# A Behavior Observation Tool (BOT) for Mobile Device Network Connection Logs

Ting Wang
SAP Asia Pte Ltd, 1 Create Way, 14-01
CREATE Building, Singapore 138602
dean.wang@sap.com

## ABSTRACT

With the advances of sensory, satellite and mobile communication technologies in recent decades, locational data become widely available. A lot of work has been developed to find useful information from these data, and various approaches have been proposed. In this work, we aim to use one specific type of locational data — network connection logs of mobile devices, which is widely available and easily accessible to telecom companies, to identify and extract active areas of users. This is a challenging topic due to the existence of inaccurate location and fluctuating log time intervals of this kind of data. In order to observe user behavior from this kind of data set, we propose a new algorithm, namely *Behavior Observation Tool (BOT)*, which uses *Convex Hull Algorithm* with *sliding time windows* to model the user's movement, and thus knowledge about the user's lifestyle and habits can extracted from the mobile device network logs.

## Categories and Subject Descriptors

H.2.1 [**DATABASE MANAGEMENT**]: Logical Design—*Data models*

## Keywords

Spatial-Temporal Data Mining, Loctional Data, Convex Hull Algorithm, Network Connection Logs, Active Area

## 1. INTRODUCTION

Over the last few decades, with the increasingly accurate positioning services (*e.g.* GPS, AIS, Mobile Phone Triangulation, RFID/Wi-Fi tracking *etc.*) and the decreasing price of their deployment, locational data becoming pervasive in our daily lives and scientific researches. Either indoor or outdoor, it is not difficult to obtain the trace, the velocity, and even the acceleration of any moving entity (in the case of this paper, a user) of our interest with proper equipments and infrastructure. As part of the "big data regime", interests in locational data has recently grown even more rapidly thanks to the new database technology and data mining techniques. When locational data coupled with time-stamps, it becomes *spatial-temporal* data — with both space (spatial) and time (temporal) information [1]. The timely sequence locations of a user defines its *trajectory*.

Trajectories of users are widely used in a variety of business and public sector applications, such as traffic modeling and supply chain management. More often, they are important sources for discovering users' movement, such as patterns, correlations, and clusters. To add more business value to this kind of studies, researchers are usually interested to find information about the *work locations*, *home locations*, *shopping places*, and *leisure areas* of certain group of people so their lifestyle could be understood. These information could be crucial to business plans and urban design projects.

Among various source of trajectory data, in this paper we focus on one specific type — the network logs of mobile devices. It is easy to access, widely available, but also suffers from inaccuracy of location and uncertainty of timing. More on this kind of data set will be discussed in Sect. 2.

To deal with the drawbacks of mobile device network logs, we propose a new algorithm in this paper, namely *Behavior Observation Tool (BOT)*. It represents the user's trajectory as a series of polygons, instead of line segments. The introduction to MOA will be given in Sect. 3, and we talk about how to use BOT to understand user's behavior in Sect. 4.

A case study of how MOA is applied to a real life dataset is discussed in Sect. 5. Sect. 6 concludes the paper.

## 2. THE DATASET AND CHALLENGES

While pervasive positioning technologies give us opportunities to access vast amount of locational data and test these solutions, they also raise challenges due to the sparse nature of data collection strategies, the diverse density of the data, and technical issues associated with the accuracy of the data. The most common source of locational data of mobile devices are from *Global Positioning System (GPS)* and Wifi signals. However, logging the locations constantly is considered energy consuming. In our experiment, it drains the device's power about 50% faster. Therefore, some alternative approach which is more enery friendly, and "transparent" to the users could be preferred.

Network logs of mobile devices could be a better option. It is the log file of *when* a given mobile device connect to *which* base station. It is widely available because it is a common log file telecommunication service providers need to keep for every user. It also preserves the user's privacy because


*WWW'14 Companion,* April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2579693.

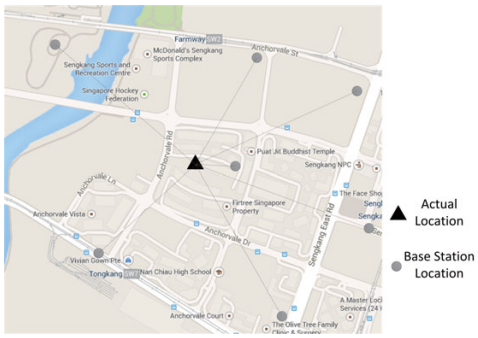


Figure 1: Actual vs. Recorded (Base Station) Locations

the identity of the user could be easily replaced by some hashkeys. However, it also poses some different challenges to the data miners than other datasets:

1. Since the position log is the location of the base stations, it is not precisely the location of the user, and can have error up to several hundreds of meters. Fig. 1 shows an example where a device is staying stationary but its connection log is shown to be all over the place.

2. Each entry is entered to the data set when the device makes connection to the base station. It could be a "keep-alive" beacon, a phone call, a sms, or data connection *etc.*. Thus the timely frequency of a single device could be fluctuation. It is still common for a device to have as low as 1 or 2 entries per hour during night, but several hundreds of entries in one hour in day time.

Meanwhile, most of the existing mobility observation techniques can be classified as one of the following three categories:

- **State Based**: states are defined by (*time*, *location*) combinations. The trajectory of a user is thus a series of states and the transitions among them. Markov-chain and other related models [2] can be used to study the underlying patterns.

- **Similarity Based**: similarity between trajectories can be calculated from the 3-dimensional or 4-dimensional proximity of the data points [3]. It is then usually used to define clusters or places of interest.

- **Density Based**: in large scale problems [6], importance of locations can simply be reflected by the density of data points in that area.

We found they are not suitable in dealing with the mobile device network logs. For example, when the log entries are scarce (either spatially or temporally), the similarity based solution may produce inaccurate results, because data points for similar trajectories may be far apart. On the other hand, for high frequency locational data (*e.g.* logs during day time), the state based approach could be overwhelmed by the enormous number of states. Also, the density based solutions will need the timely frequency of data points to be consistent, otherwise their density would not reflect the true distribution of the moving users. Moreover, we have not seen any existing solution that deals with the error in the location detections, which in fact could be crucial to the correctness of the results. Hence, when we study the behavior observation problem, we look for a solution that has the ability to adapt to errorneous data, and extract as much information as possible.

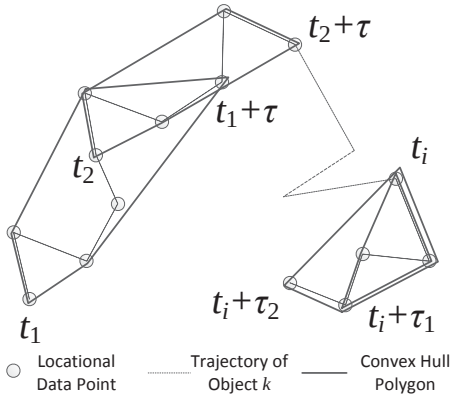


Figure 2: Convex Hull Polygons of User $k$

## 3. BEHAVIOR OBSERVATION TOOL (BOT)

As shown in Fig. 2[1], BOT takes the raw position data of a user[2] with UID $k$ as input. It splits the raw data of $k$'s trajectory into segments by time windows. A time window is determined by two factors — starting time point $t$ and window size $\tau$. Then each segment is processed by the convex hull algorithm and represented as a polygon $\mathcal{P}$. That is to say, each polygon $\mathcal{P}$ corresponds to the movement of user $k$ during a time window (determined by $t$ and $\tau$). The geometric properties, such as centroid location, area size, perimeter, and number of edges/vertices *etc.*, We denote these properties of $\mathcal{P}$ as a function of $k$, $t$ and $\tau$, written as

$$\mathbf{P}(k, t, \tau) = \{\text{geometric properties of } \mathcal{P}\}.$$

We will discuss the data storage structure for the outcome of BOT in Sect. 3.3.

### 3.1 Sliding Time Window

A time window is a time interval $[t, t+\tau)$ in which the trajectory is considered as a *segment*. We need time window in mobility observation because usually the locational data span throughout days or even months. To make sense of the data, smaller time window, with less data, could be more meaningful and simplify the process of analysis.

Moreover, the time window needs to "slide" forward as time evolves. Since trajectory data is *spatial-temporal*, sliding time windows help us to understand its "temporal" property. A single time window only shows static location of the user in the particular time interval, while a series of sliding time windows demonstrate how the user is moving around over time. We denote $i$ continuous sliding time windows as

$$\{[t_1, t_1+\tau), [t_2, t_2+\tau), \ldots, [t_i, t_i+\tau), \ldots\}.$$

[1]To show the trajectory clearly, the convex hull polygons in this figure is drawn slightly larger than they should be.
[2]Assume each user is identified by an integer UID.

For convenience of discussion, it is usually by default that the amount of the sliding window advances, represented by

$$t_\Delta = t_2 - t_1 = t_k - t_{k-1}$$

is a constant.

To avoid loss of information, we will need $t_\Delta \leq \tau$, otherwise data between two consecutive time windows would not be able to be captured. In particular, we say it's an *overlapping* sliding time window setup if $t_\Delta < \tau$, and a *non-overlapping* setup if $t_\Delta = \tau$. Being overlapping means redundancy in the data, and could result in larger data size and more processing time, but it is also a smoothing technique and able to avoid sudden "jump" between the time windows. In our experiment we found that $t_\Delta \approx 1/3\tau$ usually gives us a smooth transition among the time windows. However, it is only an empirical study, and the result is only applicable to our specific data sets and problems.

The size of the time window $\tau$ becomes a crucial property to define the segments meaningfully. Consider each segment as a snapshot of a large picture — its size should be adequately big to show some information more than only 1 or 2 pixels, yet it should not be too big and contains too much information. It also depends on the type of movement the user tend to make. For example, if we are observing something that moves swift and changes direction frequently (*e.g.* a bird, a soccer player, or a car on empty city streets) we should use small time windows; on the other hand, for slow and constant user (*e.g.* like an elephant, a pedestrian, or a car in heavy traffic) bigger time window could be adequate to show the status and changes of the user.

We may not understand the users' movement before we start observation. It could also be the case when the speed and mode of movement changes, like the car in the previous examples. Therefore we propose having a variable window size in BOT, $\tau$, to be set to different values in BOT to capture user's movement in different scales. More importantly, it gives us the freedom to tune the "detail level" of our observation — we understand more details with smaller $\tau$, and see the bigger picture with larger values of $\tau$. In Fig. 2, we show how the outcome convex hull polygon changes with two different values of $\tau$, namely $\tau_1$ and $\tau_2$ for starting time point $t_i$.

## 3.2 Convex Hull Algorithm

A *Convex Hull* is of a set of locational points in the Euclidean plane or Euclidean space which is the smallest convex set that contains the set [4]. The problem of finding convex hulls finds its practical applications in pattern recognition, image processing, statistics, GIS and static code analysis by abstract interpretation. In particular, convex hull algorithm has been used to study home range of wild-life animals [9]. BOT extend these work further to the human mobility scenario with the relationship of their social behaviour.

In computational geometry, numerous algorithms are proposed for computing the convex hull of a finite set of points, with various computational complexities. The complexity of the corresponding algorithms is usually estimated in terms of $n$, the number of input points. The Graham scan algorithm [5] for convex hulls consists of a single sorting step followed by a linear amount of additional work, and is of complexity $O(n \log n)$ time.

Using sliding window and convex hull algorithm, the trajectory is transformed to a series of polygons, each corresponds to the user's movement in a particular time window. The relationship between the geometric properties (denoted as $\mathbf{P}$) of a polygon and the user's movement will be discussed in Sect. 4. Before that, we first discuss how $\mathbf{P}$ can be stored and analyzed efficiently.

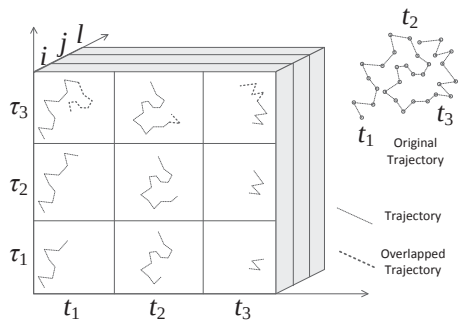


Figure 3: Use Data Cube to Store $\mathbf{P}$

## 3.3 Outcome Data Cube

In computer programming contexts, a *data cube* is a three (or higher) dimensional array of values, commonly used to describe a time series of data. It is a common data structure for *online analytical processing (OLAP)*, which is a computer-based technique for analyzing business data in the search for business intelligence.

A data cube can be considered a generalization of a high dimensional spreadsheet formed by *cells*. A cell corresponds to one particular value (in this case $\{k, t, \tau\}$) in each dimension. In conventional data cube, each cell of the cube holds a number that represents some measure of the business, such as sales, profits, expenses, budget and forecast. For example, a company might wish to summarize financial data by product, by time-period, and by city to compare actual and budget expenses. Product, time, city and scenario (actual and budget) are the data's dimensions.

Data cube is an ideal structure to store the outcome of BOT. The properties of the polygons, $\mathbf{P}$, can be summarized to user ID $k$, time window starting point $t$ and window size $\tau$, as we have discussed in the beginning of this section and shown in Fig. 3. $\{k, t, \tau\}$ thus form the three dimensions of the cube, and the polygons generated by BOT for each corresponding user in each corresponding time window. The only difference is, we may need to store more than one properties in each cell: such as area size, perimeter length, centroid position *etc.*. However, we are not putting the "properties" as a new (forth) dimension, because it is highly dependent on the actual scenario and application. Different applications requires different set of polygon geometric properties to be studied. In some applications, even only one property could be sufficient to give us the desired information. Therefore, to keep our discussion general, we only consider the outcome data of BOT as a three dimensional data cube.

# 4. BEHAVIOR OBSERVATION

The objective of BOT is to observe users' movement and extract useful behavioral information. In this section, we demonstrate how such information can be extracted from the polygons derived from convex hull algorithm, and how their geometric properties can be used to derive the behavior of

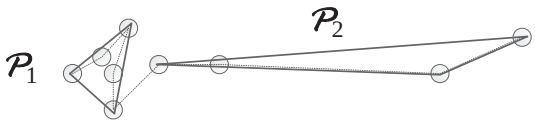


Figure 4: Examples of Active Area Detection

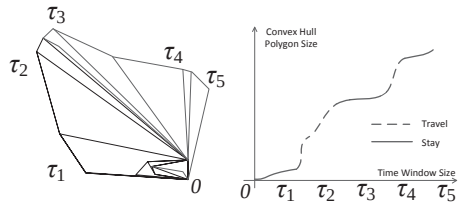


Figure 5: Polygons Sizes with Different Values of $\tau$

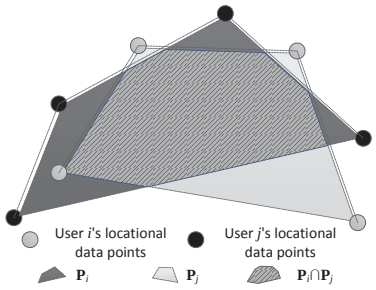


Figure 6: Similarity Between Polygons $\mathcal{P}_i$ and $\mathcal{P}_j$

the users. Four useful aspects are discussed, namely *active area*, *traveling pattern*, *similarity*, and *randomness*, of the users.

## 4.1 Active Area

One of the most interesting topic in movement observation is to understand the *active area* of the user, *i.e.* where the user stop and do something, such as working, shopping, watching a movie, or sleeping.

When the trajectory is represented as polygons in BOT, the area size of the a polygon is the area that the user has covered in the corresponding time window. When all the time windows have the unit size — *i.e.* consider only $\{t, k\}$ dimensions for a constant $\tau$ in the data cube generated by BOT — smaller polygons indicate the fact that the user spend the same amount of time within a smaller area. This could be a good indication of active area — same time window length, but less movement, as shown in Fig. 4 by $\mathcal{P}_1$. We note that this has nothing to do with the density of the signal, because we do not consider how many records are found within the polygon, but only interested in the boundary and size of it. In this way, active area in any shape can be found.

There can be extreme cases like shown in Fig. 4 by $\mathcal{P}_2$, where area size could be small even if the location records are far apart. To rule out this kind of exception, a secondary polygon property can be considered: such as number of edges, polygon perimeter, and edge length variance or deviation. If the polygon has few edges with long perimeter and large length deviation, it means the polygon's shape is similar to $\mathcal{P}_2$, and thus cannot be identified as an active area.

Another special case is when the polygon size is 0. It means only one or two locational records are found in the time window. We can not conclude the active area in this case. In this case, we can extend the time window size so that more data points show up in the time window and better conclusion could be drawn.

## 4.2 Travel Patterns

How the user from one active area to another, *i.e.* the *travel pattern*, is also often of great interests in mobility observation studies. There could be two scenarios where the polygon in a particular time window could indicate the user is traveling.

Firstly, as demonstrate in the previous section, polygons with small size, but long perimeter and large edge length variation is a result of traveling user. The long edge in this kind of polygon shows the fact that the user could be traveling during the time window, and the short edges actually gives us info about the destination and starting point of the traveling. Usually in this scenario, there are few data points on the edge to show the route of traveling, and more information such as transportation means and speed is hard to be determined.

Secondly, convex hull polygons with large size could also be a good indication that the user is traveling. We can understand it as a large active area, in which the user goes to multiple places. This is a good example when the dimension $\tau$ of the data cube could be useful. With variable time window size, we are able to find out when and where the user starts traveling. Fig. 5 shows how the polygon size evolves with the value of window size $\tau$, for fixed starting point $t = 0$. Those steep slopes indicates "traveling" while the flat parts refer to "staying".

## 4.3 Similarity

Moving users can usually be clustered by their *trajectory similarity*, which is another interesting field of study in mobility observation. In existing works, it is measured by the closeness of the locational data points. Again because of the signal quality, in particular the time of taking the records, the solution could be less effective than it sounds.

For example in Fig. 6, user $i$ and $j$ move on their corresponding routes, which are close to each other. However, due to the difference in timing, the locational data points are not close, and thus the existing solutions may not be able to recognize them as similar trajectories. BOT converts their routes to polygons, and similarity can be estimated by the overlapping area size of the two polygons constructed by $i$ and $j$'s trajectories, respectively. As shown in Fig. 6, the high percentage of overlapping indicates the two trajectories are similar to each other. Quantitatively, a *similarity index* $\mathbf{S}$ can be measured as the size of overlapping area divided by the total area covered by the two polygons, written as

$$\mathbf{S} = \frac{\mathbf{A}_{\mathcal{P}_i \cap \mathcal{P}_j}}{\mathbf{A}_{\mathcal{P}_i \cup \mathcal{P}_j}}$$

where $\mathbf{A}$ denotes the area size of the polygon. A value of $\mathbf{S}$ close to 1 will indicate the given polygons are similar to

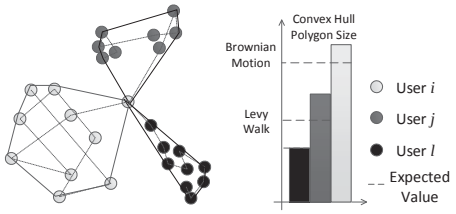


Figure 7: Randomness Reflected by Polygon Sizes

each other. We note the measure of similarity index can be easily extend to multiple users by comparing polygon sizes of same $\tau$ and $t$ across different ID's in the $k$ dimension of the data cube.

The accuracy and validity of similarity could be improved if we put dimension $t$ into consideration, too. As what we have done for travel pattern analysis in the previous section, seeing how the polygon evolve over time gives us a better understanding to the trajectories. If two users have polygons with high similarity index over several consecutive time windows, it could be more evident that these two users travels in similar pattern.

## 4.4 Randomness

Objects may or may not move randomly. Non-random movement means the user has certain purpose which may be reflected by the mobility pattern. Finding out the *randomness* of a user could lead to useful use cases such as suspect behavior detection and intention detection. As far as we came across in our research, we haven't see any existing trajectory data mining techniques can measure the randomness of the users.

Thanks to many previous sound works such as [7] on the property of convex hull, we have the expect size of the convex hull formed by the trace when the user is moving in certain mobility model, such as Brownian motion, random walk or Levi walk *etc.* [8]. For each model, the expected size of convex hull will be a function of time and speed. That is to say, if we can estimate the speed of the user[3], we can calculate a theoretic value for the convex hull size assuming the user is moving in certain mobility model. This theoretic value serves as a benchmark. By comparing the real convex hull size with this benchmark, we can evaluate how close the user's moving pattern is to the model in our assumption. Usually purposeful movement will result in smaller convex hull size. In Fig. 7, we show how th three users movement could be classified based on their randomness: user $i$ being the most random, as its convex size is close to Brownian motion benchmark; user $j$ and $l$ show certain degree of purposeful movement, with $l$ more purposeful than $j$.

# 5. CASE STUDY

BOT is tested with multiple real life network logs of mobile devices. In this paper, we present our results from a 7-day network connection log of 1500 mobile devices in a city. This data set is available for many researchers, but few of them could really make sense out of it.

[3]It is usually not that difficult: 5km/h for pedestrian, 50km/h for vehicles in city streets *etc.*.

We set up BOT as $\tau = 3$hr and $t_\Delta = 1$hr. After filtering out infrequent users, we take 1028 devices as input.

We firstly found the active areas of the devices. We use a threshold of 300m$^2$ to define active areas. Moreover, we find the repeated active area during the 7 days period as *regular areas*, which may indicate places people who carry the mobile device frequently visits and do something. Typically, they will be the home location or work location of the user of the mobile device. The results are plotted in Fig. 8, where three representative users are plotted and their regular areas are marked with red color.

- User $i$ has two frequent locations as shown in Fig. 8a. From dimension $t$ (not shown in the figure), we found that the user goes to one of these two places at night, and visits the other during day time (work hour). We may derive that these two places being his home, and work place, respectively.
- User $j$ has multiple frequent locations — one being his home and he/she visits multiple places during work hour as plotted in Fig. 8b. This could be a result of his/her work — goes multiple places to visit customer.
- User $l$ has only one frequent location — home, as depicted in Fig. 8c. His/her locational results are all over the city and does not show any other regular active area. One possible job of this user is a taxi driver, who go around the city and only returns home at night.

Another even more interesting finding is that we cluster the users based on how the size of their convex hull polygon change over time. During this 7 day period, 166 time windows are formed and 166 corresponding polygons are found by BOT. We use k-means algorithm on these 166-dimensional data to cluster the users to 5 groups. We plot the mean size of these polygons against the starting point of the time window in Fig. 9.

We understand that large polygon size means the user is traveling. Therefore we can clearly observe that some users have peaks in morning and evening rush hour, when they are going to work and going home, as pointed out by "A" in the figure. On the other hand, when the users stop moving and stay put, their polygon size reduce. We can thus see how the users stay at work or have lunch break in the middle of the day, as pointed out by "B". We can also see from the $t$ dimension ($x$ axis of the figure) that different user can be active during different time of the day, some in the day time and some at night, as pointed out by "C".

In particular, six clusters can be identified:

- Regular work far away from home
- Regular work close to home
- Almost stationary (home workers, *etc.*)
- All-day travelers (sales persons, *etc.*)
- All-day travelers with lunch break (drivers, *etc.*)
- Long-distance night-travelers (taxis, *etc.*)

This result is significant — we classified the users based on their behavior using BOT, despite of the inaccuracy and inconsistency of the source locational data. Moreover, this solution is extremely outstanding in one feature: we do not


(a) User $i$ (b) User $j$ (c) User $l$

Figure 8: Frequent Location of Users

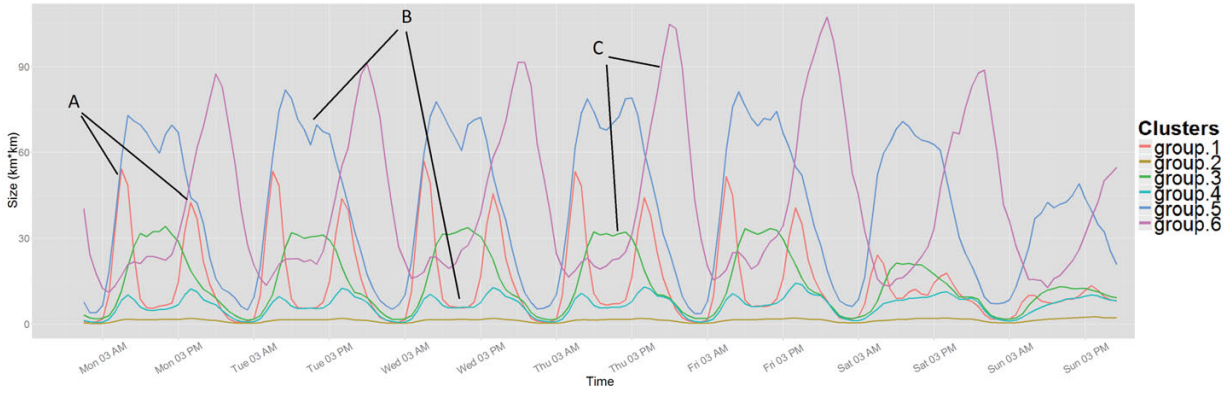


Figure 9: Clusters based on Polygon Area Size over Time

need to know the exact location of the user to study his/her lifestyle. The polygon size is irrelevant to the actual location. In this way, users' privacy could be preserved and confidential information would not be leaked in the study.

## 6. CONCLUSION

In this work, we have proposed a new scheme to study locational data, namely *Behavior Observation Tool (BOT)*. It uses two techniques: sliding time window with variable size, and convex hull algorithm to convert users' trajectories to a series of polygons, stored in a data cube structure. We have shown that trajectory properties can be extracted from the geometric properties of the polygons, and the behavioral patterns of the users can thus be observed. We found it in particular works well with mobile device network log data, which is a widely available dataset but is errorneous in space and inconsistent in time.

## 7. ACKNOWLEDGEMENT

This work was supported in part by the Singapore Economic Development Board (EDB) and National Research Foundation (NRF).

## 8. REFERENCES

[1] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data*. Springer Berlin, Germany, 2006.

[2] A. Asahara, K. Maruyama, A. Sato, and K. Seto. Pedestrian-Movement Prediction based on Mixed Markov-Chain Model. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 25–33. ACM, 2011.

[3] L. Chen, M. T. Özsu, and V. Oria. Robust and Fast Similarity Search for Moving Object Trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2005.

[4] M. De Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry*. Springer, 2008.

[5] R. L. Graham. An Efficient Algorith for Determining the Convex Hull of a Finite Planar Set. *Information Pocessing Letters*, 1(4):132–133, 1972.

[6] S. Isaacman, R. Becker, R. Cáceres, M. Martonosi, J. Rowland, A. Varshavsky, and W. Willinger. Human Mobility Modeling at Metropolitan Scales. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 239–252. ACM, 2012.

[7] S. N. Majumdar, A. Comtet, and J. Randon-Furling. Random Convex Hulls and Extreme Value Statistics. *Journal of Statistical Physics*, 138(6):955–1009, 2010.

[8] R. N. Mantegna and H. E. Stanley. Stochastic Process with Ultraslow Convergence to a Gaussian: the Truncated Lévy Flight. *Physical Review Letters*, 73(22):2946, 1994.

[9] B. J. Worton. A Convex Hull-Based Estimator of Home-Range Size. *Biometrics*, pages 1206–1215, 1995.