SOFTWARE TOOL FOR ANALYSIS AND VISUALIZATION OF GPS TRACKS IN URBAN ENVIRONMENTS

Héctor Cogollos Adrián
PhD Student, University of Burgos, Spain
Santiago Porras Alfonso
Lecturer, University of Burgos, Spain
Bruno Baruque Zanón
Senior Lecturer, University of Burgos, Spain

ABSTRACT

Traffic in urban environments vary significantly depending on the areas of the city and the time of day. Nowadays, thanks to the large number of GPS devices that are integrated in all kinds of devices, it is possible to make a quantitative analysis of traffic. This contribution presents an application developed to analyse this information in a simple way and with a visual representation. One of its main advantages is that it is adaptable to any city in the world, as its internal algorithms adapt to the available data are presenting and adjusting it to the traffic through the city. This is done by extracting information directly from the data provided by GPS devices moving around the city. In addition, an analysis of the execution times of all application processes is presented to determine which parts involve a higher execution cost and determine the overall scalability of the application.

1. INTRODUCTION

Nowadays, the general availability of devices capable of tracking the movements of a subject or vehicle makes movement data collection easier than ever. However, once the data is collected, a more difficult task must be completed to extract relevant information about this monitoring: the data must be cleaned (Chen *et al.*, 2016), organized (de Almeida *et al.*, 2020), aggregated, etc. (Zheng, 2015). This problem grows exponentially when it is necessary to analyse massive amounts of vehicles or people movements at the same time.

This work presents a web application prototype that facilitates the completion of the aforementioned analysis tasks. The tool takes individual regular GPS tracking files and performs operations to filter track errors, detect traffic clusters and nodes, calculate traffic summary indicators between nodes, and display all this information on a dashboard. In addition, it includes an interactive map and additional analytical charts allowing the user to perform exploratory analyses. Additionally, it allows the processed files and their results to be exported in standard format, so they can be used in other software solutions.

The software has been tested with real data obtained from mobility activities in some worldwide cities, obtaining interesting analytical results that will be included in the contribution as examples. These examples are drawn from two datasets. The first one, with a temporal scope of up to one year, including data on up to 442 vehicles and 1,674,141 trips. The second one with a temporal scope of just over one and a half months and 12,695 routes. The geographical scope of the application can be extended to wider areas, not just urban areas, provided the user has the necessary data.

The main capabilities of the presented tool allow us to represent a set of individual GPS files in a graphical environment. It makes possible a global exploratory analysis of the complete dataset. Furthermore, it is possible combine the tool with other analytical software. Finally, the flexibility of its design that makes it easy to extend its functionality with additional data analysis algorithms.

2. STATE OF THE ART

There are three types of commercial solutions on the market. Some are completely focused on the end user and do not allow a general traffic analysis but offer concrete solutions to individual movements (Google Maps, WAZE). The second type of applications are those that allow experienced users to analyse the data in a general way and visualise the results (CARTO, QGIS ArcGIS, and Elastic Maps). The third type are those that allow us to analyse traffic congestion or traffic events with a high level of detail (TomTom Road analytics).

In the literature, there are different techniques to analyse and visualise traffic. One of the most common and simplest method of visualization is the heat maps (Wang, Lu and Li, 2020). It is very common to use groups to classify the regions through which routes pass to analyse the traffic density. The most widespread techniques in this case are grouping by using some characteristic of the location of the data such as neighbourhoods or districts (Ibrahim and Shafiq, 2019). Another option is using clustering techniques that perform an automatic classification of the routes (Bian *et al.*, 2018). These two methods seek to analyse the relationships between the different clusters (Yuan and Raubal, 2014).

There are other analyses with the same type of data that can help us understand traffic and propose improvements. Such as the detection of anomalies and events that allow us to detect anomalous changes in traffic (Donovan and Work, 2017)(Zhong *et al.*, 2020). This also is useful for analysing the resilience of transport systems. And the detection of patterns between routes, which is used to detect which routes are the same or have a high degree of similarity (Barann, Beverungen and Müller, 2017). These techniques are widely used in car sharing services.

Unlike these methods, in this contribution we perform a classification of the route's points. The objective is that the extracted clusters adapt to the distribution of streets and traffic in the city, getting a visual representation of the traffic status through a bidirectional network that is easily understandable.

3. METHODOLOGY

This prototype is based on the methodology described below (and represented on Figure 1), firstly explaining the data structure. Then the application processes are detailed until the graphical representation of the information is obtained.

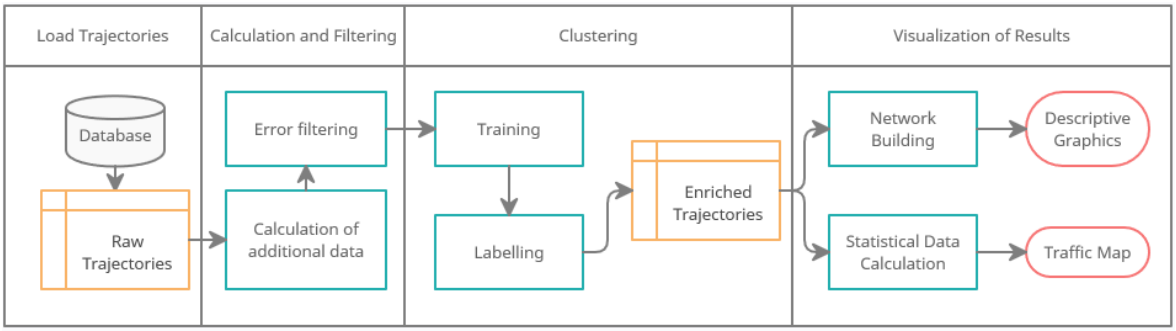

Figure 1 Pipeline of the operations performed by the application

Figure 1 shows the different phases through which the data passes until it is displayed to the user. In the first phase, the data is loaded into the application's memory. In the second phase, data such as distance, speed and time of the registered movements are calculated. This phase includes the elimination of the measurement's errors and outliers. In the third phase, a classifier is trained using this data and the data is grouped according to the geographical distribution. In the last phase, with the data applied and classified, a complex network representing the displacements is generated and the statistics are calculated. Both, the complex network and the statistics are displayed graphically to the user. In the following sections, each step is explained in more detail.

3.1. Trajectories

The basic data used by the application are GPS trajectories or routes. Which are used throughout the execution cycle of the application to perform the analysis.

The trajectories or routes consist of a series of geospatial points ordered chronologically that describe the route of an element. A trajectory T_i is formed by n points P_i that are composed of longitude$x$, latitude y and instant of time$t$.

It is important that the difference between t_n and t_{n-1} has a value small enough to make the path sufficiently descriptive and large enough so that the trajectory is not too bulky. In first data set used, the difference is 15 seconds.

From these simple paths that are stored in a database, it is necessary to calculate other data for processing. First the time intervals and the distance between the points are calculated. This also allows us to calculate the speed at which the movement between these occurs which is crucial for error filtering, as it is described next.

3.2. Error filtering

For (Yan *et al.*, 2013) the trajectories have two types of errors, the first is due to the precession of the GPS itself. These errors produce a loss of accuracy of approximately ± 15 meters. We ignore these types of errors, as they are not very relevant for the analysis. The second type is due to GPS synchronisation failures. This type of error produces failures in the real location of the device recording its location in previous locations or with displacements of kilometres. This causes the speed of an element to be calculated at an unrealistic speed.

To solve these errors, we use the speed, which is the variable that produces the failures. This error detection is context dependent. Since in the experiments we use data from motor vehicles movements in urban environments, to detect if a route contains errors, we filter those that in some intervals exceed 200 km/h. If the context is different, the speed must be adapted to a higher speed than the maximum expected.

3.3. Clustering

Clustering is a data mining technique that is used to separate data into natural groups. For (Mussardo, 2019) this technique has two functions: Summarizing the data allows us to work with a smaller volume of data, which improves processing performance and simplifies visualization. Segmentation of the data allows us to analyse the behaviour of each group. Clustering can be seen as an optimisation problem where the aim is to maximise or minimise a certain variable.

In this study, we compare two widespread algorithms for this task:

- K-means: It tries to minimize the sum of the quadratic distances with the centre of the cluster. It is necessary that we indicate the number of clusters to this algorithm.
- HDBSCAN: It is a variant of DBSCAN, this algorithm reduces the distance that can exist between children until all the points are independent building a hierarchical tree. Then it keeps those branches of the tree that have a greater stability discarding their children. The stability is the number of iterations that a branch maintains the minimum number of points or without dividing. Not all data are classified.

With these techniques, we group each point of a trajectory according to its geographical location. In this way, we can summarize a trajectory as the movement between the areas into which the map is divided, which coincide with the determined clusters. This segmentation

of the trajectories allows us to analyse the relationships that exist between the different clusters.

3.4. Visualization of Results

The processing of the information concludes with the creation of an interactive map and additional analytical graphics that allow the user to perform exploratory analyses. Additionally, the application also enables the export of the already processed data.

3.4.1. Map

A complex network is built from the clustering of points, using the centroid of each group as the node of the network. To do this, the number of edges between nodes and the speed at which the movement between nodes takes place are counted. Once the network is built, it is represented graphically on a map. This is represented in a bidirectional way, with a line thickness proportional to the traffic and with a colour that varies between green and red depending on the speed.

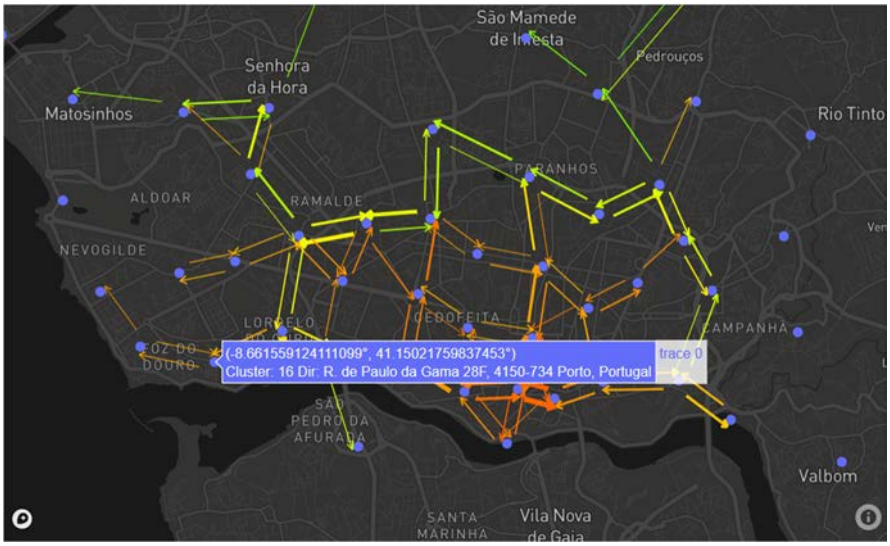

Figure 2 Map with graphical representation of traffic

All the clusters are represented on the map, but only those relationships that have a minimum support are represented. This is done so that the map is as clean and legible as possible, omitting those relationships that have little information. For this reason, there are clusters, especially those on the periphery, which do not have relations represented. As example, we show the traffic flow over the map of Porto (Portugal) can be seen on Figure 2.

3.4.2. Descriptive statistics

A series of statistics and graphs are provided to complement the information on the map. This allows to understand the data used and to analyse other factors not directly related to mobility. It also allows us to analyse the characteristics of the selected data set. As an example, Figure 3 shows a chart representing the frequency of routes completed on each hour of the day, on different days of the week.

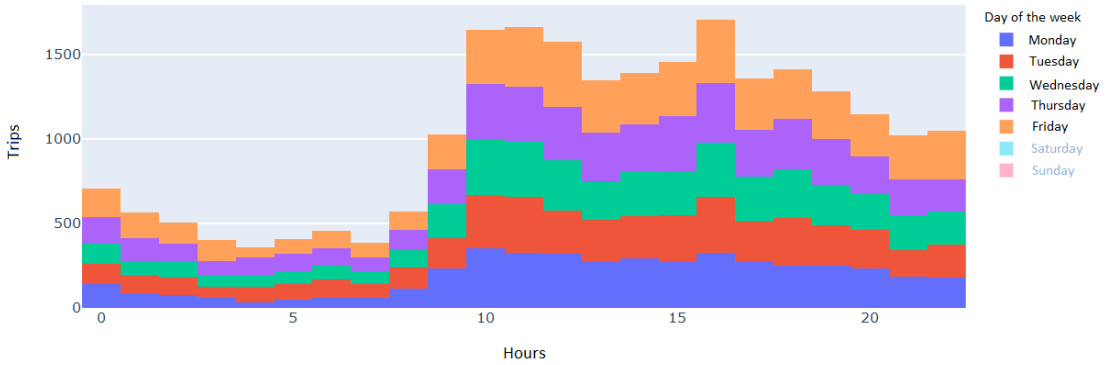

Figure 3 Example of an interactive graphic showing the frequency of routes on each hour of the day

4. EXPERIMENTS

The aim of the experiments is to evaluate the efficiency of the algorithms and them behaviour on different types of datasets. For this purpose, two different datasets are used, the first one is a dataset of taxis from Porto (*Taxi Trajectory Data | Kaggle*, 2018) and the second one is a dataset of taxis from Mexico (*Taxi Routes of Mexico City, Quito and more | Kaggle*, 2017). The difference is that the first one consists of complete routes and has 1,674,141 trips, while the second one only consists of the start and end point of the route and has only 12,695 trips. In the experiments, data loading times, clustering, calculation of the associated data and filtering, network creation and calculation of the statistical data of the routes are measured. These calculations are performed with datasets of different sizes, which allows seeing the growth of the different times. The number of points used for the calculations measures the size. The routes are randomly selected, and the data is split in such a way that there are only the set number of points for each experiment.

For the experiments, we have used a computer with 16 GB RAM, a 2-core 4-threaded processor and an SSD hard disk.

Table 1 and Table 2 show the results for each part of the execution in seconds and are the data used for the graphs in Figure 4.

	5,000	10,000	15,000	20,000	25,000
Data Loading	0.18	0.31	0.53	0.62	0.77
Clustering	0.90	1.78	2.65	3.02	3.91
Calculation and Filtering	0.65	1.33	1.93	2.79	3.13
Network Building	1.36	2.75	4.04	5.64	6.61
Statistical Data Calculation	3.36	6.70	10.21	13.41	16.22
Total	6.45	12.88	19.35	25.49	30.65

Table 1 Mexico taxi execution results. Execution times measured in seconds.

	250,000	500,000	750,000	1,000,000
Data Loading	9.63	16.44	24.08	31.58
Clustering	30.23	59.67	87.91	119.16
Calculation and Filtering	36.00	72.86	106.92	145.59
Network Building	69.88	140.52	207.71	279.83
Statistical Data Calculation	222.34	443.53	653.45	885.08
Total	368.08	733.02	1080.06	1461.24

Table 2 Porto taxi execution results. Execution times measured in seconds.

Figure 4 Graphs with the execution times of each experiment differentiating each part of the execution.

5. EXPERIMENT DISCUSSION AND LIMITATIONS

The graphs in Figure 4 show the results of the experiments, indicating the time measured in seconds, for each run and each part of the analysis. The overall time of the runs is linear, which makes it scalable. It can also be seen that all parts of the run maintain their proportionality within the same data set but vary slightly between the two sets. Finally, the runtimes for the network construction and the computation of the statistical data highlight significantly. Unlike the data loading and statistics calculation, which were optimised during the development of the program, these have not been yet optimised. These could probably be optimised or performed concurrently in the future to reduce the waiting time for the user.

The experiments are conducted with two datasets, which are limited to the mobility of two cities. This implies that only the behaviour at the urban level has been analysed. Therefore, we do not yet know the behaviour in other environments.

The application focuses only on data exploration tasks, which allows us to see the behaviour of the data over the time in which they were taken. This allows us to analyse changes in mobility in a simple way. This leads us to think that in specific contexts such as the data set used, it would allow us to make a short-term forecast of mobility.

6. CONCLUSIONS AND FUTURE WORK

The contribution has presented an application for urban traffic analysis. It is oriented towards raw route analysis and exploratory analysis. It is functional and can be used without programming skills.

The performance of the application with small to medium data sets is good using average hardware resources for computing. In the case of large datasets, the use of computing servers should be considered. In addition, the algorithms used show good scalability with linear growth.

The application has proven to be able to adapt to the data and does not need specific information from the environment in which the data was taken. Simply with low-level trajectory data the application can adapt itself to the contour of the city and to show a clear visualisation of the traffic.

The data displayed to the user on the map is clear, based on real traffic information and easily interpretable by any user. In addition, graphs and statistical tables help the user to put into context what is being visualised, giving an analysis of the characteristics of the dataset.

The complex network created from the clustering results is only used for the visual representation of the data. Moreover, it remains as a future line of work to exploit it to obtain information about traffic behaviour.

A future line of work is to improve the performance of those parts of the application that are no optimized yet, such as the calculation of statistical data. This part of the application could even be parallelised to be able to visualise the traffic status, even if this part has not yet finished executing.

REFERENCES

BARANN, B., BEVERUNGEN, D. AND MÜLLER, O. (2017) ‘An open-data approach for quantifying the potential of taxi ridesharing’, Decision Support Systems, 99, pp. 86–95.

BIAN, J. ET AL. (2018) ‘A survey on trajectory clustering analysis’

CHEN, X. ET AL. (2016) ‘Trend-residual dual modeling for detection of outliers in low-cost GPS trajectories’, Sensors (Switzerland), 16(12).

DE ALMEIDA, D. R. ET AL. (2020) ‘A survey on big data for trajectory analytics’, ISPRS International Journal of Geo-Information, 9(2).

DONOVAN, B. AND WORK, D. B. (2017) ‘Empirically quantifying city-scale transportation system resilience to extreme events’, Transportation Research Part C: Emerging Technologies, 79, pp. 333–346.

IBRAHIM, R. AND SHAFIQ, M. O. (2019) ‘Detecting taxi movements using Random Swap clustering and sequential pattern mining’, Journal of Big Data, 6(1).

KAGGLE (2017). Taxi routes of Mexico City, Quito and more. Available at: https://www.kaggle.com/mnavas/taxi-routes-for-mexico-city-and-quito

KAGGLE (2018). Taxi trajectory data. Available at: https://www.kaggle.com/crailtap/taxi-trajectory

MUSSARDO, G. (2019) Data Mining text book, Statistical Field Theor.

WANG, Q., LU, M. AND LI, Q. (2020) ‘Interactive, multiscale urban-traffic pattern exploration leveraging massive gps trajectories’, Sensors (Switzerland), 20(4), pp. 1–16.

YAN, Z. ET AL. (2013) ‘Semantic trajectories: Mobility data computation and annotation’, ACM Transactions on Intelligent Systems and Technology, 4(3).

YUAN, Y. AND RAUBAL, M. (2014) ‘Measuring similarity of mobile phone user trajectories- a Spatio-temporal Edit Distance method’, International Journal of Geographical Information Science, 28(3), pp. 496–520.

ZHENG, Y. (2015) ‘Trajectory data mining: An overview’, ACM Transactions on Intelligent Systems and Technology, 6(3).

ZHONG, R. X. ET AL. (2020) ‘Modeling double time-scale travel time processes with application to assessing the resilience of transportation systems’, Transportation Research Part B: Methodological, 132, pp. 228–248.