Visualizing large spatial time series data on mobile devices — Combining the HeatTile system with a progressive loading approach

Sebastian Meier*

* Potsdam University for Applied Sciences, Germany
meier@fh-potsdam.de

Abstract. In this paper we introduce an approach for visualizing large spatial time series data sets on mobile devices. We are employing the HeatTile system (Meier, Heidmann, & Thom, 2014) and extending it with a progressive loading approach, the *stream approach*. By combining those two approaches we are trying to overcome the performance and bandwidth limitations inherent to mobile devices. We will focus on the technological advantages of the presented approach in a performance comparison with other common approaches. Furthermore, we will introduce an animated time series visualization as well as an interface designed specifically for the presented method in order to emphasize the advantages of the approach. To further highlight the possible applications of the method we will present two real-world use cases.

Keywords: Geovisualization, Web, Time Series

1. Introduction

Maps and visualizations of spatial data have become an important component in mobile applications. From navigation to location based social networks to location-based recommender systems, maps and the display of georeferenced data have become a common UI pattern. Nonetheless, the variety of visualizations of spatial data found in common mobile applications is still limited.

One of the reasons for that are the limitations inherent to mobile devices. Even though the latest devices are equipped with powerful processors and the telecommunication companies keep on enhancing bandwidth performance, mobile devices are still inferior to desktop computers. Most modern visualizations are still optimized for those stationary machines. The growing mobile sector and its ubiquitousness in our every day life asks for innovation in the development of mobile centered maps and geovisualizations. Thus, we need to overcome the technological obstacles and improve our methods to take mobile devices into account.

In this paper, we are looking into the visualization of large spatial time series data sets, a type of data set that has received a lot of notion in the field of conventional GIS. A well-known example is the space-time cube, which was introduced by Hägerstrand in the late 1960s (Hägerstrand, 1970; Lenntorp, 1999) and more recently reviewed by Kraak (Kraak, 2003; 2008; Kraak & Koussoulakou, 2005). Current work in the area has been conducted by MacEachran (MacEachren, Wachowicz, Edsall, Haug, & Masters, 1998), Andrienko (N. Andrienko & Andrienko, 2007; N. Andrienko, Andrienko, & Gatalsky, 2003; Gatalsky, Andrienko, & Andrienko, 2004), Dykes (Dykes & Mountain, 2003) and Wood (Wood, Dykes, Slingsby, & Clarke, 2007). Even though some of the examples do use data *gathered* from mobile phones, none of the examples elaborates on visualizing those data sets on mobile devices.

2. Problem definition

Mobile devices confront developers and designers with multiple limitations in regards to implementing geovisualizations. Regarding technological aspects one can observe that, for one thing, *bandwidth limitations* are restricting the size of transmitted data. For the other thing, there are phone specific processing limitations in the form of *processor speed* and *memory*. Concerning the design of the visualization as well as the interactions and the overall experience we have to focus on *multitouch technology* as the primary input method. When implementing geovisualizations on mobile devices, we additionally need to deal with *smaller screen size*.

In the context of large spatial time series data sets the question arises how we can present such a bandwidth intensive data set on mobile devices while taking the limitations outlined above into account.

3. Related Works

Providing progressive visualizations of data-heavy objects on limited bandwidth systems is already in use in many areas. One of the most prominent examples for dealing with limited bandwidth systems can be found in more complex image formats on the web. The GIF (CompuServe Incorporated, 1990), JPEG (CCITT Study Group VIII & The Joint Photographic Experts

Group JPEG, 1992) and PNG (Boutell & Costello, 2003) formats offer so called *progressive* or *interlaced* saving modes (Wikipedia, 2014a) which means that the saving process produces multiple levels of detail (LOD), allowing the user to see a low-resolution version of the image while the high resolution image is still loading. Similar approaches are used in modern online video platforms. Here, the *adaptive bitrate streaming* method offers a video in multiple resolutions, depending on the available bandwidth and buffer (Wikipedia, 2014b).

This progressive approach to displaying bandwidth-heavy objects was recently applied to interactive web based data visualization by Glueck et al (Glueck, Khan, & Wigdor, 2014). In their recent project "Dive In", they present a similar *progressive loading* approach for graph visualizations. This method deployed by Glueck et al. allows the user to quickly browse through a large data set and receive an instant overview visualization that becomes more detailed as the client receives more detailed data from the server.

In regards to mobile maps Sester and Brenner used the same approach to apply a progressive loading tool in the form of "continuous generalization" onto maps (Sester & Brenner, 2005). Even though their primary motivation was not to enhance the loading of maps but the handling of the zoom levels and by that creating more generalized visualizations for lower zoom levels and more detailed visualizations for higher zoom levels.

The major difference between Glueck's and Sester's coarse overview methods in comparison to the *stream approach* discussed in this paper is that the latter does not only focus on the level of detail (LOD). Instead, its focus lies on the virtue of the time-dimension, which is progressively loaded, similar to videos provided through online streaming-services (see above), which makes it possible to progressively load timeseries-visualizations.

4. Visualization

In modern GIS a variety of visualization types are used to display multivariate spatial time series data, e.g. the aforementioned space-time cube. This paper, however, focuses on animated heat maps.

"Heat maps enable viewers to quickly identify high density areas without losing the general spatial context. This is typically useful when analysing multivariate geospatial data, where users need to get a sense of the correlation between geographical features and another measurement." (Pettit, Widjaja, Sinnott, & Stimson, 2012)

The feature of keeping the geospatial context becomes even more prominent when combined with slippy maps that allow users to zoom and pan in order to identify geographic regions and explore their spatial context. The use of heat maps has already been discussed in many contexts, e.g. "Volunteered Geographic Information Systems" (Trame & Keßler, 2011), "Road Incident Data" (Dillingham, Mills, & Dykes, 2011) or "Social Context" (Komninos, Besharat, Ferreira, & Garofalakis, 2013), just to name a few. The visualization (See Figure 2) itself will not be further discussed in this paper, as the focus lies on the technical aspects of the *stream approach* employed to load and generate the heat maps.

5. Technical Solution

On the server side of the *stream approach* the clustering approach of the HeatTile system (Meier et al., 2014) is used, which allows for creating GeoJSON (Butler et al., 2008) files containing the heat map data. While the existing HeatTile system is generating tiled heat maps for a specific data set, the time series, data dealt with in this context, requires multiple heat maps, one for each point in time. Consequently, a series of GeoJSONs need to be loaded in the process, depending on the length of the time series. Requesting a lot of files from a server is common bad practice in web development because a *request time* is added to the actual *response time* (Yahoo, 2014). In order to overcome this problem the recently introduced *EventSource interface* (Hickson, 2012) can be used. This method requests a file from a server and processes it as soon as the data is being pushed to the client, instead of waiting for the download process to be completed. This allows sending multiple GeoJSONs concatenated in one large file to the client and, instead of waiting until the whole file has completed loading, start processing the individual GeoJSONs, which have been concatenated, as soon as the server has finished sending the individual segment.

5.1. Reliability

When it comes to mobile networks, a problem that needs to be dealt with is the loss of connection. Especially if a user is trying to load large files, or a website is using a series of asynchronous loading requests, a loss of connection can lead to missing data or, even worse, a malfunctioning website. The *EventSource interface* has a built-in functionality to overcome this problem. More precisely, the interface has a "reconnect function", which means that the data coming from the server is only processed if it is completely transmitted. As soon as the connection is lost, the API tries to reconnect. The system even offers developers the opportunity to define a reconnect timeout. When a reconnection is established, the system proceeds with the last set of data that was sent before the loss of connection until the whole data set is received.

6. Interface Solution

The idea to use the *EventSource interface* is not only inspired by an increased loading performance. The waiting time for websites and its effect on the perceived quality of the user experience is a thoroughly discussed topic in the web HCI community (Antonides, Verhoef, & van Aalst, 2001; Egger, Hossfeld, Schatz, & Fiedler, 2012; Nah, 2004; Roto & Oulasvirta, 2005; Sears, Jacko, & Borella, 1997). Keeping latency short or, rather, file sizes small is therefore an important part of web optimization. As the waiting time cannot be completely avoided in a loading process, the stream approach is trying to improve the overall user experience of the time series visualization by making use of the *EventSource interface*'s ability to deliver data progressively.

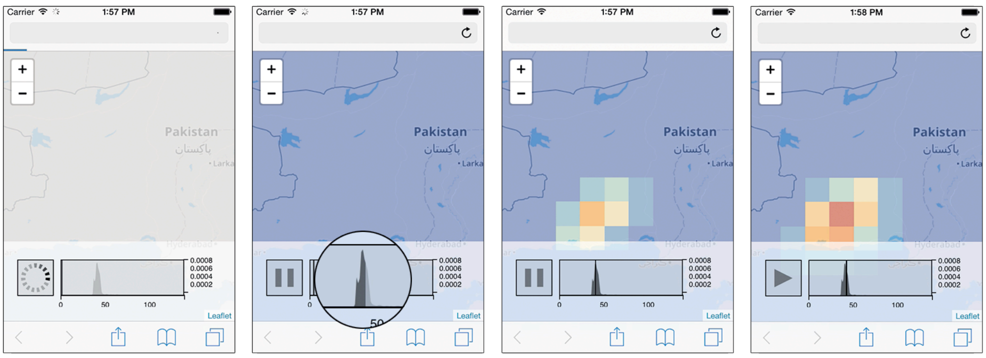

Figure 1. Implementation of the heat map animation with the overview progress bar interface at the bottom. From left to right: 1.1 Initial Loading View with overview visualization 1.2. Overview visualization is used as a progress bar 1.3. The playhead shows the current position on the timeline 1.4. The animation is paused

By utilizing the new functionality we implement two improvements. First of all, we have extended the HeatTile system in order to produce an overview data set, holding the maximum value per point in time. Depending on the visualization, this could of course be modified to include minimum, average or other values. This overview visualization is sent through the *EventSource interface* as the first data set. This allows us to quickly provide the user with a first impression of the time series data set and provide her with a first glance at the data until the whole set is loaded (Fig. 1.1).

In order to inform the user about the loading progress, we use the overview visualization to display the data that has already been loaded and include the remaining data that is still to load in form of a progress bar (Fig. 1.2).

Hence, the user has a better feeling of what is happening in the system while also reducing the perceived waiting time (Myers, 1985).

Furthermore, we calculate the remaining time that is needed for loading the complete data set. If no losses of connection occur, we are able to start the playback of the time series even if the data is incomplete because we can calculate the time needed for playing the whole set and the time remaining for loading the remaining data.

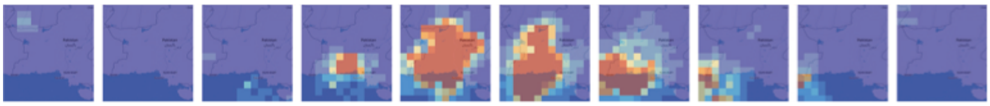

Figure 2. 10 sample frames from the earthquake animation.

7. Performance

In this paper we have presented the *stream approach* as an extension to the *HeatTile* system to optimize the visualization of multiple GeoJSON layers in form of an animation for mobile devices. In order to highlight the performance increase, the *stream approach* was compared to a *consecutive* and a *asynchronous approach* (See Figure 3). The *consecutive* loading approach requests one GeoJSON time frame, waits for the request to finish loading and afterwards loads the next time frame until all frames are completely loaded. This approach is good for estimating the complete loading time and handling errors, e.g. interruptions on mobile networks. The *asynchronous* loading approach tries to load as many time frames at the same time as possible. Therefore, all file requests are sent almost immediately in parallel to the server after the page starts to load. Since most servers can handle multiple file requests from each client at the same time, this method is a good alternative, especially on fast connections. On the other hand, handling loading errors and estimating the remaining time until the playback is ready to start is harder to calculate, as the files are not necessarily being loaded in a consecutive order.

The three methods were compared with the same data set sample (from Use Case 2) with three broadband conditions: 3G (750Kbps), DSL (2Mbps) and WiFi (30Mbps). Each method was tested 20 times under each condition, resulting in 180 speed test results. As we can see in Figure 4, the *stream approach is* faster than the other approaches, especially on the slow network.

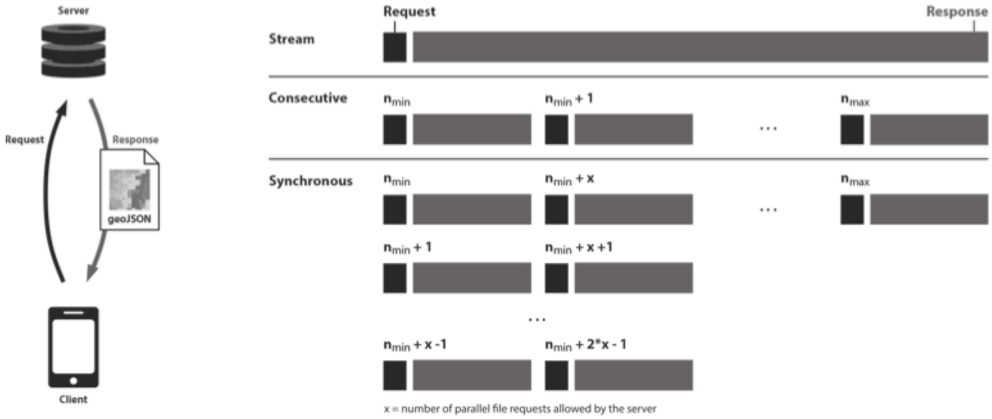

Figure 3. Comparison of file loading approaches.

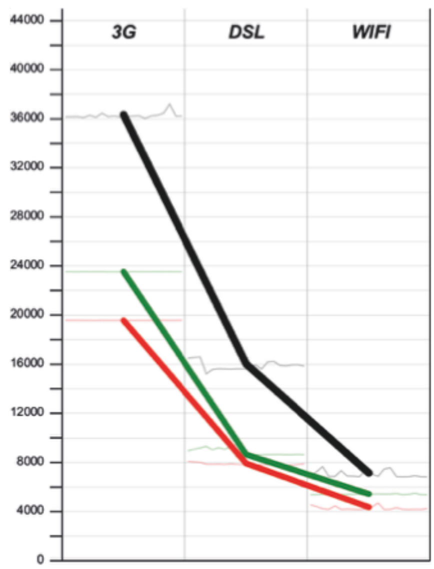

Figure 4. Performance comparison related to network speed: *red* stream approach, *green* asynchronous approach, *black* consecutive approach. Time on the y-axis in milliseconds.

8. Use Cases

Maps and spatial data are an embedded component of many mobile applications. From weather applications to recommender systems to location based services. In terms of geovisualizations and interfaces, most applications focus on simply visualizing single locations on maps or, in regards to navigation applications, on displaying lines between points. More complex visualizations like *heat maps* (Lemmela, Korhonen, Vetek, & Corporation, 2008) are still rare. The *HeatTile* system provides the possibility to effortlessly integrate heat maps into mobile applications (Meier et al., 2014). The *stream approach* built onto the *HeatTile* system as discussed above, extends the range of use cases and makes it possible to include time series visualizations on mobile devices.

8.1. People's Movement in urban environments

The location based recommender service "Foursquare" (Foursquare, 2014a) offers their users interfaces to explore nearby locations, search for specific locations e.g. restaurants or shops and extends this functionality by adding gamification and social interactions to the experience. One of those add-on features is the so called "check in" function, also featured in similar apps like "Yelp" (Yelp, 2014). It allows the user to mark a location as "visited" on a specific timestamp. Thereby, the user can create e.g. a diary-like log of her activities. The data about "check ins" is further used for providing businesses insights into their target audiences and generate crowdsourced content. This opens the opportunity, if applied on a broader scale, to visualize and analyse activity patterns in movement and behaviour in urban areas (Foursquare, 2014b; Xia et al., 2014).

Put into context, this could allow users to see in an animated time series which parts of the city are especially lively neighbourhoods during which specific hours of the day and help them make decisions on the go, e.g. where they want to go out. To demonstrate this, we collected data (24h * 7 days) of one district (Kreuzberg/Neukölln) in the city of Berlin (Figure 5). The data was then visualized in a time-lapse animation. Similar to Foursquare's "Pulse" (Foursquare, 2014b), this kind of visualization especially highlights the differences between daytime and after-work hour "check ins".

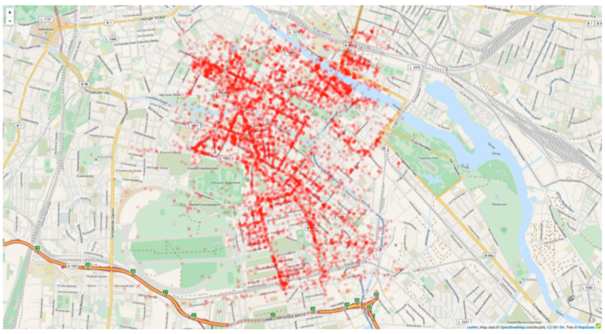

Figure 5. 7209 observed venues in Kreuzberg / Neukölln.

If this visualization is implemented with common web-based visualization tools, it would either require pre-rendered images or a video. This is the case in e.g. Foursquare's "Pulse" visualization (Foursquare, 2014b). Other common web-based visualization techniques require sending the full data set, which is then rendered on client-side. The former lacks interactivity and creates bandwidth heavy media objects. The latter is limited by the mobile device's processing and memory resources. Using the HeatTile approach we are able to keep the transmitted data small while still being able to visualize a big data set and integrate interactive features.

8.2. Earthquake Data

Beyond social interactions and urban data, the new method of the *stream approach* can be applied in early warning systems and emergency response systems. The department for earth sciences at the University of Potsdam, Germany, has been conducting a research project on remote sensing of natural disasters like e.g. earthquakes (Krüger, Ohrnberger, Ehlert, & Rößler, 2004). The institute/department is supervising a server cluster that collects data from earthquake sensor stations around the world. Once a certain seismic activity-level is exceeded, the data is recorded and processed. Metadata and processing results are then provided to the public, decision makers and e.g. health organizations via a web portal. The data is essential, especially for emergency workers in the field, e.g. working with the Red Cross, for being able to target the most severely afflicted areas after an earthquake and also for being able to identify areas that could be hit by possible aftershocks. Even though most people perceive an earthquake as a local stationary event, earthquakes are dynamic, moving events, which means that also the epicentre of an earthquake is moving. Apart from the scientific static visualizations by Krüger et al. (Krüger, Ohrnberger, Ehlert, & Rößler, 2014), animated heat maps are one of many other ways to visualize a seismic event (Figure 6). In a first prototype, four earthquake data sets were transferred into the mobile first infrastructure. Using the *streaming approach* a simple web based application was created, allowing users to explore the animated earthquake data sets (Figure 1). In addition to the advantages highlighted in the first use case, this second scenario shows how to enable the user to get an overview first and then zoom into a certain region. Thereby we are able to provide a more detailed time-series visualization of the selected region. By implementing the visualization on top of a slippy map and by limiting the requested data to the extent that can be displayed on the screen, the user can easily zoom and pan. After changing the zoom level and thereby the extent, the loading process is reinitiated with the new parameters (extent and zoom). Providing as many levels of detail for a dataset like this with common visualizations techniques would require the pre-rendering of a large amount of images. With the approach presented in this paper the system can gather the required data in real time and reduce the transmitted data to a minimum.

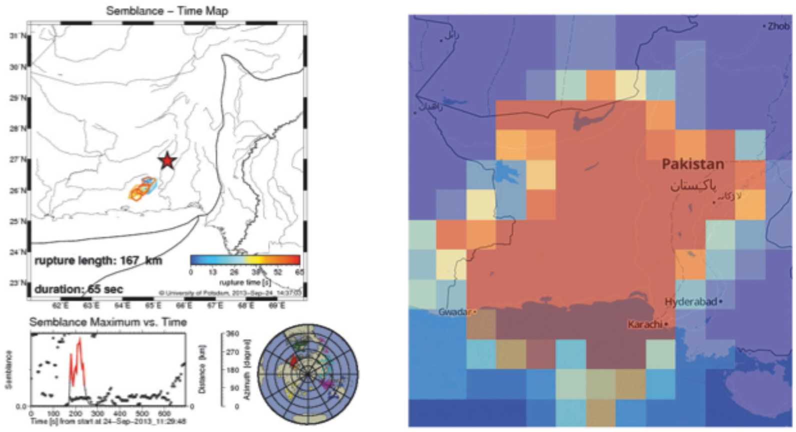

Figure 6: Left: Scientific Figure by Krüger et al. (Krüger et al., 2004). Right: Rasterized heat map visualization oft he same time series data set.

9. Conclusion

Mobile web enabled devices are an important cornerstone of our digital infrastructure, their ability to georeference themselves opens up new possibilities for mapping- and geovisualization-applications. We believe that more mobile centered visualization research is required to make use of this advantage and create more engaging visualizations and applications.

In the approach outlined in this paper we used the *HeatTile* system for clustering and compressing a large spatial time series data set to optimize it for being received by mobile devices. Furthermore, we proved that the *stream approach* is a good method for transmitting a large data set in multiple subsets. By this, request delays can be reduced and hence allow for a progressive visualization of the data and thereby cut down the waiting time until the first time frame of the time series data is visualized. Moreover, the latency until the first interaction with the visualization is possible to be minimized. Furthermore, it could be shown that using this approach it is possible to include contextual progress indicators. Summarising, we believe that the *stream approach* could be used in various mobile contexts and that further research on how to implement the *stream approach* into mobile data visualizations can lead to overcoming some of the mobile device's limitations that are described in this paper.

10. Future Research

Looking into future research possibilities, the method described in this paper could be useful in areas beyond the visualization of time series data. Having proven that loading binned data on mobile networks through the *EventSource interface* achieves better results in terms of speed and performance would imply that for example loading large data sets of non-binned individual locations could profit as well. By this, users would be enabled to perceive a progressive experience and to see the first locations appear while the rest of the set is still loading.

We are conducting further research that is looking into methods that could be used to modify the data sent by the server through the *EventSource interface* by pushing requests to the server on another channel and thereby modifying the output stream. Additionally, we are trying to implement the progressive loading approach in a way that not only allows the progressive loading on the time axis but also on the level of detail axis (LOD), similar to Glueck et al.'s approach (Glueck et al., 2014). This would allow the user to load a low-resolution heat map animation that increases its level of detail once the server sends more detailed data (Figure 7). This could decrease the waiting delay until the user can interact with data even further and allow the user to skip through the timeline and preview frames in low quality similar to “Video Scrubbing" (Matejka, Grossman, & Fitzmaurice, 2012).

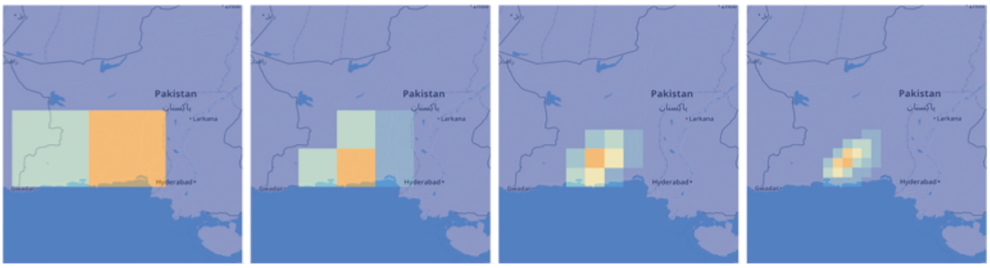

Figure 7: A range of LODs generated by changing the cluster grid size.

11. Acknowledgments

The modified *HeatTile* system, prototypes and code examples are available on GitHub under MIT’s / GPL’s open source software license: https://github.com/sebastian-meier/HeatTiles

References

Andrienko, N., & Andrienko, G. (2007). Designing Visual Analytics Methods for Massive Collections of Movement Data. *Cartographica: the International Journal for Geographic Information and Geovisualization*, *42*(2), 117–138. doi:10.3138/carto.42.2.117

Andrienko, N., Andrienko, G., & Gatalsky, P. (2003). Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, *14*(6), 503–541. doi:10.1016/S1045-926X(03)00046-6

Antonides, G., Verhoef, P. C., & van Aalst, M. (2001). Consumer Perception and Evaluation of Waiting Time: A Field Experiment. *Journal of Consumer Psychology*, *12*(3), 193–202. doi:10.1207/S15327663JCP1203_02

Boutell, T., & Costello, A. M. (2003). Portable Network Graphics (PNG) Specification (Second Edition). *W3.org*. Retrieved October 28, 2014, from http://www.w3.org/TR/PNG/

Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., & Schmidt, C. (2008). GeoJSON Specification. *Geojson.org*. Retrieved November 1, 2014, from http://geojson.org/geojson-spec.html

CCITT Study Group VIII, & The Joint Photographic Experts Group JPEG. (1992). ISO/IEC 10918-1 : 1993(E). *W3.org*. Retrieved October 29, 2014, from http://www.w3.org/Graphics/JPEG/itu-t81.pdf

CompuServe Incorporated. (1990). GIF89a Specification. *W3.org*. Retrieved October 28, 2014, from http://www.w3.org/Graphics/GIF/spec-gif89a.txt

Dillingham, I., Mills, B., & Dykes, J. (2011). Exploring Road Incident Data with Heat Maps. Poster presented at the Geographic Information Science Research UK 19th Annual Conference (GISRUK 2011), 27 - 29 Apr 2011, University of Portsmouth, Portsmouth, UK

Dykes, J. A., & Mountain, D. M. (2003). Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications. *Computational Statistics & Data Analysis*, *43*(4), 581–603. doi:10.1016/S0167-9473(02)00294-3

Egger, S., Hossfeld, T., Schatz, R., & Fiedler, M. (2012). Waiting times in quality of experience for web based services (pp. 86–96). Presented at the 2012 Fourth International Workshop on Quality of Multimedia Experience (QoMEX 2012), IEEE. doi:10.1109/QoMEX.2012.6263888

Foursquare. (2014a). Foursquare for Developers. *Developer.Foursquare.com*. Retrieved November 4, 2014, from https://developer.foursquare.com/

Foursquare. (2014b). Foursquare-Check-ins zeigen den Puls von Städten. Retrieved November 1, 2014, from https://foursquare.com/infographics/pulse

Gatalsky, P., Andrienko, N., & Andrienko, G. (2004). Interactive analysis of event data using space-time cube. *2000 IEEE Conference on Information Visualization. an International Conference on Computer Visualization and Graphics*, 145–152. doi:10.1109/IV.2004.1320137

Glueck, M., Khan, A., & Wigdor, D. J. (2014). Dive in! (pp. 561–570). Presented at the the 32nd annual ACM conference, New York, New York, USA: ACM Press. doi:10.1145/2556288.2557195

Hägerstrand, T. (1970). What about People in Regional Science? *Regional Sciences*, *24*(1), 7–24.

Hickson, I. (2012, December 11). Server-Sent Events. *W3.org*. Retrieved November

1, 2014, from http://www.w3.org/TR/eventsource/#the-eventsource-interface
Komninos, A., Besharat, J., Ferreira, D., & Garofalakis, J. (2013). HotCity (pp. 1–10). Presented at the the 12th International Conference, New York, New York, USA: ACM Press. doi:10.1145/2541831.2543694
Kraak, M.-J. (2003). The space-time cube revisited from a geovisualization perspective (Vol. 1995, pp. 1988–1996). Presented at the International Cartographic Conference 1995.
Kraak, M.-J. (2008). Geovisualization and Time– New Opportunities for the Space–Time Cube. In M. Dodge, M. McDerby, & M. Turner, *Geographic Visualization* (pp. 293–306). Chichester, UK: John Wiley & Sons, Ltd. doi:10.1002/9780470987643.ch15
Kraak, M.-J., & Koussoulakou, A. (2005). A Visualization Environment for the Space-Time-Cube. In *Developments in Spatial Data Handling* (pp. 189–200). Berlin/Heidelberg: Springer-Verlag. doi:10.1007/3-540-26772-7_15
Krüger, D. F., Ohrnberger, D. M., Ehlert, D. I. L., & Rößler, D. D. (2004). Tsunami - GITEWS: Teleseismic tracking of the rupture propagation of earthquakes. *Geo.Uni-Potsdam.De*. Retrieved November 1, 2014, from http://www.geo.uni-potsdam.de/tl_files/arbeitsgruppen/ag_array_seismologie/ruptrack/
Krüger, D. F., Ohrnberger, D. M., Ehlert, D. I. L., & Rößler, D. D. (2014, September 24). Pakistan — Teleseismic rupture tracking of large earthquakes. *Geo.Uni-Potsdam.De*. Retrieved November 1, 2014, from http://www.geo.uni-pots-dam.de/tl_files/arbeitsgruppen/ag_array_seismologie/ruptrack/pages/event_detail.php?id=up2013ssol
Lemmela, S., Korhonen, H. J., Vetek, A., & Corporation, N. (2008). Identifying interesting locations based on commonalities in location based postings. Google Patents
Lenntorp, B. (1999). Time-geography - at the end of its beginning. *GeoJournal*, *48*(3), 155–158. doi:10.2307/41147366?ref=no-x-route:a11722e899c778802cb417bbaa664dc1
MacEachren, A. M., Wachowicz, M., Edsall, R., Haug, D., & Masters, R. (1998). Constructing knowledge from multivariate spatiotemporal data: integrating geographical visualization with knowledge discovery in database methods. *International Journal of Geographical Information Science*, *13*(4), 311–334. doi:10.1080/136588199241229
Matejka, J., Grossman, T., & Fitzmaurice, G. (2012). Swift: reducing the effects of latency in online video scrubbing (pp. 637–646). Presented at the CHI '12: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, New York, USA: ACM Request Permissions. doi:10.1145/2207676.2207766
Meier, S., Heidmann, F., & Thom, A. (2014). Heattile, a New Method for Heatmap Implementations for Mobile Web-Based Cartographic Applications. In *Thematic Cartography for the Society* (pp. 33–44). Cham: Springer International Publishing. doi:10.1007/978-3-319-08180-9_4
Myers, B. A. (1985). The importance of percent-done progress indicators for computer-human interfaces (Vol. 16, pp. 11–17). Presented at the Computer Human Interactions, ACM. doi:10.1145/1165385.317459
Nah, F. F.-H. (2004). A study on tolerable waiting time: how long are Web users willing to wait? *Behaviour & Information Technology*, *23*(3), 153–163.

doi:10.1080/01449290410001669914
Pettit, C., Widjaja, I., Sinnott, R., & Stimson, R. J. (2012). Visualisation support for exploring urban space and place. *XXII ISPRS Congress*.
Roto, V., & Oulasvirta, A. (2005). Need for non-visual feedback with long response times in mobile HCI. WWW '05 Special interest tracks and posters of the 14th international conference on World Wide Web, 775–781. doi:10.1145/1062745.1062747
Sears, A., Jacko, J. A., & Borella, M. S. (1997). Internet Delay Effects: How Users Perceive Quality, Organization, and Ease of Use of Information. *CHI Extended Abstracts*, 353–354. doi:10.1145/1120212.1120430
Sester, M., & Brenner, C. (2005). Continuous Generalization for Visualization on Small Mobile Devices. In *Developments in Spatial Data Handling* (pp. 355–368).
Trame, J., & Keßler, C. (2011). Exploring the lineage of volunteered geographic information with heat maps. *GeoViz Conference, Hamburg*.
Wikipedia. (2014a, January 7). Interlacing (bitmaps) - Wikipedia, the free encyclopedia. *En.Wikipedia.org*. Retrieved October 28, 2014, from http://en.wikipedia.org/wiki/Interlacing_(bitmaps)
Wikipedia. (2014b, June 4). Adaptive bitrate streaming - Wikipedia, the free encyclopedia. *En.Wikipedia.org*. Retrieved October 28, 2014, from http://en.wikipedia.org/wiki/Adaptive_bitrate_streaming
Wood, J., Dykes, J., Slingsby, A., & Clarke, K. (2007). Interactive Visual Exploration of a Large Spatio-temporal Dataset: Reflections on a Geovisualization Mashup. *IEEE Transactions on Visualization and Computer Graphics*, *13*(6), 1176–1183. doi:10.1109/TVCG.2007.70570
Xia, C., Schwartz, R., Xie, K., Krebs, A., Langdon, A., Ting, J., & Naaman, M. (2014). CityBeat: Real-time Social Media Visualization of Hyper-local City Data. . In Companion Publication of the 23rd International Conf. on World Wide Web (WWW), 2014
Yahoo. (2014). Best Practices for Speeding Up Your Web Site. *Developer.Yahoo.com*. Retrieved November 1, 2014, from https://developer.yahoo.com/performance/rules.html
Yelp. (2014, November 1). What are Check-Ins? | Support Center | Yelp. *Yelp-Support.com*. Retrieved November 1, 2014, from http://www.yelp-support.com/article/What-are-Check-Ins?l=en_US