Discover User Behaviour from Trajectory as Polygons (TaP)

Ting Wang, *Senior Member, IACSIT*

***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. Locational data and trajectories are often used to discover behavior trends and clusters of users. In this paper, we propose a new scheme, namely *Trajectory as Polygons (TaP)*, which uses a legacy optimization technique-*Convex Hull*-to the new problem of user behavior discovery, and have made some interesting and useful discoveries. In particular, we found TaP effectively extract trajectory properties from polygons generated by convex hull algorithm with a time window. Moreover, with a case study, we show TaP is able to study peoples' lifestyle patterns while keeps their exact locations confidential.**

***Index Terms*—Spatial-temporal data model, convex hull algorithm, trajectory mining.**

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 equipment 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 have 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 define 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 used in *Trajectory Mining* [4], and are important sources for *Behavioral Studies*-a process in which information related to objects' behaviors, such as lifestyle, home location, and travel routing *etc.*, could be discovered [5].

Manuscript received November 8, 2013; revised January 14, 2014. This work was supported in part by the Singapore Economic Development Board (EDB) and National Research Foundation (NRF).

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

In this paper, we propose a new algorithm, namely *Trajectory as Polygons* (TaP). Using the legacy *Convex Hull* algorithm [6], together with a sliding time window mechanism, TaP uses polygon to represent objects trajectory, instead of using line segments like many existing works do. The mobility patterns and user behavior can be observed from the geometric properties (*e.g.* location, size, shape, and number of vertices/edges *etc.*) of these polygons. We note that TaP 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 TaP even when the source data is scarce and largely inaccurate.

A brief introduction to the existing works and challenges are discussed in Section II, followed by the introduction of the TaP algorithm in Section III. We talk about how to use TaP to study trajectories and study user behavior in Section IV, and a case study of how TaP could be used with locational data to discover people's lifestyles is discussed in Section V. Section VI concludes the paper with the strength of TaP and future research directions.

II. Existing Solutions and Challenges

Most of the existing trajectory mining 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 series of states and the transitions among them. Markov-chain and other related models [8] 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 [4]. It is then usually used to define clusters or places of interest.

Density Based: in large scale problems [9], importance of locations can simply be reflected by the density of data points in that area.

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 [10]. 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

DOI: 10.7763/IJAPM.2014.V4.250

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.

On the other hand, as brought up by Giannotti *et. al.* in [4], confidentiality maybe another factor we need to consider in the study of human mobility. For example, the users may not want themselves to be identified and "pin-pointed" by their location history. With TaP, we show that we are able to learn a person's lifestyle without knowing where she/he exactly has been by transforming trajectory to polygons using *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 [12]. 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 [13]. TaP 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 [14] to the algorithm, so that the timely change of the convex hull could be studied, and user behavior patterns can thus be observed.

III. TaP

As shown in Fig. 1[1], TaP takes the raw position data of an object[2] with UID *k* as input. It split 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 τ. Then each segment is processed by the convex hull algorithm and represented as a polygon *P*. That is to say, each polygon *P* corresponds to the movement of object *k* during a time window (determined by *t* and *τ*). The geometric properties, such as centroid location, area size, perimeter, and number of edges/vertices etc., are denoted as a function of *k*, *t* and *τ*, written as

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

An overall data flow chart of TaP can be found in Fig. 2.

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.

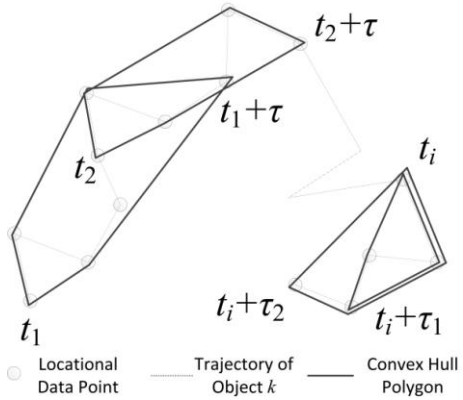


Fig. 1. Convex hull polygons of object *k*.

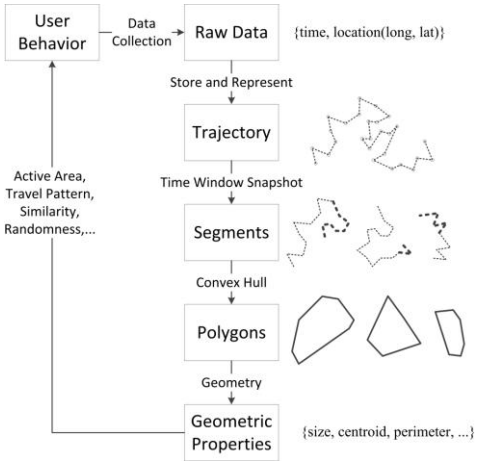


Fig. 2. Overall data flow chart of TaP.

Consider the general case when the input to the algorithm is a finite unordered set of points on a Cartesian plane. The lower bound on the computational complexity of finding the convex hull represented as a convex polygon is easily shown to be the same as for sorting using the following reduction.

For the set $x_1,\ldots, x_n$ numbers to sort consider the set of points $(x_1, x_1^2),\ldots, (x_n, x_n^2)$ of points in the plane. Since they lie on a parabola, which is a convex curve. It is easy to see that the vertices of the convex hull, when traversed along the boundary, produce the sorted order of the numbers $x_1,\ldots, x_n$. Clearly, linear time is required for the described transformation of numbers into points and then extracting their sorted order. Therefore in the general case, the convex hull of *n* points cannot be computed more quickly than sorting.

The standard $\Omega(n \log n)$ lower bound for sorting is proven in the decision tree model of computing, in which only numerical comparisons but not arithmetic operations can be performed; however, in this model, convex hulls cannot be computed at all. Sorting also requires $\Omega(n \log n)$ time in the algebraic decision tree model of computation, a model that is more suitable for convex hulls, and in this model convex hulls also require $\Omega(n \log n)$ time [12]. However, in models of computer arithmetic that allow numbers to be sorted more

[1] To show the trajectory clearly, the convex hull polygons in this figure is drawn slightly larger than they should be.

[2] We assume each object is identified by an integer UID.

quickly than O(n log n) time, for instance by using integer sorting algorithms, planar convex hulls can also be computed more quickly: the Graham scan algorithm [15] for convex hulls consists of a single sorting step followed by a linear amount of additional work.

As stated above, the complexity of finding a convex hull as a function the input size n is lower bounded by Ω(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 Θ(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 Ω(n log n) in the planar case [12]. 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*) [16]. A much simpler algorithm was developed by Chan in 1996, and is called *Chan's algorithm* [17].

A number of algorithms are known for the three-dimensional case, as well as for arbitrary dimensions [18]. 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 [19].

B. 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.

The value of τ usually does not change, which gives a series of constant length time windows. 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 times 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.

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 next section we discuss how the property of trajectory and user behavior can be derived from the geometric properties of the convex hull polygons and the sliding time windows.

IV. Discover User Behavior

The objective of TaP is to observe objects' movement and extract useful information. In this section, we demonstrate how movement information can be extracted from the polygons derived from convex hull algorithm, and how their geometric properties can be used to derive properties of the trajectory. To fully understand the object's movement, usually we need more than one of the trajectory properties to make conclusion. In this section, we demonstrate how to find out four useful properties, namely active area, traveling pattern, similarity, and randomness, of the objects. We note that in the following examples certain threshold maybe required to classify the trajectory type. However, as the exact value of the threshold is case-dependent, we will not be able to specify suitable values of them in this section.

A. Active Area

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 Section 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 trajectory mining 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 depends 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.

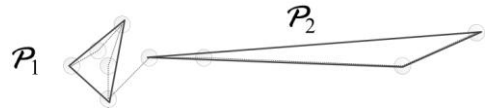


Fig. 3. Examples of active area detection.

When the trajectory is represented as polygons in TaP, the area size of a polygon is the area that the object has covered in the corresponding time window. Since the sliding time window has constant size, smaller polygons indicate the fact that the object spent 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 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 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 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 cannot 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.

B. Travel Patterns

How the object 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 object is traveling.

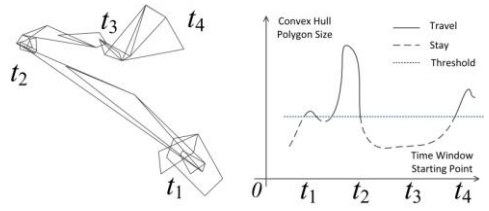


Fig. 4. Polygons sizes with different values of *t.*

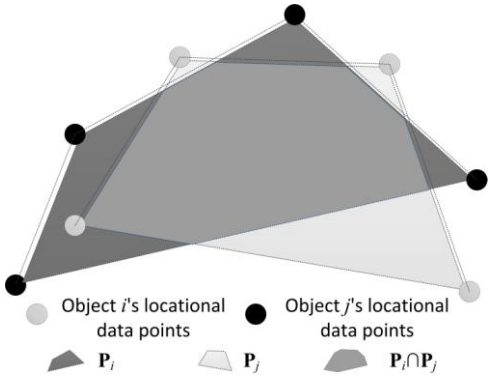


Fig. 5. Similarity between polygons $\mathbf{P}_i$ and $\mathbf{P}_j$.

Firstly, as demonstrate in the previous section, polygons with small size, but long perimeter and large edge length variation is a result of traveling object. The long edge in this kind of polygon shows the fact that the object 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 object is traveling. We can understand it as a large active area, in which the object goes to multiple places. When we observe that some polygons with bigger size appear between two or several smaller polygons, we understand the user is traveling among the active areas. Fig. 4 shows how the polygon size evolves with the value of time window starting point. Those higher values indicate "traveling" while the lower parts refer to "staying".

C. Similarity

Moving objects 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. 5, object *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. TaP converts their trajectories 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. 5, the high percentage of overlapping indicates the two trajectories are similar to each other. Quantitatively, a *similarity index* ***S*** can be measured as the size of overlapping area divided by the total area covered by the two polygons, written as

$$S = \frac{A_{P_1 \cap P_2}}{A_{P_1 \cup P_2}}$$

where *A* denotes the area size of the polygon. A value of ***S*** close to 1 will indicate the given polygons are similar to each other. We note the measure of similarity index can be easily extended to multiple objects by comparing polygon sizes of the same time window across different ID's.

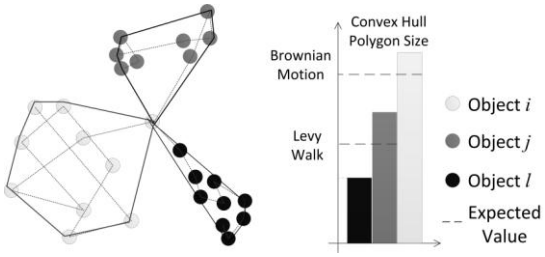


Fig. 6. Randomness reflected by polygon sizes.

The accuracy and validity of similarity could be improved if we put the value of *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 objects have polygons with high similarity index over several consecutive time windows, it could be more evident that these two objects travel in similar pattern.

D. Randomness

Objects may or may not move randomly. Non-random movement means the object has certain purpose which may be reflected by the mobility pattern. Finding out the randomness of an object 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 objects.

Thanks to many previous sound works such as [20] on the property of convex hull, we have the expect size of the convex hull formed by the trace when the object is moving in certain mobility model, such as Brownian motion, random walk or Levi walk *etc.* [21]. 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 object[3], we can calculate a theoretic value for the convex hull size assuming the object 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 object's moving pattern is to the model in our assumption. Usually purposeful movement will result in smaller convex hull size. In Fig. 6, we show how the three objects movement could be classified based on their randomness: object *i* being the most random, as its convex size is close to Brownian motion benchmark; object *j* and *l* show certain degree of purposeful movement, with *l* less random than *j*.

V. Case Study-Lifestyle Discovery

TaP 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 or signal triangulation. 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. 7 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.

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

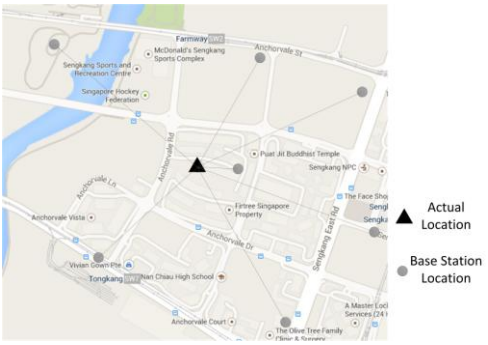


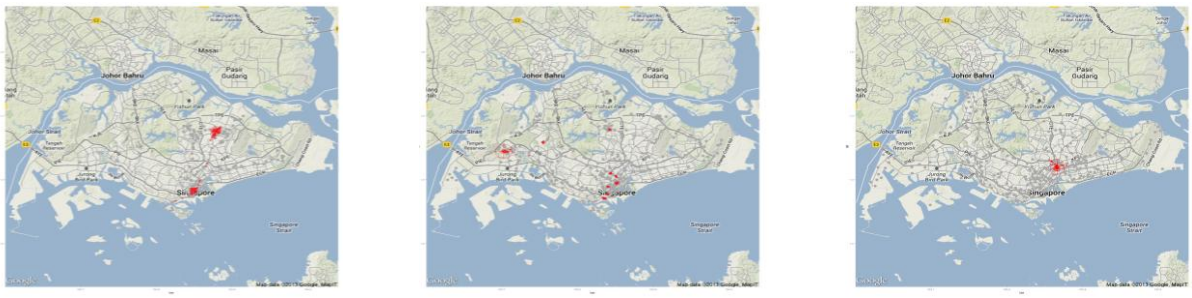
Fig. 7. Actual vs. recorded (base station) locations.

We set up TaP as τ = 3hr and t = 1hr. 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. 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. 8 a). From timestamps of the records (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. 8 b). 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. 8 c). His/her locational results are all over the city and do 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 TaP. 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. The percentage of users belong to each cluster can be found in Fig. 10.

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".



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

Fig. 8. Frequent location of users.

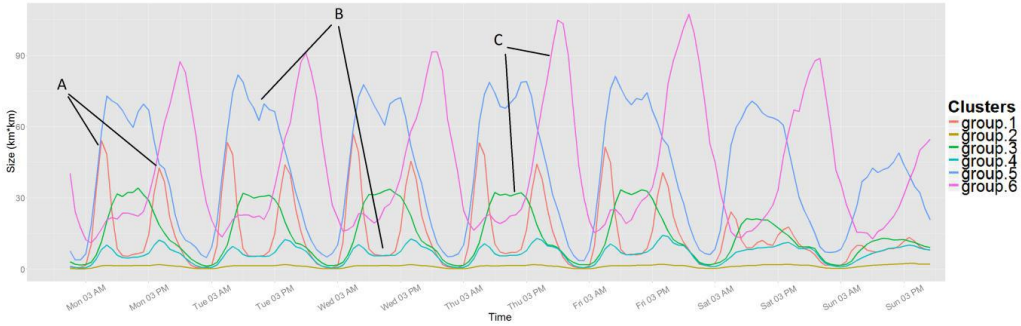


Fig. 9. Clusters based on polygon area size over time.

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 travellers (sales persons, etc.)
5) All-day travellers with lunch break (drivers, etc.)
6) Long-distance night-travellers (taxis, etc.)

This result is significant — we classified the users based on their behaviour using TaP, 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.

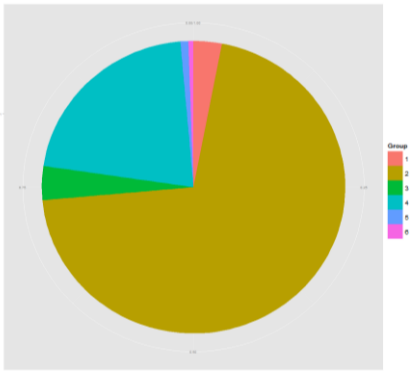


Fig. 10. Pie chart of cluster membership.

VI. Conclusion

In this work, we have proposed a new scheme to study locational data and dervie user behavior patterns, namely Trajectory as Polygons (TaP). It uses two techniques: sliding time window, and convex hull algorithm to convert objects' trajectories to a series of polygons. 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 TaP 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 TaP 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 systematic 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*, Berlin, Germany: Springer, 2006.

[3] Forum, *Improving Reliability on Surface Transport Networks*, OECD Publishing, 2010.

[4] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory Pattern Mining," in *Proc. the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2007, pp. 330–339.

[5] W. Gao and G. Cao, "Fine-Grained Mobility Characterization: Steady and Transient State Behaviors," in *Proc. the eleventh ACM international symposium on Mobile ad hoc networking and computing*, ACM, 2010, pp. 61–70.

[6] F. P. Preparata and S. J. Hong, "Convex Hulls of Finite Sets of Points in Two and Three Dimensions," *Communications of the ACM*, vol. 20, no. 2, pp. 87–93, 1977.

[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] K. Maruyama *et al*., "Pedestrian-Movement Prediction based on Mixed Markov-Chain Model," in *Proc. the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, 2011, pp. 25–33.

[9] S. Isaacman, R. Becker, R. Caceres, M. Martonosi, J. Rowland, A. Varshavsky, and W. Willinger, "Human Mobility Modeling at

Metropolitan Scales," in *Proc. the 10th international conference on Mobile systems, applications, and services*, ACM, 2012, pp. 239–252.
[10] 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.
[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] M. De Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry*, Springer, 2008.
[13] J. Worton, "A Convex Hull-Based Estimator of Home-Range Size," *Biometrics*, pp. 1206–1215, 1995.
[14] 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.
[15] R. L. Graham, "An Efficient Algorith for Determining the Convex Hull of a Finite Planar Set," *Information Pocessing Letters*, vol. 1, no. 4, pp. 132–133, 1972.
[16] G. Kirkpatrick and R. Seidel, "The Ultimate Planar Convex Hull Algorithm?" *SIAM journal on computing*, vol. 15, no. 1, pp. 287–299, 1986.
[17] 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.
[18] 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.
[19] D. Avis, D. Bremner, and R. Seidel, "How Good are Convex Hull Algorithms?" *Computational Geometry*, vol. 7, no. 5, pp. 265–301, 1997.
[20] S. N. Majmdar, A. Comtet, and J. Randon-Furling, "Random Convex Hulls and Extreme Value Statistics," *Journal of Statistical Physics*, vol. 138, no. 6, pp 955–1009, 2010.
[21] R. N. Mantegna and H. E. Stanley, "Stochastic Process with Ultraslow Convergence to a Gaussian: the Trancated Levy Flight," *Physical Review Letters*, vol. 73, no. 22, pp. 2946, 1994.

Wang Ting was born in 1983 in Chengdu, Sichuan, China. He came to Singapore for his undergraduate studies in 2001. He obtained his bachelor's degree with honors in 2005 and subsequently Ph.D in 2011 both at Nanyang Technological University (NTU), Singapore.

He worked as a demand planner at Apple South Asia and joined SAP as a Data Scientist in 2012.

His research interests include data mining, mathematical modeling and algorithmic optimization.

Dr. Wang loves music and sports. He considers family as his greatest award. He has a son, and he is a good cook-said his wife.