Visual analysis of historical bus data

Bruno P. Schettino

Instituto de Computação – Universidade Federal Fluminense (UFF)
Rua Passo da Pátria, 156, São Domingos - Niterói – RJ - CEP: 24210-240

bschettino@id.uff.br

Abstract. *Urban mobility is a challenge for major cities around the world. The traffic jam is one of the most common problems in large cities and has serious consequences for the quality of life for all urban residents. Transport is one of the great challenges of any city that has an interest in becoming a smart city and is a critical component of urban design. This work focuses on the visual analysis of bus data from the city of Rio de Janeiro. The main objective is to make queries on historical bus data, compare different moments of traffic and visualize these data on a map in an interesting way.*

1. Introduction

Transport is one of the great challenges of any city that has an interest in becoming a smart city and is a critical component of urban design. The traffic jam is one of the most common problems in large cities and has serious consequences for the quality of life for all urban residents.

According to Thiagarajan et al. [1], the real-time bus tracking, where available, has been well received by transit riders. Knowing where a bus is at the moment and when it will arrive in a certain position reduces the waiting time, increasing the efficiency and improving the safety and comfort of the users.

For doing this kind of prediction with an acceptable precision, it is necessary to understand the traffic as a whole and also understand the patterns and irregularities of the data provided. The goal of this paper is to provide interfaces to help us to understand the traffic behavior through the organization of the bus positions data in a specific moment and comparing distinct moments to understand how they are different.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 shows briefly the components of the background used to support the visualization part of the work. Section 4 details the visualization features, and Section 5 has the conclusions of the work done and presents future work.

2. Related Work

There are several works in the area of bus tracking aiming to predict the bus arrival time at a given point as [2]–[5]. As these works focus on suggesting methods, they could be done without concerning about the input data itself.

When you want analyze real data, you need to verify the consistency of the information. You might need to discard some records. For instance, when a bus is standing at the garage with the GPS turned on or when the record is duplicated. But sometimes, it's hard to identify all cases that should make us discard a record.

A visualization of the data in a map can help us to identify and understand some of these cases, besides helping to see the valid data and understanding the bus traffic. There are some works in the visualization area as [6]–[10]. These works aims to show the traffic only a specific moment of the time. Our work consists in make queries on historical bus data, compare different moments of traffic and visualize these data on a map in an interesting way.

3. Background

To support the visualization part of the work, a background implementation is necessary. We need a collection of relevant data to be shown to users and we need to fetch this data as fast as we can. Section 3.1 presents the data collection and Section 3.2 shows the loading and filtering strategies used.

3.1. Data collection

For this paper, we used the data provided by the Rio de Janeiro City Hall through their server [11]. This server provides a JSON file that is generated every minute with the most updated information about the position, speed and line of every bus circulating in Rio de Janeiro city. Since a new file is generated every minute, if just access the information and do not storage it somehow, we would not be able to access that information later. Because of this, we have created a script that requests and stores the JSON file with the most updated information every minute. In the end of the day, all the JSON files are compressed in a ZIP file named with the date timestamp. This ZIP files are available in our server [12].

Beside the buses positions, we can access information about the bus lines. The server also provides the line stops positions and the line route positions. Since this information does not change frequently, we only requested and stored them once.

3.2. Loading and filtering strategies

After collecting the information, we need to load them to a database, so we can easily perform fast searches. Since the ZIP files mentioned in the previous section are available in our server, we can download them and read the JSON files to load their data to our database.

After some analysis, we noticed that some data appeared to be abnormal. For situation like this, we created a new table called Disposals. This table keeps all the records considered abnormal and the Bus Positions table keeps all the records considered normal.

In the first place, we noticed that many records were duplicated. The duplication can cause problems, for instance when we want to calculate the speed average of a bus. If a record is duplicated, the average would consider the same information twice and the result would be wrong. To avoid this kind of problem, when we are inserting a new bus position, we compare the timestamp of the new record with the timestamp of the last inserted record of the same bus. If the timestamp of the new record is not greater than the last one, this new record is inserted in the Disposals table. Otherwise, it will be inserted in the Bus Positions table.

Besides this first problem, we also noticed that some records had above normal speed. For instance, we found speeds up to 1256 kilometers per hour. Again, this kind of value would cause errors when we want to calculate a speed average for instance. Then, to avoid this kind of error, when we are inserting a new bus position, we compare the position of the new record with the position of the last inserted record of the same bus. If the distance between the position of the new record and the position of the last one is greater than 2 kilometers, this new record is inserted in the Disposals table. Otherwise, it will be inserted in the Bus Positions table. We used the 2 kilometers distance threshold based on the fact that if a bus is traveling at 120 kilometers per hour, the maximum distance that it would have traveled in one minute would be 2 kilometers. Even though that a bus in Rio de Janeiro city should not travel over 80 kilometers per hours, even in highways, we decided to set the threshold to 120 kilometers per hour to consider periods when the road is clear and the bus can travel in a higher speed.

The last problem that we noticed was that some buses do not turn off the GPS sensor when they are inactive, such as when they are standing in the garage. We could notice several records in sequence, which a bus had, speed zero and only a few meters traveled. Even when the bus is standing in the garage, the subsequent records may have few meters distance between them due to the attached error in GPS devices. This kind of problem would cause errors in some kinds of analysis, as the two other problems shown above. To eliminate this problem, when we are inserting a new bus position, we compare the position of the new record with the position of the last inserted record of the same bus. If the distance between the position of the new record and the position of the last one is lower than 15 meters, this new record is inserted in the Disposals table. Otherwise, it will be inserted in the Bus Positions table.

4. Visual Analysis

To help us to identify and understand problems like those presented on Section 3.2, we created the visual analysis. A good way to view information about bus positions is showing the records in a convenient manner on a map. For the visual analysis, we decided to use Google Maps API [13] as the base view of the work. The visual analysis is divided in three subsections: Section 4.1 introduces the line positions interface; Section 4.2 shows the positions heat map interface; and Section 4.3 presents the speed heat map interface.

4.1. Line Positions

In the line positions interface is possible to choose a bus line by it's number, and visualize the route made by the buses operating in this line. A switch can be turned on or off to show or hide the stop points of the line. Figure 1 shows an example of the line positions interface, in which we can see the route of the line 10.

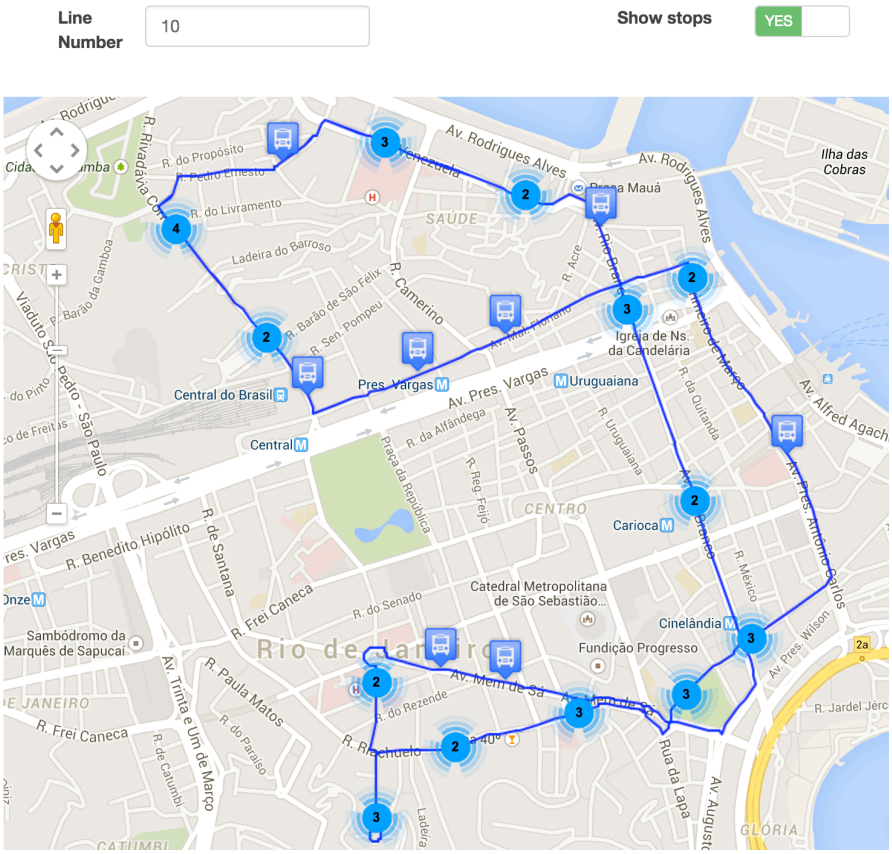

Figure 1. Line positions interface example

4.2. Positions Heat Map

In the positions heat map interface is possible to choose a bus line by its number, and visualize a heat map composed by the positions of the buses operating in this line. As in the line positions interface, a switch can be turned on or off to show or hide the stop points of the line. This interface can show two types of heat maps: query a single situation and compare two situations.

The query heat maps also have other 3 filtering options besides the line option: by the time of the day; by the day of week; and by a specific date. These filtering options can be combined with the line number filtering option to show a different heat map for each situation chosen by the user.

The first one is filtering by the time of the day. As the traffic can vary greatly during the day, this kind of filtering is useful for enabling the user to see the heat map generated in different moments of the day. For instance, it is expected to be more buses operating in the range of 6 AM to 10 PM than at other times.

The second filtering option is the day of week. As the time of day, it is a interesting filtering option because the traffic can vary greatly from one day of the week to another. For instance, it is expected that the traffic is more intense in Fridays than on Sundays.

The last filtering option is the date. This filtering option was created to see the heat map in a specific day. For instance, the user can choose this option to see the heat

map in a day when a big accident has occurred. This option can be combined with the time of the day option but cannot be combined with the day of the week. This restriction was implemented to avoid incompatibility errors. For instance, if the combination was allowed, the user could select January 1, 2014 (which was a Wednesday) in the date field, and select Sunday in the day of week field. So, combining both filters, the result would be empty.

In the query tab, the results of filtering are shown using Google Maps Heat Map. In this visualization, the red color is used to paint the areas with the highest number of points, the green color is used to paint the areas with the lowest number of points and the intermediate colors are used to paint areas with intermediate number of points. Thus, the fewer points in one area, it is painted with a color closest to green and the more points, the area is painted with a color closest to red. Figure 2 presents a positions heat map interface example, in which we can see the areas that have higher and lower number of bus positions of the line 10, in Tuesdays, between 12 and 7 PM.

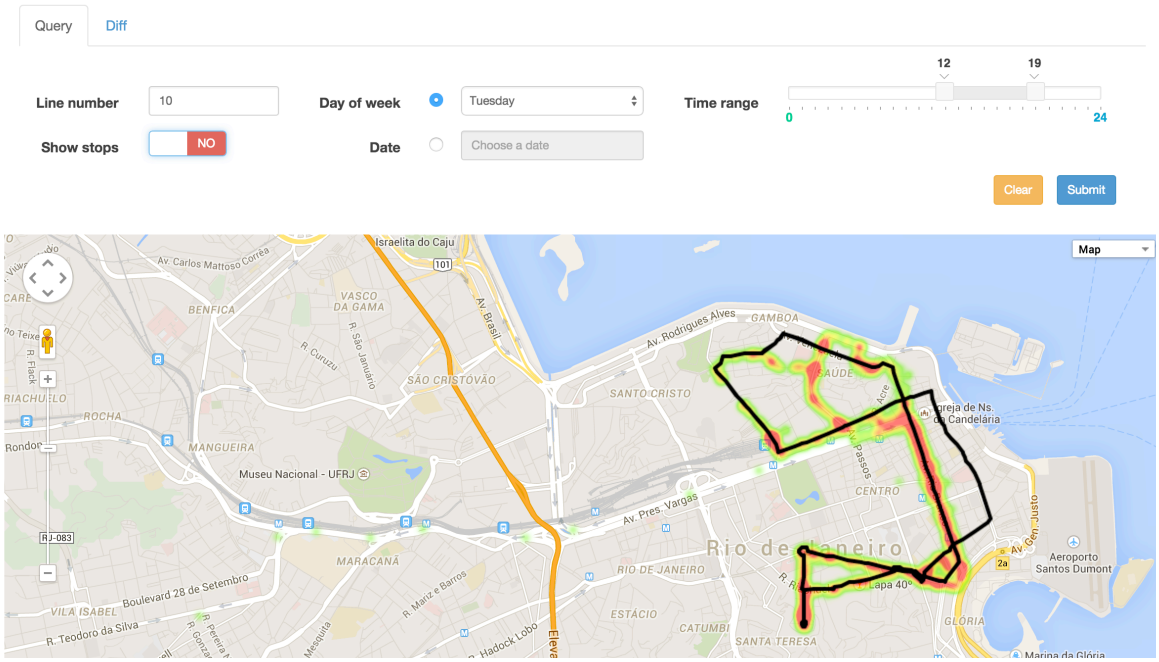

Figure 2. Positions heat map query interface example

In the diff tab, we can compare two situations and see a heat map with the distinction between them. The points outside of the line route are discarded because it is not trivial how to calculate the difference between far points. So, for each point inside the line route, we compare the amount of bus positions near it in both situations and calculate the difference between the two values in percentage. To draw the colors, we preferred to use the Google Maps Polylines feature rather than the Heat Map feature because it is easier to fill the entire line route and draw the specific color that we want.

We have five colors to represent the degree of difference between the situations. In the color scale, black means that the difference between the situations is lower than 10 %. Yellow and orange colors means that the difference is between 10% and 30%. In case of yellow, means that the value in the base situation is lower than the value in the second situation and in case of orange, means that the base situation value is higher than

the second one. Red and green means that the difference is higher than 30%. In case of green, means that the value in the base situation is lower than the value in the second situation and in case of red, means that the base situation value is higher than the second one. Figure 3 shows an example of the positions heat map diff interface, in which we can see the comparison between the line 10, in Tuesdays, between 12 and 7 PM and the same line in Thursdays, at the same time range.

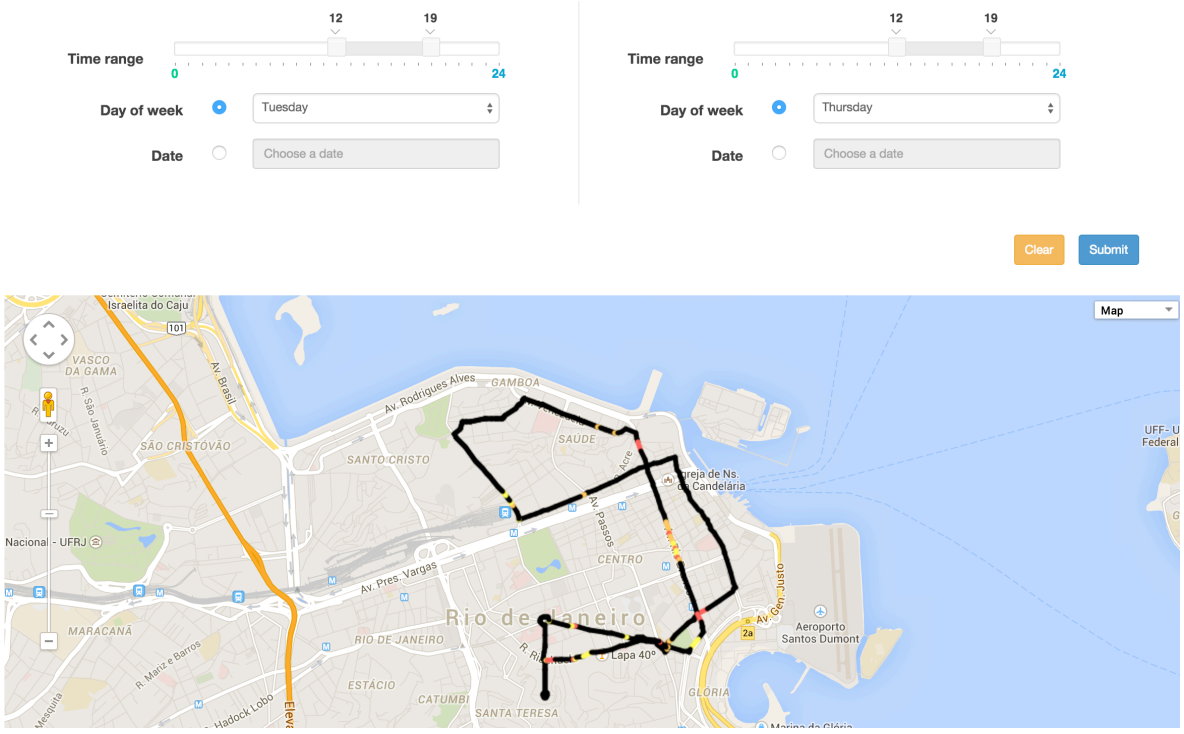

Figure 3. Positions heat map diff interface example

4.3. Speed Heat Map

In the speed heat map interface is possible to choose a bus line by its number, and visualize a heat map composed by the speed of the buses operating in this line. This interface has the same filtering options presented in the positions heat map and it also has the query and diff tabs.

In both, query and diff tabs, the results of filtering are shown using Google Maps Polylines for the same reason we used it in the positions heat map. In the query visualization, we pick each point inside the line route and calculate the speed average near it.

We have four colors to represent the speed intervals. The red color is used to paint the paths that have average speed lower than 15 kilometer per hour. The orange is used to paint the paths in which average speeds are between 15 and 30 kilometers per hour. The yellow color is used to paint the paths in which average speeds are between 30 and 60 kilometers per hour. The green color is used to paint the paths that have average speed higher than 60 kilometer per hour. Figure 4 shows an example of the speed heat map query interface, in which we can see the speed average through the line 10, in Tuesdays, between 12 and 7 PM.

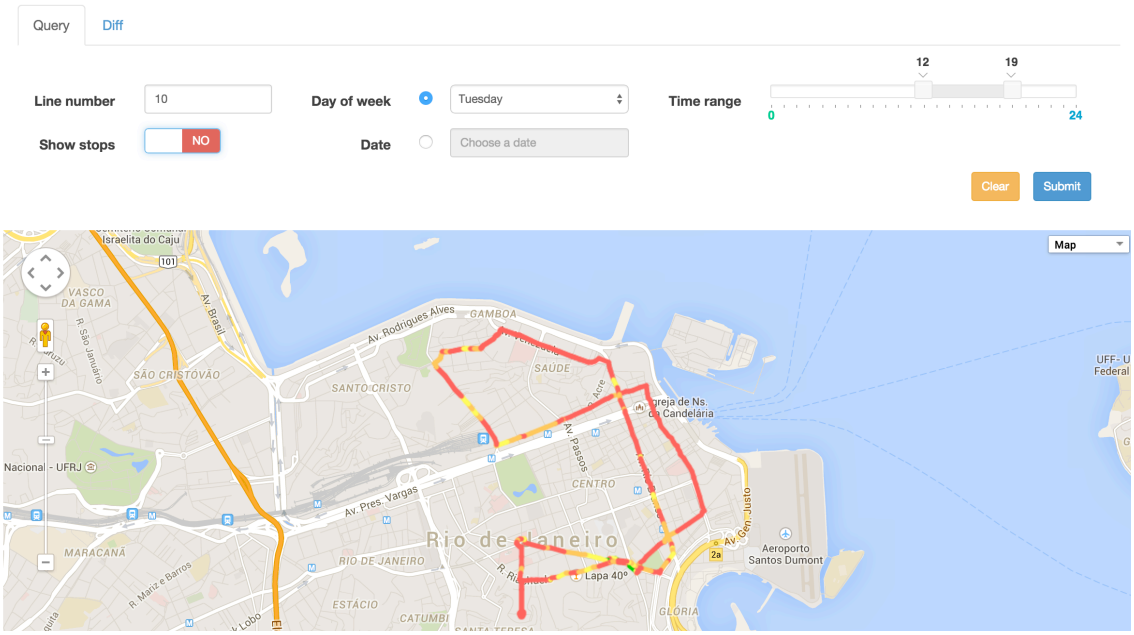

Figure 4. Speed heat map query example

In the diff tab, we can compare two situations and see a heat map with the distinction between them. For each point inside the line route, we compare the speed average near it in both situations and calculate the difference between the two values, subtracting the second value from the base value. Figure 5 presents an example of the speed heat map diff interface, in which we can see the comparison between the line 10, in Tuesdays, between 12 and 7 PM and the same line in Thursdays, at the same time range.

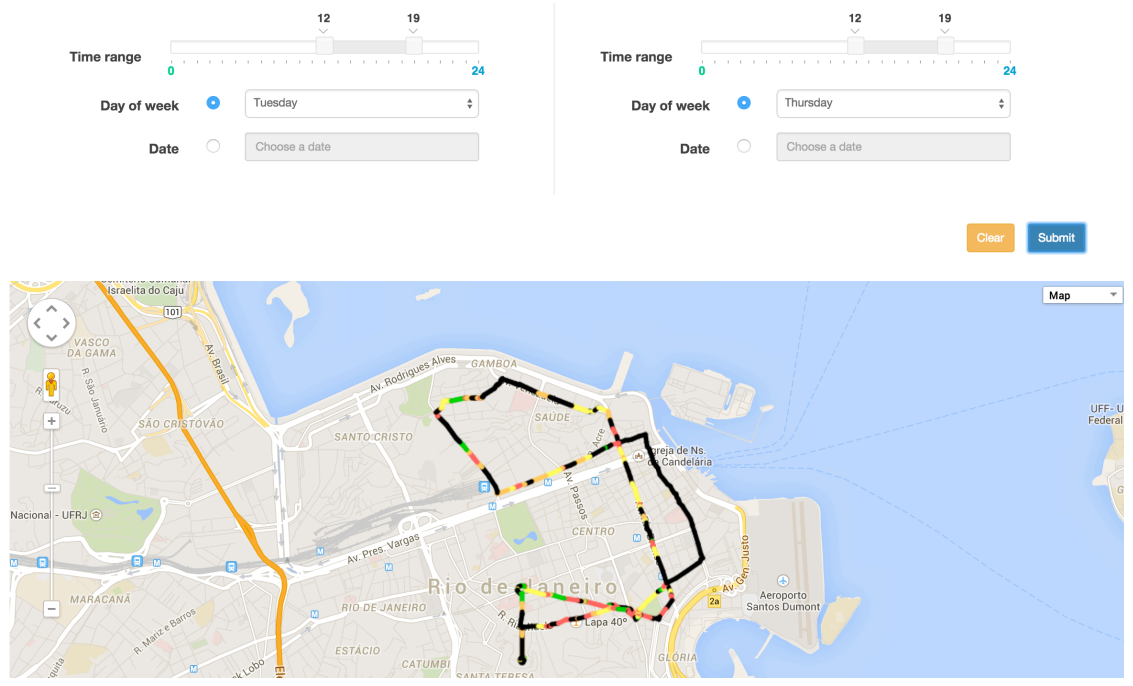

Figure 5. Speed heat map diff interface example

We have five colors to represent the degree of difference between the situations. In the color scale, black means that the difference between the situations is lower than 5 kilometers per hour. Yellow and orange colors means that the difference is between 5 and 20 kilometers per hour. In case of yellow, means that the value in the base situation is lower than the value in the second situation. In case of orange, means that the base situation value is higher than the second one. Red and green means that the difference is higher than 20 kilometer per hour. In case of green, means that the value in the base situation is lower than the value in the second situation. In case of red, means that the base situation value is higher than the second one.

5. Conclusion

This paper presents a system that provides visual analysis for historical bus data for helping to understand the bus traffic. Besides the simple query, this paper also discusses the importance of comparing different scenarios.

For future work, it is possible to create visual interfaces for discarded data, filtering it by the disposal reason combined with the filtering options presented in this paper. Another possible work is to implement the diff between heat maps outside of the line route.

References

[1] A. Thiagarajan, J. Biagioni, T. Gerlich, e J. Eriksson, "Cooperative Transit Tracking using Smart-phones".

[2] J. Gong, M. Liu, e S. Zhang, "Hybrid dynamic prediction model of bus arrival time based on weighted of historical and real-time GPS data".

[3] R. Jeong e L. R. Rilett, "Bus Arrival Time Prediction Using Artificial Neural Network Model", *IEEE Intell. Transporlatlon Syst.*, 2004.

[4] P. Zhou, Y. Zheng, e M. Li, "How Long to Wait?: Predicting Bus Arrival Time with Mobile Phone based Participatory Sensing", *ACM Int. Conf. Mob. Syst. Appl. Serv. MobiSys*, 2012.

[5] C. Coffey, A. Pozdnoukhov, e F. Calabrese, "Time of Arrival Predictability Horizons for Public Bus Routes", *ACM SIGSPATIAL Int. Workshop Comput. Transp. Sci. IWCTS*, 2011.

[6] L. Barbosa, M. Kormáksson, M. R. Vieira, R. L. Tavares, e B. Zadrozny, "Vistradas: Visual Analytics for Urban Trajectory Data". .

[7] C.-T. Lu, A. P. Boedihardjo, e J. Zheng, "AITVS: Advanced Interactive Traffic Visualization System", in *Proceedings of the 22nd International Conference on Data Engineering, 2006. ICDE '06*, 2006, p. 167–167.

[8] J. Pu, S. Liu, Y. Ding, H. Qu, e L. Ni, "T-Watcher: A New Visual Analytic System for Effective Traffic Surveillance", in *2013 IEEE 14th International Conference on Mobile Data Management (MDM)*, 2013, vol. 1, p. 127–136.

[9] S. Shekhar, C. T. Lu, R. Liu, e C. Zhou, "CubeView: a system for traffic data visualization", in *The IEEE 5th International Conference on Intelligent Transportation Systems, 2002. Proceedings*, 2002, p. 674–678.

[10] Z. Yan, L. Spremic, D. Chakraborty, C. Parent, S. Spaccapietra, e K. Aberer, “Automatic Construction and Multi-level Visualization of Semantic Trajectories”, in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, New York, NY, USA, 2010, p. 524–525.

[11] “Portal da Prefeitura da Cidade do Rio de Janeiro - rio.rj.gov.br - Portal de dados abertos”. [Online]. Available at: http://data.rio.rj.gov.br/. [Acessado: 26-nov-2014].

[12] “Zip files containing Rio de Janeiro City bus fleet data”. [Online]. Available at: http://sel.ic.uff.br/bus/. [Acessado: 26-nov-2014].

[13] “V3: a solução para aplicativos do Google Maps para dispositivos desktop e móveis - API Javascript do Google Maps v3 — Google Developers”. [Online]. Available at: https://developers.google.com/maps/documentation/javascript/?hl=pt-br. [Acessado: 26-nov-2014].