A Data Cube for Mobility Observation (D3MO)

Ting Wang, *Member IAENG*

***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 has been proposed. *Mobility Observation* is the study of mobile objects' trajectory and movement patterns, with the goal of deriving insights from their locational records. In this paper, we introduce a method that use data cube to store trajectory data and use it for data mining, namely *Data Cube (D^3) for Mobility Observation (D3MO)* (pronounced /demō/), and have made some interesting and useful discoveries. In particular, we found D3MO, when used together with the *convex hull* algorithm, is able to represent the spatial-temporal data and enable its hidden property to be discovered. Moreover, with a case study, we show how D3MO is used to study peoples' lifestyle patterns.**

***Index Terms*—Spatial-Temporal Data Mining, Data Cube, Modeling, Data Structure**

I. 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 (referred to as an *object* in this paper) of our interest with proper equipments and infrastructure. Massive data have been collected in various research projects since early 90's [1]. 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 [2]. The timely sequence locations of an object defines its *trajectory*.

Trajectories of objects are widely used in a variety of business and public sector applications, such as traffic modeling and supply chain management [3]. More often, they are important sources for *Mobility Observation* — a process in which information related to objects' movement, such as patterns, correlations, and clusters, could be discovered [4].

In this paper, we propose a new scheme, namely *Data Cube (D^3) for Mobility Observation (D3MO)* (pronounced /demō/). Using sliding time windows with variable size, D3MO constructs a cubical data structure [5] to represent objects trajectory. The major difference between D3MO and existing works is that we represent the object's trajectory in a multi-dimensional way, instead of points and line segments. Moreover, we use *convex hull* algorithm to convert the trajectories to a series of polygons. The mobility patterns can thus be observed from the geometric properties (*e.g.* location, size, shape, and number of vertices/edges *etc.*) of these polygons.

Many other terms have been used for the study of trajectory and locational data — *Mobility Characterization/Modeling* [4], *Trajectory Mining* [6], *Movement Prediction/Inferring* [7] and *Mobility Patterns/Profile Discovery* [8] *etc.*. We use *Mobility Observation* in this paper, because it is a more general term, in the sense that clustering, pattern recognition, prediction, and activity detection are all kinds of observations, which can be done with D3MO. D3MO is not only a solution to one single specific problem, but also a general method to treat and represent locational data, to discover information and extract knowledge — regardless the quality and density of the source data. For example, we will demonstrate how peoples' lifestyle patterns can be extracted, classified and clustered using D3MO even when the source data is scarce and largely inaccurate.

A brief introduction to the existing works and challenges are discussed in Sect. II, followed by the introduction of the D3MO algorithm in Sect. III. We talk about how to use D3MO with convex hull algorithm to detect active areas of users in Sect. IV, and a case study of how D3MO could be used with locational data to discover people's lifestyles is discussed in Sect. V. Sect. VI concludes the paper with the strength of D3MO and future research directions.

II. Background

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 [9], [10]. The enormous volume of data can easily overwhelm human analysis. This problem is usually solved by some compression algorithm such as [11], while almost all existing work uses {time, location} tuple to store the locational records. This motivates the need for new and more efficient data structure for trajectory mining and other spatial-temporal data mining problems. D3MO is the solution we propose to this problem. It efficiently process and store the locational data in a multi-dimensional data cube, enabling useful information to be extracted.

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 [7]. The trajectory of an object is thus a sequence of states and the transitions among them. Markov-chain and other related models [12] 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 [6]. It is then usually used to define clusters or places of interest.

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

Dr. Wang Ting is with SAP Asia Pte Ltd, Singapore. (E-mail: dean.wang@sap.com)

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- **Density Based**: in large scale problems [13], importance of locations can simply be reflected by the density of data points in that area.

However, these solution all have their weakness when dealing with different data sets. For example, if the data points 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.* continuously generated by positioning sensors), the state based approach could be overwhelmed by the enormous number of states. Compression algorithm such as [11] will be needed to pre-process the data and could result in inefficiency. Also, the density based solutions will need the timely frequency of data points to be normalized, otherwise their density would not reflect the true distribution of the moving objects. 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 mobility observation problem, we look for a solution that has the ability to adopt to different source locational data, and extract as much information as possible.

III. D3MO

D3MO is a type of data structure, which uses sliding time window with variable window size to store trajectory-related data, such as speed, distance traveled or energy consumed. In this work, to make it more interesting, we use the data cube to store the size of the convex hull polygon that constructed from the trajectory. In this section, we first introduce how D3MO and convex hull work and work together.

A. 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 object in the particular time interval, while a series of sliding time windows demonstrate how the object is moving around over time. We denote i continuous sliding time windows as

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

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 $t_\Delta = \tau$. Overlapping means redundancy in the data to be analyzed. It 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 τ 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 object 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 object (*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 object.

We may not understand the objects' 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 D3MO, τ, to be set to different values in D3MO to capture object'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 τ, and see the bigger picture with larger values of τ.

B. 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 [5]. The term *hypercube* is sometimes used, especially for data with more than three dimensions. 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 trajectory. Certain property of the moving object, denoted as $\mathbf{P}$, can be summarized to object ID k, time window starting point t and window size τ, as shown in Fig. 1. $\{k, t, \tau\}$ thus form the three dimensions of the cube, and the polygons generated by D3MO for each corresponding object in each corresponding time window. The only difference is, we may need to store more than one properties in each cell: such as speed, location, acceleration *etc.*. However, we are not putting the "properties" as a new (forth) dimension, because it is highly dependent on the actual scenario and application.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

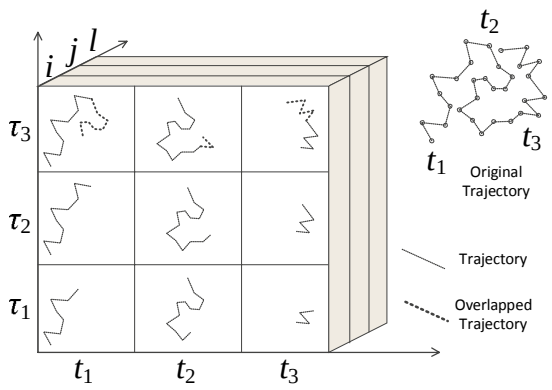

Fig. 1: Use Data Cube to Store $\mathbf{P}$

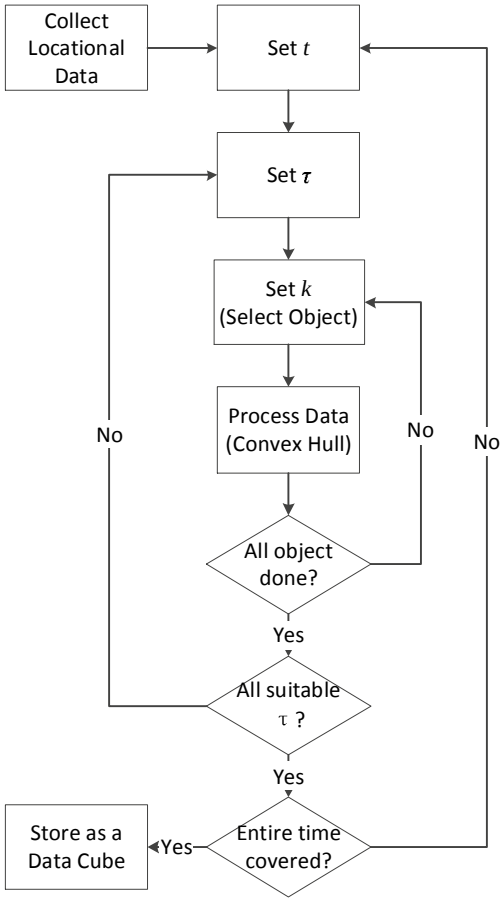

Fig. 2: Flowchart to Construct D3MO

Different applications require different set of properties to be studied. In some applications, even only one property could suffice to give us the desired information. Therefore, to keep our discussion general, we only consider the outcome data of D3MO as a three dimensional data cube, and we will use it to store the sizes of the convex hull polygons. Fig. 2 depicts the data flow in D3MO, and how the data cube is constructed.

IV. D3MO with Convex Hull

The *Convex Hull* algorithm is another technique we used in this work. 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 [14]. 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 [15]. D3MO extend these work further to the human mobility scenario with the relationship of their social behavior. More importantly, we add a sliding window with variable size [16] to the algorithm, so that the timely change of the convex hull could be studied, and mobility patterns can thus be observed.

A. Convex Hull Algorithm

In computational geometry, numerous algorithms are proposed for computing the convex hull of a finite set of points, with various computational complexities. Computing the convex hull means that a non-ambiguous and efficient representation of the required convex shape is constructed. The complexity of the corresponding algorithms is usually estimated in terms of n, the number of input points, and h, the number of points on the convex hull.

The complexity of finding a convex hull as a function the input size n is lower bounded by $\Omega(n \log n)$. However, the complexity of some convex hull algorithms can be characterized in terms of both input size n and the output size h (the number of points in the hull). Such algorithms are called output-sensitive algorithms. They may be asymptotically more efficient than $\Theta(n \log n)$ algorithms in cases when $h = O(n)$.

The lower bound on worst-case running time of output-sensitive convex hull algorithms was established to be $\Omega(n \log n)$ in the planar case [14]. There are several algorithms which attain this optimal time complexity. The earliest one was introduced by Kirkpatrick and Seidel in 1986 (who called it *the ultimate convex hull algorithm*) [17]. A much simpler algorithm was developed by Chan in 1996, and is called Chan's algorithm [18].

A number of algorithms are known for the three-dimensional case, as well as for arbitrary dimensions [19]. For a finite set of points, the convex hull is a convex polyhedron in three dimensions, or in general a convex polytope for any number of dimensions, whose vertices are some of the points in the input set. Its representation is not so simple as in the planar case, however. In higher dimensions, even if the vertices of a convex polytope are known, construction of its faces is a non-trivial task, as is the dual problem of constructing the vertices given the faces. The size of the output may be exponentially larger than the size of the input, and even in cases where the input and output are both of comparable size the known algorithms for high-dimensional convex hulls are not output-sensitive due both to issues with degenerate inputs and with intermediate results of high complexity [20].

Using sliding window and convex hull algorithm, the trajectory is transformed to a series of polygons, each corresponds to the object's movement in a particular time window. In particular, we found that the object's *active area* can be found from convex hull polygon size.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

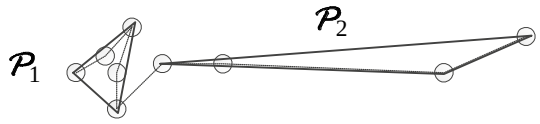

Fig. 3: Examples of Active Area Detection

B. Determining Active Area using D3MO

One of the most interesting topic in movement observation is to understand the *active area* of the object, *i.e.* where the object stop and do something. If the object is human, active area would reflect where she/he lives, works or do shopping (which will be discussed in detail in Sect. V). If the object is a mobile sensor, it could show where is the place that the sensor is trapped, or has more data to process. If the object is an animal, the active area would demonstrate the living area distribution of it, which would be crucial for some zoologists.

In conventional mobility observation algorithms it is usually done from a signal density perspective — places with denser locational records are considered as active areas. But the effectiveness of this kind of solutions hugely depend on the source locational data quality. If the object does not sense its location frequently, there would not be a clear different in signal density between active area and non-active area. Moreover, it could be difficult to find out the boundary of the active area from the signal density. To use these solution, we usually need to pre-define area shapes (such as grids or hexagons) to calculate the density. Thus the exact location of the active area can hardly be determined.

In D3MO with convex hull, the area size of the a polygon is the area that the object 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 τ in the data cube generated by D3MO — smaller polygons indicate the fact that the object 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. 3 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. 3 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.

V. Case Study — Lifestyle Discovery

D3MO is tested with multiple locational data sets, with different positioning techniques. In this paper we discuss one of them: mobile device signal location data set. It's 7-days data of a city's mobile devices. This data set is available for many researchers, but few of them could really make sense out of it, due to the following challenges:

1) The positioning technique for this data set is not GPS nor signal triangle. Each entry is a timestamp with the location of the *base station* that the mobile device is connecting to. The error could be hundreds of meters. Fig. 4 shows an example where a device is staying stationary but its connection 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 quite low. In pre-processing, we have filtered out some extremely infrequent devices, but it is still common for a device to have as low as 1 or 2 entries per hour.

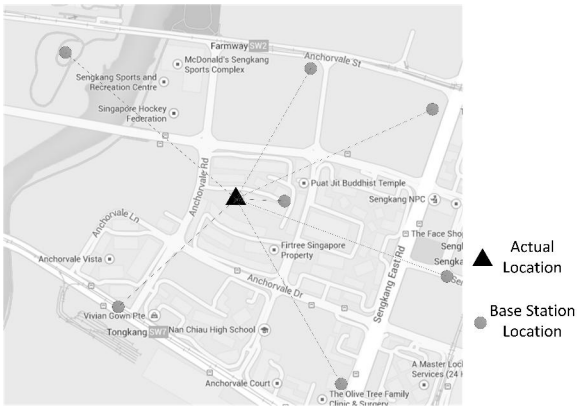

Fig. 4: Actual vs. Recorded (Base Station) Locations

We set up D3MO as $\tau = 3$hr and $t_\Delta = 1$hr. There are 1500 devices in the original data set. 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. 5, where three representative users are plotted and their regular areas are marked with black triangles. We note this application looks at the $\{\tau, t\}$ dimension of the data cube.

- User i has two frequent locations as shown in Fig. 5a. 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. 5b. 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. 5c. His/her locational results are all over the city and does not show any other regular active area.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

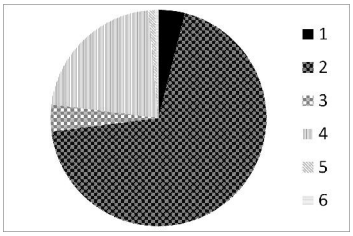

Fig. 7: Pie Chart of Cluster Membership

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 stored for each user in D3MO for $\tau = 3$hr. 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. 6. The percentage of users belong to each cluster can be found in Fig. 7. This is actually comparing the users behavior across the k dimension of the data cube of D3MO.

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:

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

This result is significant — we classified the users based on their behavior using D3MO, despite of the inaccuracy and inconsistency of the source locational data. Moreover, this solution is extremely outstanding in one feature: we do not 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.

VI. Conclusion

In this work, we have proposed a new scheme to study locational data, namely *Mobility Observation with Convex Hull Algorithm (D3MO)*. It uses two techniques: sliding time window with variable size, and convex hull algorithm to convert objects' 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 movement patterns of the objects can thus be observed. With the case study of mobile device locational data, we show that D3MO can be used effectively to study users' behavior and lifestyle. Moreover, it tolerates errors such as signal inaccuracy and varying frequency, and preserves users' private information hidden to the outside.

For future directions, we are extending D3MO to more kinds of locational data studies. We also want to formulate the relationship between trajectory property and polygon geometric properties, so that the analysis could be carried out in a standardized way.

References

[1] T. L. Nyerges, "Locational Referencing and Highway Segmentation in a Geographic Information System," *ITE Journal*, vol. 60, no. 3, pp. 27–31, 1990.
[2] N. Andrienko and G. Andrienko, *Exploratory Analysis of Spatial and Temporal Data*. Springer Berlin, Germany, 2006.
[3] I. Forum, *Improving Reliability on Surface Transport Networks*. OECD Publishing, 2010.
[4] W. Gao and G. Cao, "Fine-Grained Mobility Characterization: Steady and Transient State Behaviors," in *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2010, pp. 61–70.
[5] D. Pareek, *Business Intelligence for Telecommunications*. CRC Press, 2006, vol. 2.
[6] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory Pattern Mining," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 330–339.
[7] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Learning and Inferring Transportation Routines," *Artificial Intelligence*, vol. 171, no. 5, pp. 311–331, 2007.
[8] E. Clayirci and I. F. Akyildiz, "User Mobility Pattern Scheme for Location Update and Paging in Wireless Systems," *Mobile Computing, IEEE Transactions on*, vol. 1, no. 3, pp. 236–247, 2002.
[9] C. A. Patterson, R. R. Muntz, and C. M. Pancake, "Challenges in Location-Aware Computing," *Pervasive Computing, IEEE*, vol. 2, no. 2, pp. 80–89, 2003.
[10] Y. Lu and Y. Liu, "Pervasive Location Acquisition Technologies: Opportunities and Challenges for Geospatial Studies," *Computers, Environment and Urban Systems*, vol. 36, no. 2, pp. 105–108, 2012.
[11] W. Ting, "An Online Data Compression Algorithm for Trajectories (An OLDCAT)," *International Journal of Information and Education Technology*, vol. 3, no. 4, pp. 480–487, 2013.
[12] 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*. ACM, 2011, pp. 25–33.
[13] R. Becker, R. Cáceres, K. Hanson, S. Isaacman, J. M. Loh, M. Martonosi, J. Rowland, S. Urbanek, A. Varshavsky, and C. Volinsky, "Human Mobility Characterization from Cellular Network Data," *Communications of the ACM*, vol. 56, no. 1, pp. 74–82, 2013.
[14] M. De Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry*. Springer, 2008.
[15] B. J. Worton, "A Convex Hull-Based Estimator of Home-Range Size," *Biometrics*, pp. 1206–1215, 1995.
[16] M. Deypir, M. H. Sadreddini, and S. Hashemi, "Towards a Variable Size Sliding Window Model for Frequent Itemset Mining over Data Streams," *Comput. Ind. Eng.*, vol. 63, no. 1, pp. 161–172, Aug. 2012.
[17] D. G. Kirkpatrick and R. Seidel, "The Ultimate Planar Convex Hull Algorithm?" *SIAM journal on computing*, vol. 15, no. 1, pp. 287–299, 1986.
[18] T. M. Chan, "Optimal Output-Sensitive Convex Hull Algorithms in Two and Three Dimensions," *Discrete & Computational Geometry*, vol. 16, no. 4, pp. 361–368, 1996.
[19] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.
[20] D. Avis, D. Bremner, and R. Seidel, "How Good are Convex Hull Algorithms?" *Computational Geometry*, vol. 7, no. 5, pp. 265–301, 1997.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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

Fig. 5: Frequent Location of Users

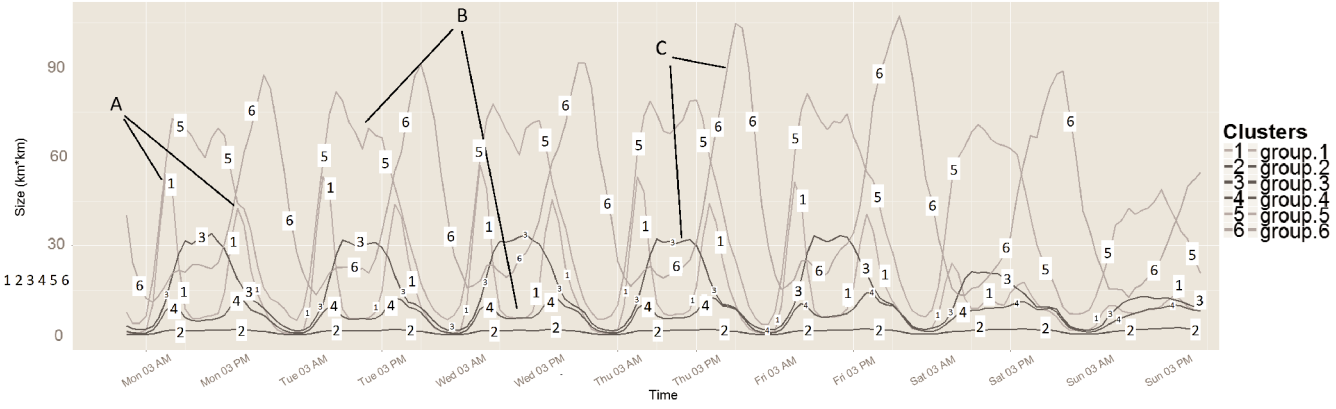

Fig. 6: Clusters based on Polygon Area Size over Time

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)