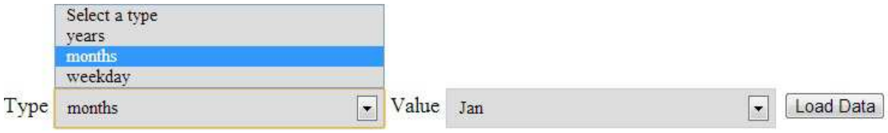
3.2 Data

The data we have used comes from the call history of LAFD over five years. This data set contains approximately 2.5 million records, each including the incident type, location and several time stamps such as the time of incidence, arrival time of dispatched units, etc. that correspond to the occurrence of various events.

The data records are geocoded in Perl using the Geo::Coder::US module. These records are then binned by QGIS into adjacent geographic cells which are saved separately. Various statistics are collected from the data records in each cell using Python. The process is repeated over various *time resolutions*, i.e. all data entries that belong to one particular weekday, month or year are grouped and the relevant statistics are collected. If the data contains multiple types, multiple sets of statistics are created, one for each type of the data, and stored. This is the only data specific component of the whole design and by suitably modifying these scripts, the approach can be applied to visualize any spatio-temporal data set. The statistics that are collected over all the time resolutions and cells are saved independently and made available to a web server in JSON format.

3.3 The Front end

The end user works with a web page consisting of a map focused on the locality of the data. The map is rendered using the polymaps framework [20] and the tiles are fetched from cloudmade servers [21]. The grids are visualized using the framework D3. The data driven design of D3 was especially accommodating for



Type months

Value Jan

Fig. 3. Menu to control the time resolution

our needs. Each cell from the hexagonal grid structure was colored independently according to some relevant statistic of that cell using D3.

When loaded, the web page presents the user with a default grid layout overlayed on the map along with two menu items. In the first menu, the user selects a time resolution from one of weekday, month or year. Based on the user's selection, the second menu is populated with all the available time frames of that resolution as shown in Fig. 3. For example, if the user selects year, the second menu is populated with all the years that appear in the data set. Based on the selection from the second menu, a JSON file corresponding to that time frame is loaded from the server. These two menus provide the user a way to narrow down the visualization to one particular time frame and resolution. If the data contains multiple types, all the available types are listed as radio buttons. Based on the current selection of the radio button, the corresponding data set is visualized.

3.4 User Interactions

A slider is provided as shown in Fig. 4 through Fig. 6 to add interaction to the visualization. The slider was created using jQuery's slider plugin and has thirty divisions with each division representing time in minutes. By moving the slider, the color of each cell of the grid layout is changed based on the current value of the slider and some pre-calculated statistic from the data set corresponding to that geographic cell. With the LAFD dataset, we computed the percentage of records that take response time greater than or equal to the current value of the slider. For a given value of the slider, the percentage of records that had response time greater than the value of the slider is converted into a color using d3's scale function and visualized on the cells. Higher the number of records with time more than the slider's value, more darker the cells are, as shown in Fig. 4.

3.5 Discussion

Our observations have revealed few key patterns and confirmed a few previously established ones associated with LAFD's operations. For example, as shown in Fig. 4, the outer grids are generally darker than the inner grids. Since the intensity of color signifies the no of incidents with high response time, we can draw the intuitive conclusion that LAFD takes significantly higher times to respond to localities that lie in the boundary of the city as opposed to ones that are in the center. It is also clear that regions such as Bel Air, Los Angeles that have mountains and other difficult to navigate regions have higher average response time when compared to other parts of the city. This is evident from the left figure in Fig. 4 in which Bel Air is located in the center dark cluster.

We also noticed that the average response time is much higher on a Sunday compared to the rest of the week as shown in Fig. 5. Non-emergency incidents showed relatively high average response times compared to emergency incidents as shown in Fig. 6. However they did not appear to show the same pattern of delayed response time in the boundary regions as the emergency incidents. We did not notice any significant difference across the 12 months but we did notice a gradual increase in the number of regions with higher than average response times each year.

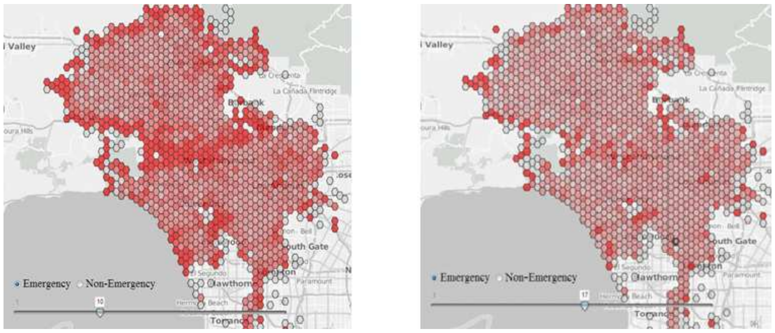


Fig. 4. Visualization at slider value=10(left) and slider value=17

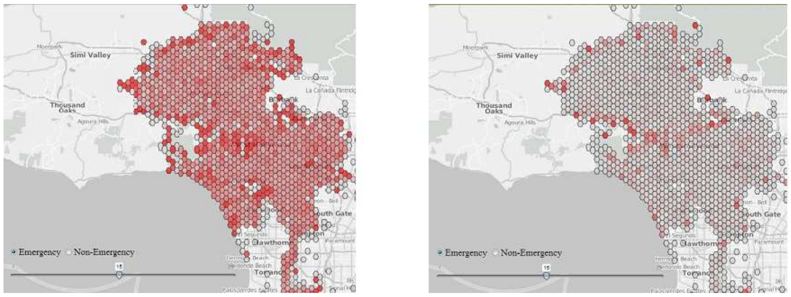


Fig. 5. Visualization for Sunday(left) and Thursday

4 Conclusion

We presented a Spatio-Temporal Visualization system that combines open source technologies with modern web frameworks to create powerful web based visual-

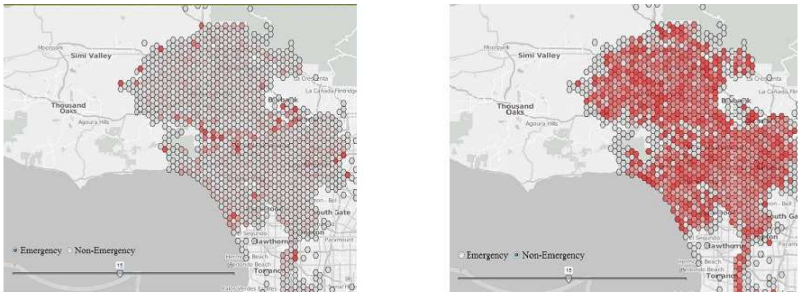


Fig. 6. Visualization for Emergency(left) and Non-Emergency incidents

izations. This system was applied to visualize the call history of LAFD and a few previously observed patterns in the data were identified, successfully demonstrating the effectiveness of the system. In addition, several interesting inferences were made about the operations of LAFD.

Future work includes providing a more continuous navigation across the time attributes using a slider and providing additional interactions to control other attributes of the data.

Bibliography

[1] Alan M. MacEachren and Menno-Jan Kraak. Research challenges in geovisualization. *Cartography and Geographic Information Science*, 28(1), 2001.

[2] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503 – 541, 2003.

[3] Diansheng Guo, Jin Chen, Alan M. MacEachren, and Ke Liao. A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1461–1474, November 2006.

[4] Aidan Slingsby, Jason Dykes, Jo Wood, and Keith Clarke. Interactive tag maps and tag clouds for the multiscale exploration of large spatio-temporal datasets. In *Proceedings of the 11th International Conference Information Visualization*, IV '07, pages 497–504, Washington, DC, USA, 2007. IEEE Computer Society.

[5] J. Wood, J. Dykes, A. Slingsby, and K. Clarke. Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1176–1183, 2007.

[6] N. Andrienko, G. Andrienko, and P. Gatalsky. Visualization of spatio-temporal information in the internet. In *Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on*, pages 577–585, 2000.

[7] Andreas Buja, Dianne Cook, and Deborah F. Swayne. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5:78–99, 1996.

[8] J.A Dykes and D.M Mountain. Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications. *Computational Statistics & Data Analysis*, 43(4):581 – 603, 2003.

[9] A.M. MacEachren, M. Gahegan, W. Pike, I. Brewer, G. Cai, E. Lengerich, and F. Hardistry. Geovisualization for knowledge construction and decision support. *Computer Graphics and Applications, IEEE*, 24(1):13–17, 2004.

[10] Daniel A. Keim, Christian Panse, and Mike Sips. Information visualization: Scope, techniques and opportunities for geovisualization. *Exploring Visualization*, 2005.

[11] How fast is lafd where you live. *http://graphics.latimes.com/how-fast-is-lafd*, 2012.

[12] List of available geocoding software. *http://geoservices.tamu.edu/Services/Geocode/About/GeocoderList.aspx*.

[13] The cpan geocoder module. *http://search.cpan.org/ sderle/Geo-Coder-US/US.pm*.

[14] United states census bureau. tiger/line data. *http://www.census.gov/geo/maps-data/data/tiger.html*.

[15] Quantumgis. *http://qgis.org.*
[16] Mmqgis. *http://michaelminn.com/linux/mmqgis.*
[17] Python. *http://www.python.org.*
[18] The django project. *https://www.djangoproject.com.*
[19] Data-driven documents. *http://d3js.org.*
[20] Polymaps. *http://polymaps.org.*
[21] Couldmade. *http://cloudmade.com.*