(ESCORT)Visual Analytics of Trajectories for Selecting Billboard Locations

(Loai Hamdy, Mina Kamal, Amr Ali Farrag, Ahmed Wael

Mostafa Abdo, Mai El-Shehaly, Youssef Moubarak)

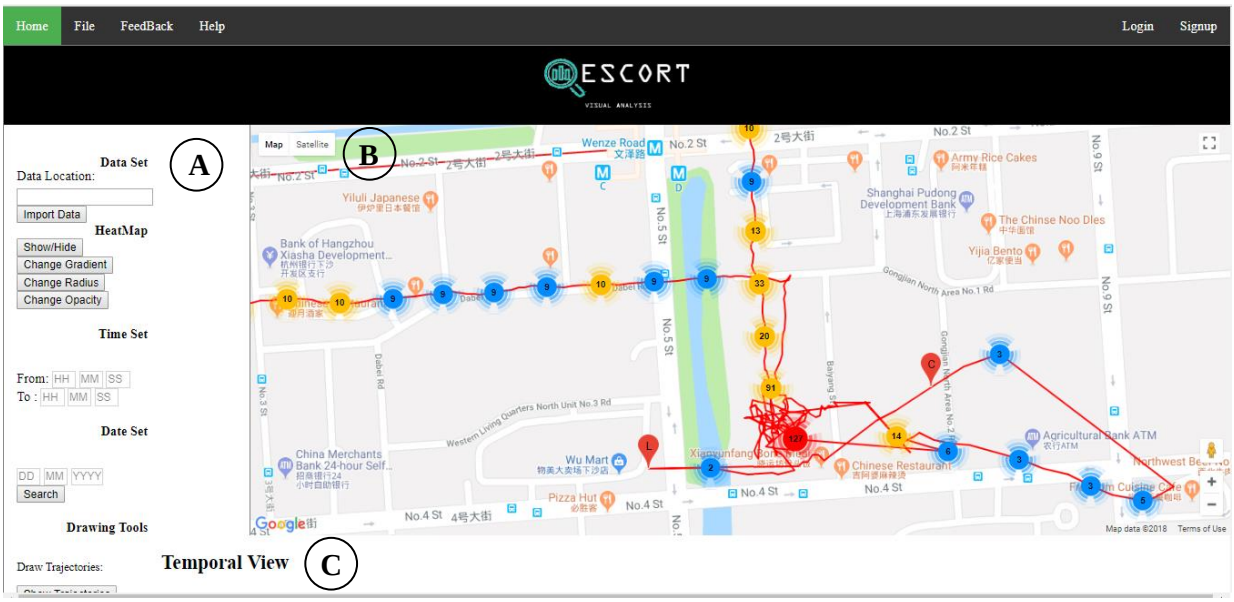

Fig.1 **Escort system (main screen). (A) Dashboard View shows the information of the system to add, edit and delete any of the data. It's include Dataset details to import data, HeatMap settings, Time Intervals to search by specific time interval, Date section to search by specific date, Drawing Section to show or hide the data as points, trajectories and selecting a specific area to search for the best Billboard location for this area, Statistics Options to print and show final statistics. (B) Map View provides a visual summary of the geospatial environment. (C) Temporal View show the time statistics of the entered data and the Statistics of the best billboard location.**

***Abstract*— This document is to present a detailed description of the ESCORT system: a web-enabled system for the analysis of traffic trajectories. It will consist of some web pages accessible with any web browser and traffic anomalies for automated ratings. The features of the system, and describe its interfaces. The document will also explain what the system will do, the constraints under which is must operate, and how the system will react to user input therefore, intended for the stakeholders and developers of the ESCORT system. This document targets the front end users like marketing companies and security solution companies that will use Escort system to satisfy their needs. It will also be beneficial and helpful for data analysts and developers that may work on the Escort system in the future.**

1. *INTRODUCTION:*

This document targets the front end users like marketing companies that will use Escort system to satisfy their needs. It will also be beneficial and helpful for data analysts and developers that may work on the Escort system in the future. Also provides an overview of the systemthat the system some functions is working to show the clustering and visualizing of trajectory. Understanding traffic patterns and crowd motion is a challenging task for urban developers and law enforcement agencies. Spatial-temporal data collected through positioning systems and smart phones offer great opportunities for analyzing motion trajectories and identifying patterns. However, a challenge remains due to the large size and dimensionality of the collected data. This project aims to develop a web-based tool for the visualization and analysis of motion trajectories and the speed of the car in a specific time. Specifically, the idea of the project is to study the traffic patterns in areas, Detected patterns along with raw data will be fed to the visualization system to support high level inference.

2. *RELATED WORK*

This section discusses the prior studies closely related to our work.

1. *SmartAdP*

SmartAdP: Is a web-based application developed under the full-stack framework of MEAN.js (i.e., MongoDB, Express JS, AngularJS, and Node.js). The visual analysis module is implemented using D3.js and Leaflet.js. We deployed the back-end part into our server with 2.40GHz Intel Xeon E5-2620 CPU and 64GB memory. Fig. 2 shows the SmartAdP system architecture. The solution generator helps users formulate a candidate solution. Users need to select several target areas initially, and then two types of Heat Maps are provided to help users determine the befitting solution areas to place billboards. When the solution areas are determined, users set the parameters of model and obtain a recommended solution from the location optimizer. Meanwhile, users can assess whether the selected locations or the generated solutions are good enough and make adjustments accordingly. To further explore and compare multiple solutions, users can switch to solution explorer that comprises three sub-views.

2. *Semantical Trajectory*

Semantical Trajectory system's approach is to detect Taxis traveling over cities detect massive trajectory datasets. They record the data of taxi by detecting samples of the data of GPS location (longitude and latitude) in the interval of a few second in a given specific time to be like that (Car ID, speed, time, occupancy status, direction, and possibly other attributes) for one taxi of each trip which would be hired by the passengers.

3. Urban Pulse

Urban Pulse Is system capturing the Rhythm of Cities: The goal of this to understand the city in the context of the different data sets. They put forth the idea that a city "is involved in the vital processes of the people who compose it, and is a product of nature and particularly of human nature". They suggest an identification between the process occurring within a city and the heart beat or pulsation of a human body.

4. Taxi Cluster

A Visualization Platform on Clustering Algorithms for Taxi Trajectories Taxi is usually considered as the probe of roads in a city. A large amount of taxi GPS mobility data is able to reflect the human mobility and city traffic. The data is described in spatial and temporal form, from which more information can be mined. One kind of the information is related to the basic statistics of the taxi, such as the taxi id, average/min/max speed, travel distance, load or not etc. Other information such as the taxi's trajectory or regions of interest in the city can also be obtained. The analysis methods for taxi GPS data can be generally classified into two types, which are trajectory-based and GPS point-based. The former one is a challenge task due to the calculation of the trajectory similarities. Taxi's travel data is full of information on a city. they introduce a platform so called Taxi Cluster to discuss the clustering algorithms for both the taxi's trajectories and taxi GPS points. The implementation of a trajectory-clustering algorithm that they used, three different clustering algorithms for GPS data points are compared and analyzed. Taxi Cluster platform combines both the analysis capability and the visualization function.

3. *BACKGROUND:*

This section introduces the background on billboard advertising and the types of data used in the system. Thereafter, the analytical tasks are discussed below.

Background Knowledge:

Billboard location selection is a multidisciplinary research problem that involves advertising, communication, and urban computing. In the past year, we have been working with three experts in these fields. In particular, one expert is a manager from an advertising agency who has considerable experience in advertisement planning (1), another one is a postdoctoral researcher in communication (2), and the third is a senior researcher in urban computing (3).

(2) and (3) were invited specially to solve the billboard location selection problem that was originally proposed by EA. As with the real estate business, billboard locations are considered a decisive factor for a billboard campaign. However, different people may have different opinions on locations. We run structured interviews with EA for several rounds and summarize the main challenges for billboard location selection as follows.

Finding befitting areas to place billboards:

The first step is to determine several areas to place billboards based on customers' requirements. Areas frequently visited by the target audience are desired. However, this type of information is difficult to gather. Thus, planners from outdoor advertising companies often make recommendations based on their own experience and knowledge.

4. SYSTEM ARCHITECTURE

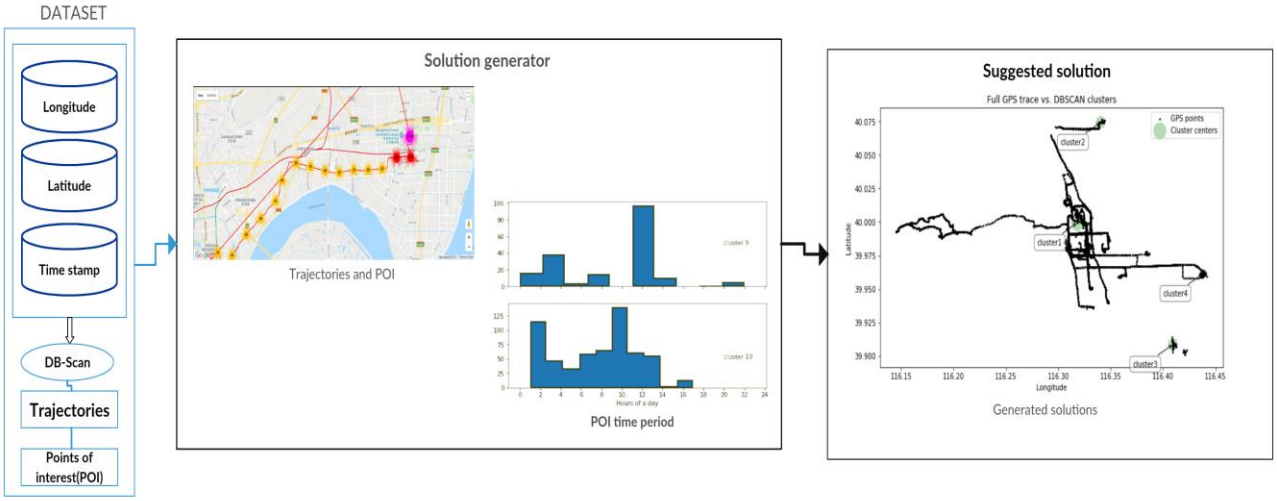

Fig. (2) System Architecture, describing the Dataset, Solution generator and suggested solution.

- **View:**

1-Customer interface: contains a full map view, dashboard controls and temporal view which contains system's statistics as shown in Fig(1) .

- **Controller:**

This is a class that match between view and models. Controller take the input from the view and send the data to database. The coming input of the data is getting from GPS trajectories and see the request of location details, sending these raw data to DB-scan clustering algorithm to develop trajectories and solution clusters.

- **Model:**

1) **DB-SCAN:** *The DBSCAN algorithm can identify clusters in large spatial data sets by looking at the local density of database elements, using only one input parameter. Furthermore, the user gets a suggestion on which parameter value that would be suitable. Therefore, minimal knowledge of the domain is required. The DBSCAN can also determine what information should be classified as noise or outliers. In spite of this, its working process is quick and scales very well with the size of the database – almost linearly. By using the density distribution of nodes in the database, DBSCAN can categorize these nodes into separate clusters that define the different classes. DBSCAN can find clusters of arbitrary shape.*

The Idea Behind DBSCAN

The idea behind DBSCAN can be explained with the help of it's two parameters epsilon and min_points being used in the algorithm.

Pseudo code:

```
DBSCAN(D, epsilon, min_points):
    C = o
    for each unvisited point P in dataset
        mark P as visited
        sphere_points = regionQuery(P, epsilon)
        if sizeof(sphere_points) < min_points
            ignore P
        else
            C = next cluster
            expandCluster(P, sphere_points, C
              , epsilon, min_points)

expandCluster(P, sphere_points, C
        , epsilon, min_points):
    add P to cluster C
    for each point P' in sphere_points
        if P' is not visited
            mark P' as visited
            sphere_points' = regionQuery(P', epsilon)
            if sizeof(sphere_points') >= min_points
                sphere_points = sphere_points joined
                    with sphere_points'
        if P' is not yet member of any cluster
            add P' to cluster C

regionQuery(P, epsilon):
    return all points within the n-dimensional
            sphere centered at P with radius
            epsilon (including P)
```

Fig. (3) DBscan Pseudo code.

Things worth mentioning:

- ➔ *DBSCAN is a flexible algorithm, in the sense that it is dynamic with respect to the data.*
- ➔ *The parameters needed to run the algorithm can be obtained from the data itself, using adaptive DBSCAN.*
- ➔ *It gives a more intuitive clustering, since it is density based and leaves out points that belong nowhere.*
- ➔ *It is very fast compared to traditional clustering techniques like K – Means Clustering since it has complexity O(n^2), n being the number of data points.*

2) *K means clustering:* *K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.*

In the clustering problem, we are given a training set x(1),...,x(m)x(1),...,x(m), and want to group the data into a few cohesive "clusters." Here, we are given feature vectors for each data point x(i)∈Rnx(i)∈Rn as usual; but no labels y(i)y(i) (making this an unsupervised learning problem).

Our goal is to predict k centroids and a label c(i) c(i) for each data point. The k-means clustering algorithm is as follows:

Fig (4) K-Means Math.

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set
   $$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

   }

Expectation Maximization

K-Means is really just the EM (Expectation Maximization) algorithm applied to a particular naive bayes model.

To demonstrate this remarkable claim, consider the classic naive bayes model with a class variable which can take on discrete values (with domain size kk) and a set of feature variables, each of which can take on a continuous value (see figure 2). The conditional probability distributions for P(fi=x|C=c)P(fi=x|C=c) is going to be slightly different than usual. Instead of storing this conditional probability as a table, we are going to store it as a single normal (gaussian) distribution, with it's own mean and a standard deviation of 1. Specifically, this means that:

P(fi=x|C=c)∼N(μc,i,1)P(fi=x|C=c)∼N(μc,i,1)

Learning the values of μc,iμc,i given a dataset with assigned values to the features but not the class variables is the provably identical to running k-means on that dataset.

Pseudo code:

```
1:   Let n be the number of clusters you want
2:   Let S be the set of feature vectors
       (|S| is the size of the set)
3:   Let A be the set of associated clusters f
       or each feature vector
4:   Let sim(x,y) be the similarity function
5:   Let c[n] be the vectors for our clusters
6:   Init:
7:     Let S  ' = S
8:     //choose n random vectors to start our clusters
9:     for i=1 to n
10:      j = rand(|S'|)
11:      c[n] = S'[j]
12:      S' = S' - {c[n]}
         //remove that vector from S'
         //so we can't choose it again
13:    end    //assign initial clusters
14:    for i=1 to |S|
15:      A[i] = argmax(j = 1 to n) { sim(S[i], c[j]) }
16:    end
17:  Run:
18:    Let change = true
19:    while change
20:      change = false     //assume there is no change
                    //reassign feature vectors to clusters
21:      for i = 1 to |S|
22:        a = argmax(j = 1 to n) { sim(S[i], c[j]) }
23:        if a != A[i]
24:          A[i] = a
25           change = true    //a vector changed affiliations
     //so we need to recompute our cluster vectors
     //and run again
26:        end
27:      end
       //recalculate cluster locations if a change occurred
28:      if change
29:      for i = 1 to n
30:        mean, count = 0
31:        for j = 1 to |S|
32:          if A[j] == i
33:            mean = mean + S[j]
34:            count = count + 1
35:          end
36:        end
37:        c[i] = mean/count
38:      end
39:    end
```

Fig (5) K-Means Pseudo code.

5. RESULTS AND DISCUSSION

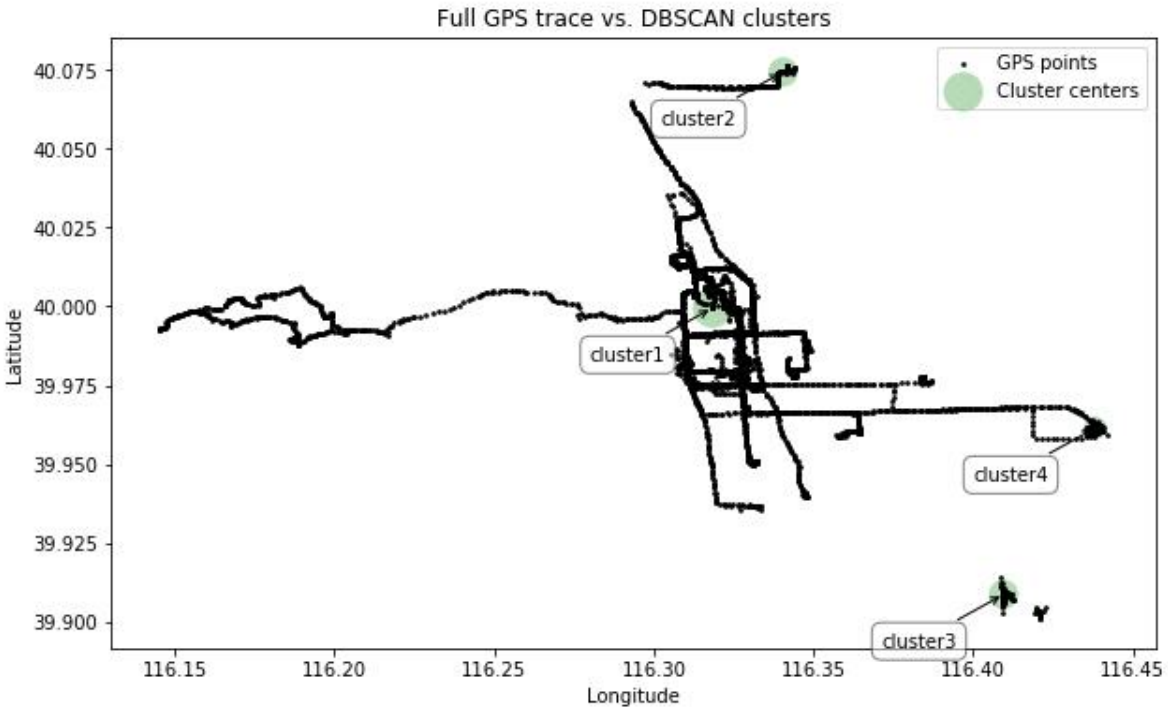

Fig (6): Suggest ad solutions, Trajectory clusters (green) and GPS trace points (Black).

1. Data Description

Using GPS trajectory dataset that collected in (Microsoft Research Asia) Geolife project by 182 users in a period of over five years (from April 2007 to August 2012). A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude and altitude. This dataset contains 17,621 trajectories with a total distance of 1,292,951kilometers and a total duration of 50,176 hours. The trajectories were recorded by different GPS loggers and GPS-phones, and have a variety of sampling rates. 91.5 percent of the trajectories are logged in a dense representation, e.g. every 1~5 seconds or every 5~10 meters per point.

This dataset recoded a broad range of users' outdoor movements, including not only life routines like go home and go to work but also some entertainments and sports activities, such as shopping, sightseeing, dining, hiking, and cycling. This trajectory dataset can be used in many research fields, such as mobility pattern mining, user activity recognition, location-based social networks, location privacy, and location recommendation.

2. Data Set Details

The distributions of distance and duration of the trajectories are presented in Figure 7 and Figure 8.

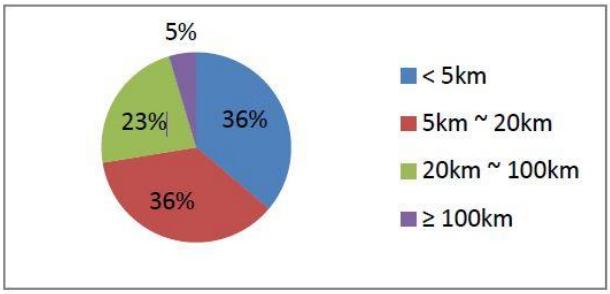

Fig (7) Distribution of trajectories by distance

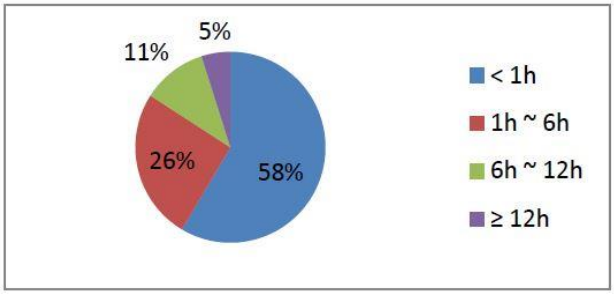

Fig (8) Distribution of trajectories by effective duration

In the data collection program, a portion of users have carried a GPS logger for years, while some of the others only have a trajectory dataset of a few weeks. This distribution is presented in Figure 8, and the distribution of the number of trajectories collected by each user is shown in Figure 9 and Figure 10.

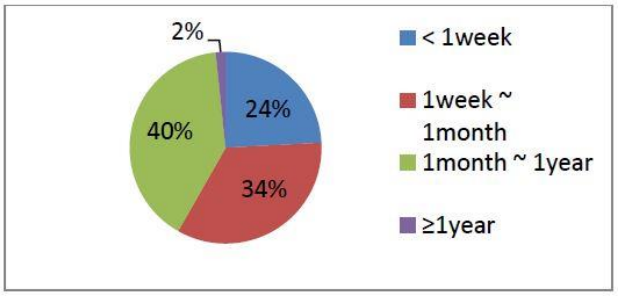

Fig (9) Distribution of users by data collection period

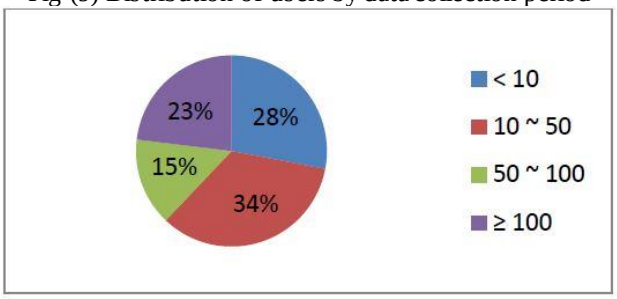

Fig (10) Distribution of users by trajectories

Algorithms results shown in the Figure (11) and Figure (12):

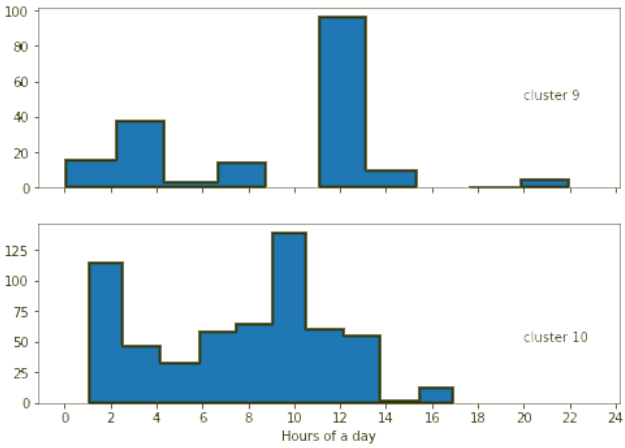

Fig (11): Data statistics showing number of trajectory on a time scale 24 hour (DBscan Algorithm).

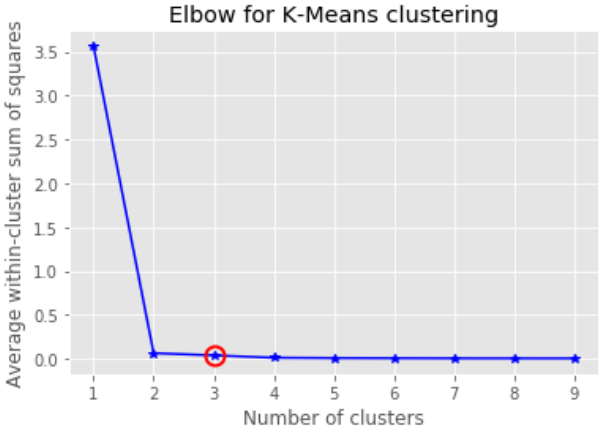

Fig (13): Data statistics showing number of Clusters on a scale 9 clusters and average within cluster sum of squares (K-Means Algorithm).

6. CONCLUSION AND FUTURE WORK

In this paper, we worked on studying the problem of identifying the optimal billboard locations using massive amount of Geo-location dataset. By using trajectories generated by DBscan algorithm, classifying them into clusters with specific features, using those clusters to identify the optimal solution for placing billboard advertisements Closely working with some major stakeholders in this business enabled us to derive two major challenges facing automated billboard advertising, creating and comparing multiple solutions in an immediate and accurate perspective. Hence, we present to the market ESCORT system, an interactive visual analytics system that develops multiple solution that satisfies user's need. We conduct research papers and expert interviews to develop the system the system. Also we have a plan for future work by developing ESCORT mobile application that help clients to choose the optimal locations for rental stores.

Fig (14): Escort Logo

REFERENCES

[1] *Chen, C., Zhang, D., Castro, P.S., Li, N., Sun, L. and Li, S.,2011, December. Real-time detection of anomalous taxi trajectories from GPS traces. In International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (pp. 63-74). Springer, Berlin, Heidelberg*

[2] *Liao, Zicheng, Yizhou Yu, and Baoquan Chen. "Anomaly detection in GPS data based on visual analytics." In Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on, pp. 51-58. IEEE, 2010.*

[3] *Ma, Jun, Chaoli Wang, and Ching-Kuang Shene. "FlowGraph:A compound hierarchical graph for flow field exploration."Visualization Symposium (PacificVis), 2013 IEEE Pacific. IEEE, 2013.*

[4] *Liu Z, Wang Y, Dontcheva M, Hoffman M, Walker S, Wilson A. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. IEEE Transactions on Visualization and Computer Graphics. 2017 Jan;23(1):321-30.*

[5] *Pahins CA, Stephens SA, Scheidegger C, Comba JL. Hashedcubes: Simple, low memory, real-time visual exploration of big data. IEEE transactions on visualization and computer graphics. 2017 Jan;23(1):671-80.*

[6] *Ferreira N, Poco J, Vo HT, Freire J, Silva CT. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. IEEE Transactions on Visualization and Computer Graphics. 2013 Dec;19(12):2149-58.*

[7] *Ferreira N, Poco J, Vo HT, Freire J, Silva CT. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. IEEE Transactions on Visualization and Computer Graphics. 2013 Dec;19(12):2149-58.*

[8] *Ferreira, Nivan, Jorge Poco, Huy T. Vo, Juliana Freire, and Cláudio T. Silva. "Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips." IEEE Transactions on Visualization and Computer Graphics 19, no.12 (2013): 2149-2158.*

[9] *Correll M, Heer J. Surprise! Bayesian Weighting for De-Biasing Thematic Maps. IEEE transactions on visualization and computer graphics. 2017 Jan;23(1):651-60.*

[10] *Laxhammar, R. (2011). Anomaly detection in trajectory data for surveillance applications (Doctoral dissertation, Örebro universitet).*

[11] *Monteiro, Gonçalo, et al. "A framework for wrong way driver detection using optical flow." International Conference Image Analysis and Recognition. Springer, Berlin, Heidelberg, 2007.*

[12] *Schauer, Lorenz, and Martin Werner. "Clustering of Inertial Indoor Positioning Data." 1st GI Expert Talk on Localiza on (2015): 21.*

[13] *Lee JG, Han J, Whang KY. Trajectory clustering: a partition- and-group framework. InProceedings of the 2007 ACM SIGMOD international conference on Management of data 2007 Jun 11 (pp. 593-604). ACM.*

[14] *Ma J, Wang C, Shene CK. FlowGraph: A compound hierarchical graph for flow field exploration. InVisualization Symposium (PacificVis), 2013 IEEE Pacific 2013 Feb 27 (pp. 233-240). IEEE.*

[15] *Lee, Jae-Gil, Jiawei Han, and Kyu-Young Whang. "Trajectory clustering: a partition-and- group framework." In Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pp. 593-604. ACM, 2007.*

[16] *Jagadeesh GR, Srikanthan T, Zhang XD. A map matching method for GPS based real-time vehicle location. The Journal of Navigation. 2004 Sep;57(3):429-40.*

[17] *Gotsman R, ???? ??????. Generating Map-based Routes from GPS Trajectories and their Compact Representation. Technion-Israel Institute of Technology, Faculty of Computer Science; 2013 Jun.*

[18] *UM, Jung-Ho, et al. k-Nearest neighbor query processing algorithm for cloaking regions towards user privacy protection in location-based services. Journal of Systems Architecture, 2012,58.9:354-371*