Vistradas: Visual Analytics for Urban Trajectory Data

Luciano Barbosa[1], **Matthías Kormáksson**[1], **Marcos R. Vieira**[1], **Rafael L. Tavares**[1,2], **Bianca Zadrozny**[1]

[1]IBM Research – Brazil

[2]Univ. Federal do Rio de Janeiro – Brazil

{lucianoa,matkorm,mvieira,rafaelt,biancaz}@br.ibm.com

Abstract. *In the past few years a growing number of cities have started monitoring the position of public transportation vehicles using GPS devices. Most of these trajectory data are released in raw format and usually have issues, such as measurement errors. Providing insights from these valuable (and noisy) data is a major challenge in larger cities. In this paper we present a system, called Vistradas, for visual analytics of urban trajectory data. Vistradas allows users to analyze use cases related to trajectories of public buses such as: analysis of bus uniformity, verification of bus route, and the impact of events in bus traffic. Our proposed Vistradas system helps user to get insights into various aspects of public transportation.*

1. Introduction

There is a fast growing use of new technologies (e.g., cheap GPS devices, ubiquitous sensor and cellular networks) that generate large amount of data in the form of trajectories. Basically, a trajectory is a sequence of pairs (location, timestamp), which may contain some other attributes (e.g., temperature, velocity), generated by a moving object. Since trajectory data in their raw format do not bring much valuable information to users, data analytics have a key role to help people get useful insights from trajectory data.

This paper presents an on-going research project for visual analytics of urban trajectory data. Vistradas is a Web-based visualization system that allows users to visualize use cases related to trajectories of public vehicles. In this paper, we make use of trajectories of public buses in the city of Rio de Janeiro. As the data presented noisy information, we implemented a pre-processing step to deal with these issues, and also a data normalization step for our use cases (see Section 3). For the processed data, we created use cases (described in Section 4) that focused on quality of bus service, namely: **(1)** analysis of bus uniformity, **(2)** verification of bus route, and **(3)** the impact of events in bus traffic. Data reports from these use cases can help city bus authority to inspect and monitor bus services in the city. Finally, in Section 5, we briefly describe the currently research we are conducting with the Vistradas system.

2. Related Work

There are several research and commercial systems that provide features to monitor and visualize trajectory data. For instance, [Pu et al. 2013] uses taxi trajectory data to build visual reports for monitoring city traffic. AITVS [Lu et al. 2006] is a visualization system to analyze, monitor and report traffic conditions. CubeView [Shekhar et al. 2002] is a

Web-based visualization package for building summarizations of traffic trends on top of a multi-dimensional data warehouse. [Albuquerque et al. 2013] describes a system to monitor truck fleets, which can be integrated with tweet data to georeference traffic-related facts. SeMiTri [Yan et al. 2010] is a system that semantically enriches trajectories (i.e., sequence of places where a trajectory has passed/stayed) by using other geographic data sources.

More related to our proposed system are CommonGIS [IAIS 2014] and M-Atlas [M-Atlas 2014]. CommonGIS is a general GIS tool with some capabilities for cleaning, integrating and reporting basic summary statistics of trajectory data. M-Atlas provides mechanisms to store and query trajectory data (e.g., range, nearest neighbor queries with a temporal/relational predicate), as well as features for mining trajectory patterns. Examples of such patterns include finding frequent pattern of movement, finding dense areas of traffic jams, among others.

Nevertheless, our proposed Vistradas system is different from previous systems since we are not only building visualizations/reports for traffic monitoring or a tool to store and query trajectories. Instead, we are interested in providing tools for cleaning, managing, integrating, and analyzing statistically large urban trajectory data in order to provide city insights to public managers. To the best of our knowledge, such analysts is not facilitated in previous works. In particular, we are not aware of any previous systems that analyze the use cases in this paper.

3. Data Analysis

In order to describe the uses cases we first characterize the GPS data obtained from buses operating in the city of Rio de Janeiro. The raw trajectory data was obtained from September 26, 2013 to January 9, 2014. It contains information for more than 9,000 buses of around 400 bus lines in Rio de Janeiro[1]. In total there are more than 100 million GPS entries for the mentioned period.

Each GPS data entry has the location of a bus (latitude and longitude), timestamp, bus ID, line ID, and bus velocity. The time between consecutive GPS measurements ranges from anywhere under a minute to over 10 minutes, with an average of 4 minutes. We also had access to GTFS data, which contain general information about the bus routes, such as bus stop locations and expected schedules. In general each route consists of two trips, one going from origin to destination and the second representing the return trip. The GTFS data contain a complete definition of each such trip as a sequence of line segments tracing the streets of the route from origin to destination.

The raw trajectory data itself presented problems, such as: no information about the direction in which the bus is traveling; and, in some cases, wrong latitude/longitude positions and poor time resolution (i.e., much higher than 10 minutes). To deal with these issues and also to normalize the data, we implemented a pre-processing component, depicted in Figure 1. In the following, we present the main steps of this process.

Cleaning: the first step in the pre-processing phase is, for each bus trajectory tr_i in the dataset, to remove GPS entries with distance higher than δ_s (e.g., δ_s=100 meters) from expected route, and trajectories with GPS resolution higher than δ_t (e.g., δ_t=10 minutes);

[1]The GPS data can be obtained in `www.rio.rj.gov.br/web/dadosabertos`.

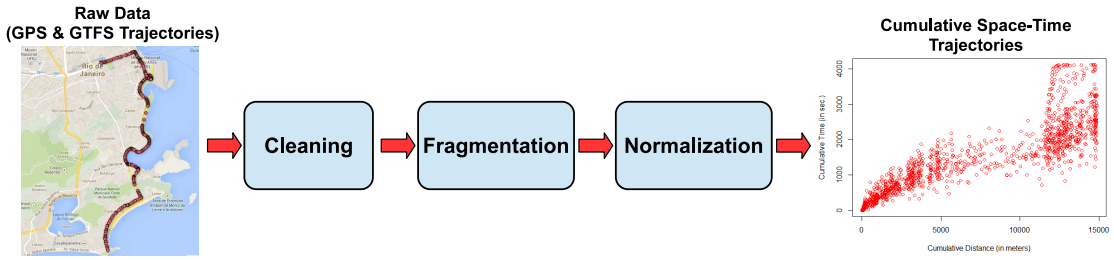

Figure 1. Data pre-processing workflow: cleaning, fragmentation and normalization of raw trajectory data to build cumulative space-time bus movements.

Fragmentation: from the cleaned trajectory dataset, the next step is to discover the bus direction for a given period of time. We fragment tr_i where each fragment represents the bus trip in one of the line directions. To perform these tasks we rely on the expected routes of the line directions using the GTFS data;

Normalization: the final step is to normalize the segmented bus trajectories, which are going to be used in the use cases. This step calculates the cumulative time and distance from the beginning of the bus route for a given bus trip (see examples of cumulative space-time trajectories in Figure 1). To calculate the cumulative distance, we measure the Euclidean distance between consecutive measurements in the expected route provided in the GTFS data (high resolution data). Then, we project a given GPS coordinate on its expected route and calculate its distance to the previous GPS coordinate in the trajectory. Since it is very rare (from our data) to have GPS measurements at the beginning of the bus trips t_0, we calculate t_0 by interpolating the time using the last reported GPS measurement before t_0 and immediately after t_0. Once t_0 is computed, all the other times relatively to t_0 can be easily calculated along the bus route.

4. Use Cases Description

We now describe three use cases our Vistradas system supports using the normalized trajectory data we previously described. These use cases are all related to quality of service of bus lines in the city of Rio de Janeiro.

4.1. Analysis of Bus Uniformity

The first use case is *bus bunching*, a common problem that occurs when two or more buses of the same line are very close to each other along their routes. To detect this problem, for each line we compute a *bus_bunching_score* defined by Equation 1. This score measures how much the spatial distribution of buses along their routes deviates from an expected distribution. A bus line with higher *bus_bunching_score* is more likely to suffer from the bus bunching problem.

The steps to calculate *bus_bunching_score* are: **(1)** for a given bus line, we compute the expected distance between buses by dividing the total number of buses n running on a given time interval (e.g., 4 hours) by the route length l_r; **(2)** then, for all adjacent pairs $n-1$, we calculate how much buses deviate from the expected distance by subtracting it from the observed distance d_i of each pair of adjacent buses (on route) in a given line; **(3)** the final bus bunching score of a line is then the average deviation.

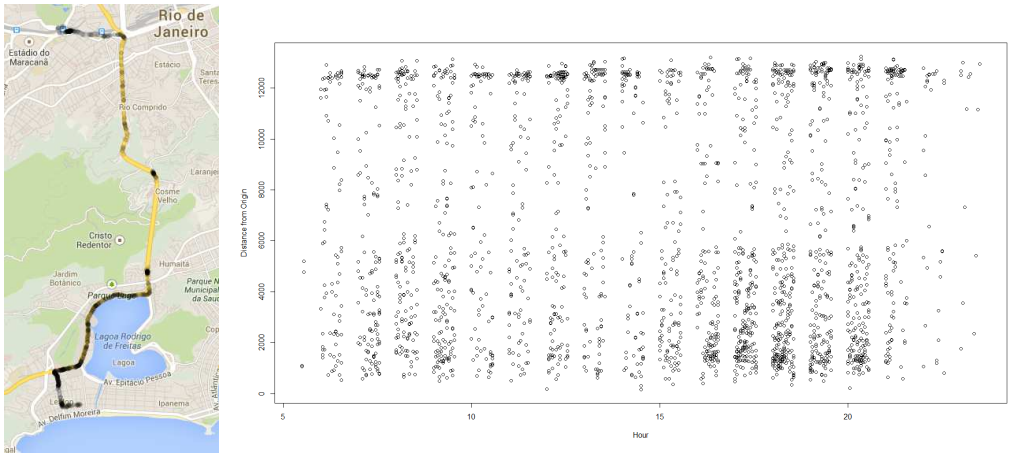

Figure 2. Space-time distribution of buses for line 460 on Nov. 13, 2013.

$$bus\_bunching\_score = \frac{1}{n-1}\sum_{i=1}^{n-1}(d_i - \frac{n}{l})^2 \quad (1)$$

For the *bus bunching* analysis, we first rank the buses according to the *bus_bunching_score* value during a particular day. Then, given a ranked bus line, we show a map with the corresponding distribution of buses and a graph presenting the distribution of the buses in time and space. For this visualization, each dot on the map (and on the graphic) represents a single bus. For example, Figure 2 shows a high bus bunching score for the distribution of buses in line 460 on November 13, 2013. From this example we can observe that buses were very close to each other during rush hours (between 4pm and 8pm) near 4km of their origin route point (at Rodrigo de Freitas Lagoon).

4.2. Verification of Bus Route

The second use case involves the verification of whether buses are respecting their expected routes. This can be useful, for instance, for contract auditors interested in knowing whether the bus companies are respecting their routes, or even for bus companies ensuring their drivers are respecting the expected routes.

Our route verification algorithm works as follows: **(1)** given the expected route r (obtained from the GTFS data) for a line l_r, and the bus GPS measurement p, we calculate the shortest Euclidean distance $min\_dist(p, r)$ between every p for a particular bus in l_r and the sequence of lines defining r; **(2)** in the second step, we measure how much all buses in l_r deviate from the expected route r. We calculate the deviation score *deviation_score* by taking the average $min\_dist(p, r)$ over all GPS measurements of line l_r, as:

$$deviation\_score = \frac{1}{n}\sum_{i=1}^{n} min\_dist(p_i, r) \quad (2)$$

Using Vistradas we are able to detect bus lines with the highest deviations, which can help us understand the source of the problem. In Figure 3, we show an example of bus line 906 that obtained a high deviation value. We can clearly see the buses for line

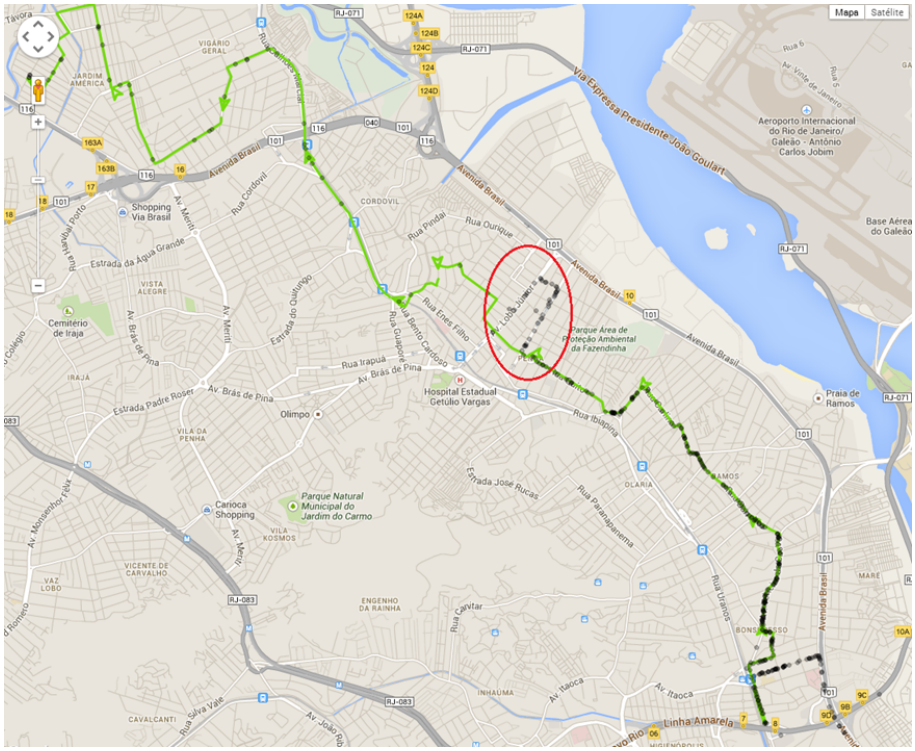

Figure 3. Route verification using the bus line 906. Green line is the expected route provided in the GTFS data, and black dots are the GPS measurements.

906 systematically changed their route on this particular day. We can also observe both the expected route (green line) and the actual route (black dots) on the same visualization for a given line, where there are many GPS measurements (black dots) far away from their expected route (green line). Since the GPS used in the buses are not so accurate in reporting their real locations, the GPS data may contain imprecise measurements, which may be due to the nature of device or occlusion in the area. Nevertheless, we can see many black dots very close to the expected route and a few ones (marked with a red circle) far away from the expected route, which may indicate a possible detour.

4.3. Impact of Events in Bus Traffic

The third use case involves the impact of events on bus traffic. For example, road constructions, natural phenomena (e.g., heavy rain, mudslide), sport or music events, among others, are some events that can have a negative impact on the city's traffic. In order to measure the impact of events, we calculate the difference between the average traffic velocity on the bus routes in a period before and after a given event. Vistradas allows users to select a particular date and then plot the difference of velocities before and after the given date. Figure 4 shows the impact of the Perimetral overpass fall on bus route 121. The region in red color shows a significant impact on bus velocity: a decrease of 8Km/h. We also present the information in a line graphic, so that the user can compare the difference easily and, if desired, select a point in the line to see on the map. This kind of information can be very useful for city planners to verify the impact of planned and unplanned events on city's traffic.

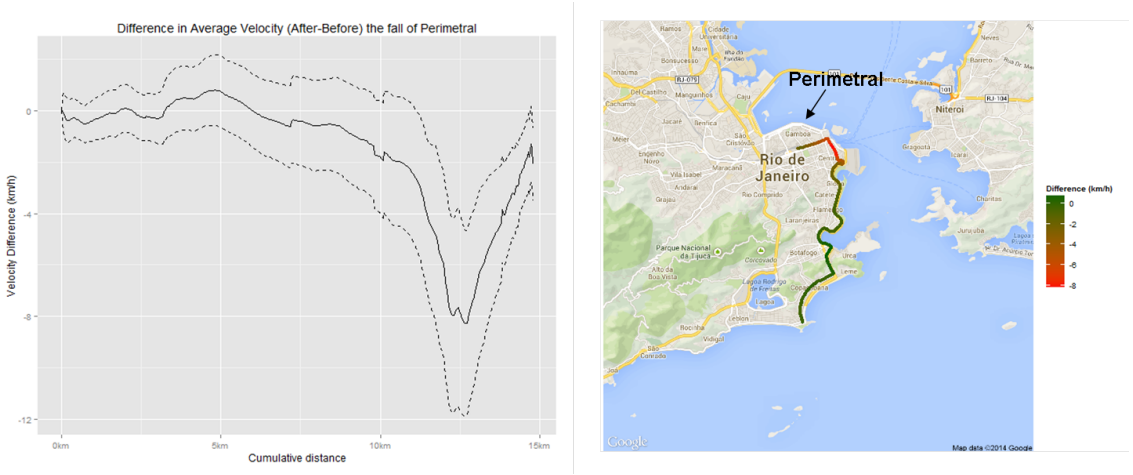

Figure 4. The traffic impact before and after the implosion of Perimetral overpass.

5. Conclusion and Ongoing Research

This paper describes Vistradas, a Web-based system that provides visual analytics of urban trajectory data. This paper shows a few use cases supported by our Vistradas system using real bus GPS data.

As for the current status of our Vistradas system, we are developing two new use cases for real problems that the city of Rio de Janeiro faces: **(1)** we are integrating the bus GPS data with data coming from weather stations, microbloging, and news websites to get new insights of the city; and **(2)** using the historical GPS data, we are building predictive models to make time-prediction of buses [Kormaksson et al. 2014] for a few particular scenarios (e.g., when will the bus line 121 arrive at my stop 15? when will I get to my final destination?).

References

Albuquerque, F., Casanova, M., Macêdo, J., Carvalho, M., and Renso, C. (2013). A proactive application to monitor truck fleets. In *Proc. of IEEE MDM*, pages 301–304.

IAIS, F. (2014). CommonGIS. www.iais.fraunhofer.de/1871.html?&L=1.

Kormaksson, M., Barbosa, L., Vieira, M., and Zadrozny, B. (2014). Bus travel time predictions using additive models. In *Proc. of IEEE ICDM*.

Lu, C.-T., Boedihardjo, A. P., and Zheng, J. (2006). Aitvs: Advanced interactive traffic visualization system. In *Proc. of IEEE ICDE*, pages 167–167.

M-Atlas (2014). M-Atlas. http://www.m-atlas.eu/.

Pu, J., Liu, S., Ding, Y., Qu, H., and Ni, L. M. (2013). T-watcher: A new visual analytic system for effective traffic surveillance. In *Proc. of IEEE MDM*, pages 127–136.

Shekhar, S., Lu, C. T., Liu, R., and Zhou, C. (2002). Cubeview: a system for traffic data visualization. In *Proc. of IEEE Int'l. Transp. Sys.*, pages 674–678.

Yan, Z., Spremic, L., Chakraborty, D., Parent, C., Spaccapietra, S., and Aberer, K. (2010). Automatic construction and multi-level visualization of semantic trajectories. In *Proc. of ACM SIGSPATIAL*, pages 524–525.