



**Queensland University of Technology**  
Brisbane Australia

This is the author's version of a work that was submitted/accepted for publication in the following source:

Lau, Billy Pik Lik, Hasala, Marakkalage Sumudu, Kadaba, Viswanath Sanjana, [Balasubramaniam](#), [Thirunavukarasu](#), Yuen, Chau, Yuen, Belinda, & [Nayak, Richi](#)  
(2017)

Extracting point of interest and classifying environment for low sampling crowd sensing smartphone sensor data. In *IEEE International Conference on Pervasive Computing and Communication Workshops (IEEE PerCom 2017)*, 13-17 March 2017, Kona, Big Island, Hawaii.

This file was downloaded from: <https://eprints.qut.edu.au/104992/>

© 2017 [Please consult the author]

**Notice:** *Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:*

# Extracting Point of Interest and Classifying Environment for Low Sampling Crowd Sensing Smartphone Sensor Data

Billy Pik Lik Lau\*, Marakkalage Sumudu Hasala\*, Viswanath Sanjana Kadaba\*,  
Balasubramaniam Thirunavukarasu†, Chau Yuen\*, Belinda Yuen\*, Richi Nayak†

Engineering Product Development, Singapore University of Technology and Design, Singapore \*  
School of Electrical Engineering and Computer Science, Queensland University of Technology, Australia †  
Email: billy\_lau@sutd.edu.sg, marakkalage@mymail.sutd.edu.sg, yuenchau@sutd.edu.sg

**Abstract**—The advancement of smartphones with various type of sensors enabled us to harness diverse information with crowd sensing mobile application. However, traditional approaches have suffered drawbacks such as high battery consumption as a trade off to obtain high accuracy data using high sampling rate. To mitigate the battery consumption, we proposed low sampling point of interest (POI) extraction framework, which is built upon validation based stay points detection (VSPD) and sensor fusion based environment classification (SFEC). We studied various of clustering algorithm and showed that density based spatial clustering of application with noise (DBSCAN) algorithms produce most accurate result among existing methods. The SFEC model is utilized for classifying the indoor or outdoor environment of the POI clustered earlier by VSPD. Real world data are collected, benchmarked using existing clustering method to denote effectiveness of low sampling rate model in high noise spatial temporal data.

**Keywords**—Big Data, POI, Clustering, Environment Classification, Trajectory, Sensor Fusion

## I. INTRODUCTION

Data collected using mobile devices such as smart phone and wearable devices have been explicitly studied by various researchers in [1], [2] as they are able to generate various insights about a particular user life pattern. Among all the information captured, point of interest (POI), user mobility, user life pattern, and transportation mode gained a lot of attention; but major existing crowd sensing applications suffer from the drawbacks of high battery consumption. The reason behind high battery consumption is due to high sampling rate as well as limited battery capacity of mobile devices, which render most of the methods only suitable for short term data collection.

To solve the aforementioned challenges, we proposed *ultra low* sampling architecture for extracting POI utilizing various sensor information available on a smart-phone. One of the main challenges of ultra low sampling architecture is trade off for accuracy in order to obtain data at a very long interval compared to traditional method. Another problem that may arise is highly correlated with the data quality, which may produce inaccurate output after computation.

The main objective of the paper is to study POI based on the various sensor information generated by smart phone using low sampling rate. Such technique will find application in better understanding user life pattern, which in return can improve user life style. However, in a crowd sourcing /

sensing environment, there are a wide variety of devices by different users, and it is not possible to retrieve ground truth from the users. In this paper, our main focus is to verify the effectiveness of clustering algorithm based on data collected from volunteer users. Our model require minimal attention and is suitable for helping elderly to understand their life pattern after retirements. These insights can be used for the authorities to improve certain location facilities, which will be reported in the continued works.

Various information has been collected through the application we have previously developed in [3]. Data collected from smart-phone are uploaded to the server and off-line analysis are performed to gain insights about POI and trajectory. We first extract data from database, and later perform several preprocessing techniques namely *denoise* and *timeSync*. The former mentioned technique allows us to remove any potential noise, outliers, and duplicate data; whereas the latter mentioned technique is used for synchronizing the data with different timestamps. Subsequently, we apply VSPD algorithm to extract the POI, while the DBSCAN is used for clustering similar POI. Trajectory can be obtained through labeling each POI with the time stamp based on duration of each stay point. As our main concern is to determine POI, we further study the characteristic of the POI and classify it into indoor vs outdoor, or private vs public using SFEC technique. The SFEC is implemented using multi sensor data obtained from mobile devices.

Our contributions in this paper are listed as follows:

- We present VSPD algorithm, which is capable to distinguish valid stay points due to poor the data quality of low sampling rate. Stay points are validated using the proposed module and later clustered into POI based on temporal data.
- SFEC method is presented to distinguish POI by its indoor and outdoor characteristic. In addition, we also categorize the POI by its property, whether it is private or public premises.

The rest of the paper is structured as follows: In Section II, we propose ultra low sampling architecture and discuss about the data processing techniques. In Section III, VSPD algorithm is introduced and benchmarked against existing clustering and stay point detection algorithms. After clustering similar POI, we performed environment classification

using SFEC method and show the efficiency of our methods in Section IV. Finally, we conclude our work in Section V.

## II. SYSTEM ARCHITECTURE AND FRAMEWORK

In this section, we present the ultra low sampling rate architecture for data collection and processing.

### A. User Trajectory and Point of Interest Detection

User mobility has been studied in [4], [5] and several features extraction show the necessity of extracting stay points before obtaining insights from user trajectory. Stay points are known as POI in our study scenario, where it can be a particular user home, work place, leisure spot, and etc.

To study the POI, trajectory data and user mobility are often used to visualize the path a user takes from one POI to another. Extracting trajectory features yields a lot of important information such as travel distance, duration, POI location, transportation mode, and etc. On the other hand, transportation mode (TM) is identified through the feature extraction using methods such as probability estimation [6]. Other related works such as [1], [7] use mobile phone as platform for data collection and try to capture trajectory data using various interpolation methods (linear, cubic, and nearest). Finer detail of TM, such as motion patterns have been studied in [8], which uses Hidden Markov Model (HMM) and  $k$ -means clustering for obtaining motion patterns; then it is also serve as anomaly detection.

However, most of the methods only deal with high sampling rate of data and are not suitable for day long data collection due to limited battery capacity. We incorporated validation function for the stay point detection in order to validate low sampling rate of data.

### B. System Architecture and Data Processing

The system framework for collecting POI information can be divided into two phases, which are data collection and data processing. Data collection phase involves installation of mobile application at volunteers' smart phone. Data collected from mobile devices, which are (1) Location information, (2) Social Information, and (3) Activity Information. The location information consists of user geographical positioning data, while the social information is composed of sound information collected from the mobile devices microphone. The activity information consists of data such as TM and type of activity that users are engaged in.

We have previously developed the application using Android frameworks in [3] and is used for data collection in this paper. It has been further fine-tuned to adapt the sampling rate of 5 minutes and optimized for off-line data collection, when there is no real-time Internet connection. The overview of the system architecture is shown in Fig. 1.

The preprocessing of data is performed in order to generate usable data to extract useful features. First, the raw data will be passed through *denoise* process to remove any duplicate records. The denoise module checks the data time-stamp and eliminates data with same time-stamp. The reason

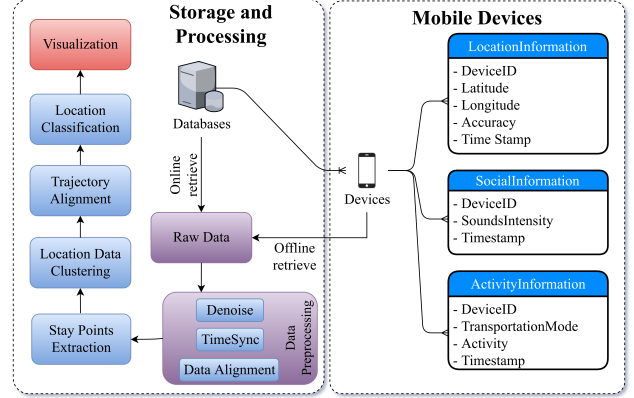


Figure 1. Overview of the system architecture

for duplicated data is mostly due to the redundant data kept by mobile phone, which fails to receive acknowledgment from server upon upload. The timeSync module act as the function for linking different data to a similar time stamp in order to perform data fusion computation, which will be presented in later part. Data alignment starts by aligning the data from different database based on the synchronized time for each dataset into single matrices. The overall data processing is shown in the Algorithm 1.

#### Algorithm 1 Overall System Algorithm

---

**Data:**  $\{L, Q\}$  where  $i^t = 1, 2, \dots, N$   
**begin**  
  preprocess( $\{L, Q\}$ )  
  stayPoints = ValidationStayPointAlgorithm( $P$ )  
  clusteredData = DBSCAN(stayPoints)  
   $T_x$  = Trajectory(stayPoints,  $P$ , clusteredData)  
  timeSync( $\{L, Q\}$ )  
  SFEC( $\{L, Q\}$ )

---

We first define set of location data collected from the mobile devices as  $L = \{P_1, P_2, \dots, P_N\}$ , where  $P$  can be denoted as  $P_i = \{\varphi_i, \lambda_i, a_i, t_i\}$ . The latitude and longitude data are represented by  $\varphi$  and  $\lambda$  respectively, which is part of GPS Information  $P_i$ . The  $t$  represents the time-stamp of data obtained and  $a$  denotes the accuracy of GPS module.

We denote  $Q = \{S_1, S_2, \dots, S_N\}$  as segment of stay points and define each sensor value for each  $Q$  as following  $S_i = \{v_i, a_i, \mathcal{N}_i, \beta_i, \phi_i\}$ . Each data are partitioned into segments based on the stay points. The  $v_i$  represents the velocity of the user and this is captured using the mobile device's API, and  $a_i$  is the accuracy of location data.  $\mathcal{N}_i$  is the normalized noise intensity, and  $\phi_i$  represents the maximum occurred activity within a time segment.

We noticed different mobile devices possess different noise levels due to different hardware component and manufacturer. In order to generate a fair comparison of noise level across other devices, we will normalized the noise data using the Equation (1) below:

$$\mathcal{N} = 10 \left( \frac{s - s_{min}}{s_{max} - s_{min}} \right) \quad (1)$$

where  $s$  represents the noise amplitude level. The  $s_{min}$  and  $s_{max}$  represent minimum and maximum noise level captured

respectively based on user data at a particular time frame (i.e. one month data).

### III. POINT OF INTEREST DETECTION

In this section, we discuss the POI extraction module such as stay points detection and data clustering, that group similar POI into cluster.

#### A. Point of Interest Clustering

$K$ -means clustering [9] is one of the common clustering techniques in handling numerical data since it offers fast and easy usage compared to others. However, finding an optimal  $k$  is one of the challenges as it is an NP hard problem. In [10], they have studied the use of  $k$ -means clustering on spatial and temporal data for POI clustering. However,  $k$ -means is not capable of identifying arbitrary shape of cluster, which makes it not ideal for our proposed architecture. Alternate common clustering method is density based clustering such as DBSCAN [11] and OPTICS [12]. In [13], they have used customized DBSCAN algorithm for geo-tagging photo which uses density threshold and adaptive density. However, their proposed method only concentrates on spatial information and ignores the temporal data.

Low ultra sampling data collection has a high impact on the quality of data, which may results inaccurate data clustering and wrong POI generated. Therefore, we studied different kind of clustering technique as well as stay point detection to ensure best features extraction technique is chosen.

#### B. Validation based Stay Points Detection (VSPD) Algorithm

VSPD algorithms utilize the stay point detection algorithm first proposed in [4] and we added validation function to remove incompetent POI. The validation function in VSPD algorithm has been deployed in order to adapt ultra low sampling architecture, which is presented in Algorithm 2.

#### Algorithm 2 Validation based Stay Point (VSPD) algorithm

**Data:**  $P_i$  where  $i = 1, 2, \dots, N$

**Result:**  $SP_x$

```

for  $k \leftarrow 1$  to  $N$  do
  while  $j \leftarrow (k + 1)$  to  $N$  do
    Check( $a_{\{i, i+1, \dots, j\}}$ , threshold( $a$ )) using Equation (7)
    Calculate  $d$  using Equation (4)
    if  $d > a$  then
      Calculate  $\Delta t$  using Equation (5)
      if  $\Delta t > \text{threshold}(t)$  then
        if Validity( $d, \Delta t$ ) then
          addStayPoint( $P_i, P_j$ )
        else
          break

```

Before going into detail of the algorithm, we first define the following key information, which are stay points and trajectory. Stay points are defined as  $SP$ , where it represents potential POI a particular user is at. Each stay point can be arranged to trajectory by adding time-stamp, resulting  $T_x = \{SP_1, SP_2, \dots, SP_x\}$ .

Reachability of distance between two points is a crucial piece of information required for DBSCAN in order to correctly cluster similar POI. Throughout this paper, the distance between two points  $\{\varphi_1, \lambda_1\}$ , and  $\{\varphi_2, \lambda_2\}$  are calculated using the Harvesine formula as shown in Eqn. (2):

$$\text{hav}\left(\frac{d}{r}\right) = \text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1) \quad (2)$$

where harvesine function  $\text{hav}()$  can be found in [14] and  $d$  is the distance between two points and  $r$  is the radius of earth. The  $\text{hav}()$  used in Eqn.(3) is defined as follows:

$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2} \quad (3)$$

To obtain the distance between two points, we apply the inverse Harvesine function for Equation (4) by using arcsine function as follows:

$$d = r \times \text{hav}^{-1}(h) = 2r \arcsin\left(\sqrt{h}\right) \quad (4)$$

After obtaining the distance between two points, we can formulate duration difference between two stay points  $\Delta t$  as follows:

$$\Delta t = t_{i+1} - t_i \quad (5)$$

where  $t_i$  is the time-stamp for current sample and  $t_{i+1}$  is the timestamp for the previous sample.

Validation function is as a core function in detecting valid stay points. Ultra low sampling may generate high noise data from environment such as indoor buildings, tunnel, and underground. To eliminate potential noise accounted into stay points, we formulate the validation function as follows:

$$\text{validaty}(d, \Delta t) = \Delta t < \Theta t \text{ and } d < \Theta d \quad (6)$$

where  $\Theta t$  is the threshold for time and  $\Theta d$  is the threshold for distance.

Subsequently, the  $\text{eps}$  and  $\text{minPts}$  need to be determined before applying DBSCAN algorithm. The  $\text{eps}$  are derived using the accuracy of GPS location as follows:

$$\text{eps} = \begin{cases} a_i + a_{i-1} & \text{if } a_i + a_{i-1} < \Theta l \text{ where } \text{minPts} = 1 \\ \Theta l & \text{if } a_i + a_{i-1} \geq \Theta l \text{ where } \text{minPts} = 1 \end{cases} \quad (7)$$

where accuracy of GPS data are calculated is considered for checking the validity of the stay points. For example, if an user entered a tunnel and GPS points remain fixed until he/she exit the tunnel after  $t$  time. Our validation model in Eqn. 6 will deem the period of entering tunnel as part of traveling because the time taken and distance traveled is considered not valid. In Eqn.(7), a threshold of  $\Theta l = 200$  was included to prevent GPS accuracy become too large and form a potential cluster. The  $\text{minPts}$  is set to 1 since we do not treat any POI as outliers at the moment.

#### C. Evaluation of Stay Point Detection and Clustering

To evaluate the proposed VSPD algorithm, we have few volunteers installed the application on their smart phone and proceed with their daily life. After data collection performed, we will request ground truth from every participant in order to correctly identify the POI they have been to.

## Effect of Different Clustering Algorithms



Figure 2. Comparison of Different Stay Points Extraction and Clustering Techniques

First, we study the POI detection algorithm, which consists of (1) VSPD algorithm, (2) stay point algorithm [4] and (3) online clustering [3]. Next, We evaluate the clustering method for each stay point detection algorithm using (1) DBSCAN, (2) K-Means, (3) Hierarchical. It should be noted, that the  $k$ -means clustering's  $k$  is determined using Davies Bouldin index. The comparison of different stay point extraction algorithm is presented in Figure 2.

Based on the results, VSPD algorithm and DBSCAN managed to extract POI correctly compared to others. Original stay point algorithm clustering wrong results are shown in red solid lined box. It is unable to detect the POI correctly as it is unable to distinguish the stay points due to poor indoor GPS accuracy. We manually inspect the GPS points located in the cluster 10 and found out that indoor low accuracy led to incorrect stay points extraction. It is observed that cluster 6 provide incorrect POI compared to ground truth and we highlighted using dotted grey box. By comparing the different clustering methods, stay point extraction is more crucial in term of extracting the correct POI from the raw database. The clustering technique used after stay point extraction algorithm yields significant changes afterwards. However, by comparing the consistency of the algorithm, we prefer DBSCAN over other clustering techniques due to its capabilities of forming arbitrary shape cluster, as the data collected is never in a consistent shape to begin with.

## Effect of Different Clustering Parameters

The next data extraction depends on the duration of stay time at a particular POI. We adjust the delta  $t$  accordingly to examine the granulated detail of possible stay point. The highlighted area of solid black box and grey dash lined box in Figure 3 generates correct POI according to the ground truth.

However, there is a slight error as framed in black dotted box which is due to transition of GPS accuracy from high to low. This causes a gap between two point and the algorithm treat it as two different location and hence error occurred. After verifying with the volunteers, the reason for such occurrence is die to traffic congestion. This led the algorithm to think the users is at a fixed position as it the time does not exceed the  $\Delta_t$ . They appear to be invalid in the validation function in VSPD algorithm and does not form a cluster.

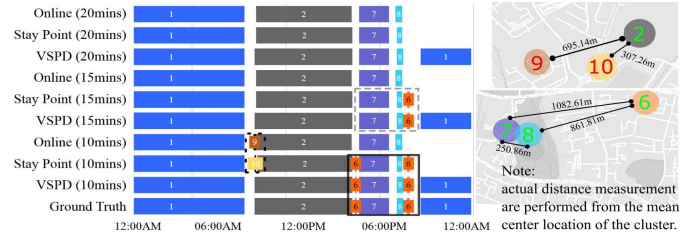


Figure 3. Comparison of different time threshold for stay point extraction algorithm

## Effect of Different Mobile Devices

Data collection using different smart phone such as Samsung Tab 7, Galaxy Alpha A8 , LG G-Pro 2 are tested. Do note that this is the first phase of our works, we have deployed the application and collected the data more than 100 users in our next works. In 4 same POI are labeled and colored as same number, where other locations are denoted differently. Slight drift of POI is observed through all devices tested in the same POI as highlighted in Grey solid box and black dash lined box. Despite slight difference in check-in and check-out time, VSPD algorithm and DBSCAN managed to extract the POI correctly regardless of device vendor. The VSPD algorithm has the computational complexity



Figure 4. Data collected using different Mobile Devices

of  $O(N^2)$  where validation function is  $O(1)$  step with no extra computational cost and is able to eliminate wrong stay points.

## IV. SENSOR FUSION BASED ENVIRONMENT CLASSIFICATION

### A. Indoor and Outdoor Classification

IO classification have been studied by applying several different approaches. It is can be divided into two approaches, which uses image processing and sensor fusion.

In [15], IODetector has been proposed to classify IO environment using three sensors (cellular module, light sensor, and magnetic sensor). They have combined sensors and wireless approximation methods to determine whether the environment belongs to indoor, outdoor out corridor. They have utilized HMM in conjunction with Viterbi algorithm to estimate the IO condition. However, Radu et al. [16] claimed that their supervised methods are far superior by introducing more features into IO classification. They have applied co-training methods and managed to achieve 92% of accuracy compared to GPS and IODetector. Other related work can be found in [17], [18].

In [19], they have proposed IO classification using image processing. Waleed T. et al [20] has presented classification technique that uses GIST descriptor in conjunction with neural network classifiers to decide whether image is taken indoor or outdoor. Based on their proposed method, they managed to achieve 90.8% of accuracy by extracting GIST features and using feed-forward neural network to feed training samples.

Since image processing requires training and a camera to be turned on during data collection, it will end up with high battery consumption. Hence, it is not suitable for ultra low sampling rate. Considering the case, mobile device is in pocket, it might end up in inaccurate representation of the IO. To overcome such scenario, we applied the notion of sensor fusion for data collection, in order to achieve better battery management and more accurate environment classification.

### B. Sensor Fusion based Environment Classification (SFEC)

The stay points are classified as indoor vs. outdoor and private vs. public, using sensor fusion based environment classification. An indoor environment is where the average GPS accuracy for the POI is above a certain threshold, while an outdoor environment is where the average GPS accuracy for the POI is below that threshold (here, the GPS accuracy is returned by Android API, in which, when a higher value for accuracy is returned, it means the GPS accuracy is lower). A private environment is where the noise level for the POI is below a certain threshold and, while a public environment is where the noise level for the POI is above that threshold. Since we cannot obtain a 100% estimation to determine the type of environment where the POI is, we assign percentage confidence levels for determination.

The classifier requires multi-sensor data (i.e. GPS accuracy, Noise, Battery Level, Light etc.), from start and end time of the POI. The duration of POI is divided into 5 minute slots, and each of those slots are given a confidence percentage for type of above mentioned environments. Each type of environment is labeled into one of the 4 different categories, such as indoor, outdoor, private, and public, which are encoded into  $\{1, 2, 3, 4\}$  respectively.

Total percentage confidence level for a particular type of environment, for each type of slot, is calculated using the Equation (8), where,  $n$  is number of 5 minute slots in POI,  $P_c$  is confidence percentage,  $S_k^c$  is percentage of  $k^{\text{th}}$  slot being type  $c$ , and  $c$  is type of environment where  $c = \{1, 2, 3, 4\}$ , and  $(1 \leq k \leq n)$ . If there is no data in a slot for the classification, we use  $P_0$  to indicate the environment type 'Unclassified'.

$$P_c = \frac{1}{n} \times \sum_{k=1}^n S_k^c \quad ; \text{ if } n > 0 \quad (8)$$

The confidence level calculation for the environment type (1) Indoor and (2) Outdoor, are presented in the Eqn. (9) and Eqn. (10) respectively.

For  $P_1$ , percentage contributions from sensors are, 90% by GPS accuracy( $G$ ), 5% by battery level ( $\beta$ ), where  $\beta = 1$  if battery is charging, and  $\beta = 0$  otherwise, and 5% by Activity = 'Still' (denoted by  $\alpha_s$ , where  $\alpha_s = \{0, 1\}$ ) which is returned by location API. In Eqn.(9) and Eqn.(10),  $Th_G$  is threshold GPS accuracy, and  $x$  is average GPS accuracy in the slot.

$$P_1 = \left[ \left( \frac{x - Th_G}{Th_G} \times 0.9 \right) + \left( (\beta + \alpha_s) \times 0.05 \right) \right]; \text{ if } x > Th_G \quad (9)$$

For  $P_2$ , percentage contributions from sensors are, 90% by  $G$ , and 10% by light level ( $l$ ), where  $l = 1$  if light level is above threshold level  $Th_l$  or  $l = 0$  otherwise.  $Th_G = 30$ ,  $Th_N = 5$ , and  $Th_l = 1000$  based on empirical studies.

$$P_2 = \left[ \left( \frac{Th_G - x}{Th_G} \times 0.9 \right) + \left( Th_l \times 0.1 \right) \right]; \text{ if } x < Th_G \quad (10)$$

The confidence level calculation for the environment type (2) Private and (3) Public, are presented in Eqn. (11) and (12) respectively where,  $Th_N$  is threshold noise level, and  $y$  is average normalized noise level in the slot.

For  $P_3$ , percentage contributions from sensors are, 90% by noise level, and 10% by Activity = 'Still'( $\alpha_s$ ).

$$P_3 = \left[ \left( \frac{Th_N - y}{Th_N} \times 0.9 \right) + \left( \alpha_s \times 0.1 \right) \right]; \text{ if } y < Th_N \quad (11)$$

For  $P_4$ , percentage contributions from sensors are, 90% by noise level, and 10% by Activity = 'Walking'(denoted by  $w$ , where  $\alpha_w = 0, 1$ ).

$$P_4 = \left[ \left( \frac{y - Th_N}{Th_N} \times 0.9 \right) + \left( \alpha_w \times 0.1 \right) \right]; \text{ if } y > Th_N \quad (12)$$

### C. Evaluation of SFEC

SFEC is evaluated in real-world scenario by collecting multi-sensor information from mobile device of a volunteer user. Table I shows calculated percentage confidence levels of 4 POI along with their ground truth. When there is no data for a particular environment type, it is labeled as '-'. For POI 1,  $P_2$  and  $P_4$  have substantial difference when compared to  $P_1$  and  $P_3$  respectively. Similarly, for POI 2,  $P_1$  and  $P_3$  have substantial difference in comparison with  $P_2$  and  $P_4$  respectively. For POI 3,  $P_1$  and  $P_4$  have substantial difference in contrast with  $P_2$  and  $P_3$  respectively. For POI 4, no noise data found possibly due to power management profile of device which causes absence of data and indoor outdoor classification turned out to be incorrect. Moreover,  $P_1$  and  $P_2$  have no substantial difference to distinguish the environment type. The environment of all POI (except POI 4) were able to estimate correctly, using SFEC method. However, by comparing the  $P_1$  and  $P_2$  values of POI4, they are closed to each another, this may give us hint that the estimation could be in error.

Table I  
CONFIDENCE PERCENTAGES FOR SFEC

Classification	POI 1	POI 2	POI 3	POI 4
P1 (Indoor)	-	83.94	59.78	35.28
P2 (Outdoor)	60	2.42	16.24	23.94
P3 (Private)	-	22.35	0.61	-
P4 (Public)	73.12	-	29.66	-

Estimation	Outdoor Public	Indoor Private	Indoor Public	Indoor Unknown
Ground Truth	Outdoor Public	Indoor Private	Indoor Public	Outdoor Public

## V. CONCLUSION

In a nutshell, this paper focuses on low sampling and sensor fusion techniques for POI extraction and environment classification for smart-phone location data. We introduced validation function for the POI extraction and clustered identical POI to generate user trajectory. Furthermore, we implemented SFEC method to identify the type of environment using various sensor input. In future, we will expand the analysis model to study regional mobility and travel pattern with a large scale crowd sourcing / sensing activity, where ground truth is hard to obtain.

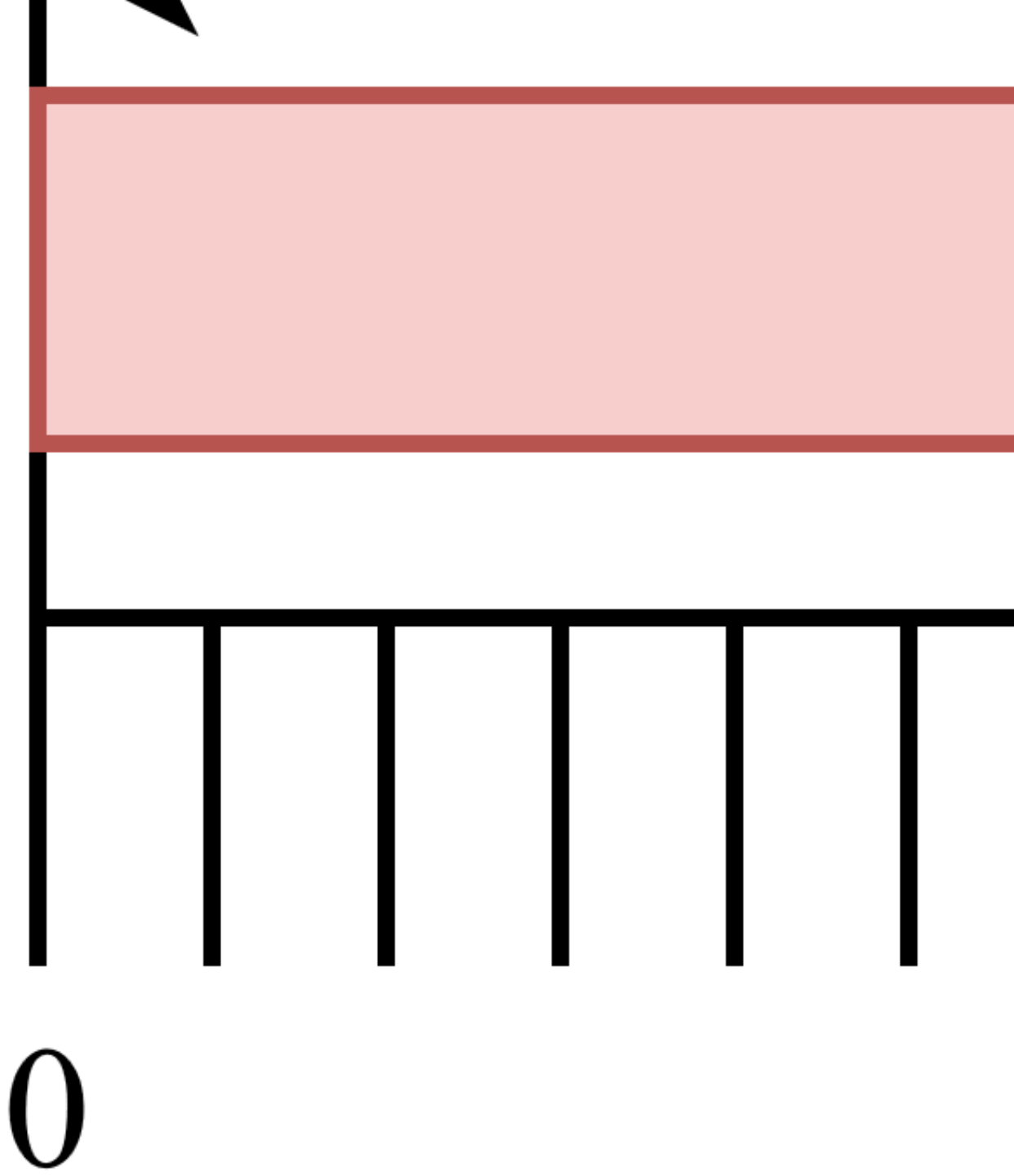
## ACKNOWLEDGMENT

Authors would like to express their sincere gratitude to all the participants in the study who allowed us to collect their location information. This research is supported by the Lee Kuan Yew Centre for Innovative Cities under Lee Li Ming Programme in Aging Urbanism.

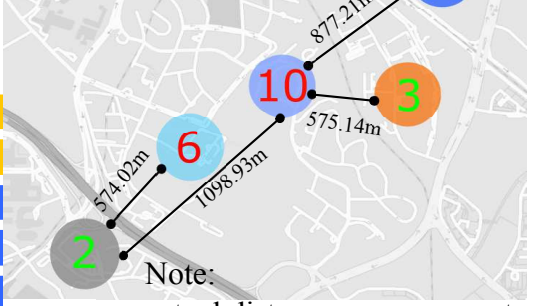
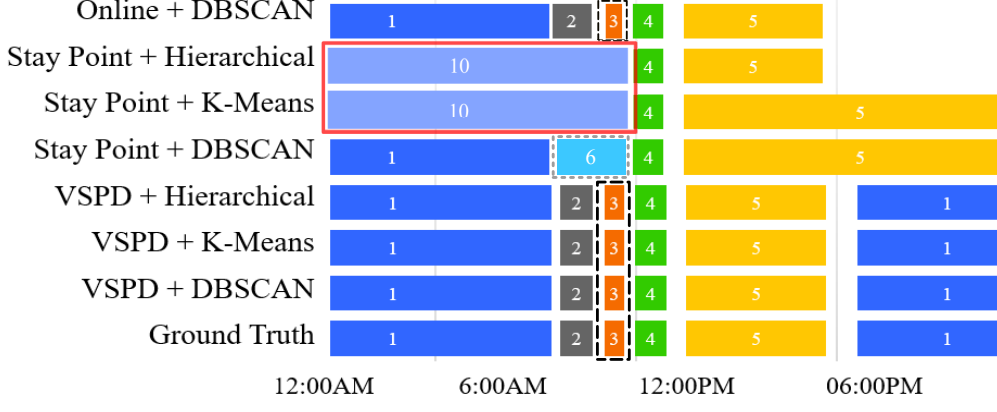
## REFERENCES

- [1] S. Hoteit, S. Secci, S. Sobolevsky, C. Ratti, and G. Pujolle, "Estimating human trajectories and hotspots through mobile phone data," *Computer Networks*, vol. 64, pp. 296–307, 2014.
- [2] C. Kang, S. Sobolevsky, Y. Liu, and C. Ratti, "Exploring human movements in singapore: a comparative analysis based on mobile phone and taxicab usages," in *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*. ACM, 2013, p. 1.
- [3] S. K. Viswanath, C. Yuen, X. Ku, and X. Liu, "Smart tourist-passive mobility tracking through mobile application," in *International Internet of Things Summit*. Springer, 2014, pp. 183–191.
- [4] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. ACM, 2008, p. 34.
- [5] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate gps trajectories," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 352–361.
- [6] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw gps data for geographic applications on the web," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 247–256.

- [7] Y. Zheng, X. Xie, and W.-Y. Ma, "Geolife: A collaborative social networking service among user, location and trajectory," *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.
- [8] N. Suzuki, K. Hirasawa, K. Tanaka, Y. Kobayashi, Y. Sato, and Y. Fujino, "Learning motion patterns and anomaly detection by human trajectory analysis," in *2007 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2007, pp. 498–503.
- [9] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
- [10] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.
- [11] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [12] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod Record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.
- [13] S. Kisilevich, F. Mansmann, and D. Keim, "P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos," in *Proceedings of the 1st international conference and exhibition on computing for geospatial research & application*. ACM, 2010, p. 38.
- [14] C. Veness, "Calculate distance, bearing and more between latitude/longitude points," *not dated*, <http://www.movable-type.co.uk/scripts/latlong.html>, 2010.
- [15] M. Li, P. Zhou, Y. Zheng, Z. Li, and G. Shen, "Iodetector: A generic service for indoor/outdoor detection," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 2, p. 28, 2015.
- [16] V. Radu, P. Katsikouli, R. Sarkar, and M. K. Marina, "A semi-supervised learning approach for robust indoor-outdoor detection with smartphones," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. ACM, 2014, pp. 280–294.
- [17] I. Bisio, A. Delfino, and F. Lavagetto, "A simple ultrasonic indoor/outdoor detector for mobile devices," in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2015, pp. 137–141.
- [18] H. Zou, H. Jiang, Y. Luo, J. Zhu, X. Lu, and L. Xie, "Bluedetect: An ibeacon-enabled scheme for accurate and energy-efficient indoor-outdoor detection and seamless location-based service," *Sensors*, vol. 16, no. 2, p. 268, 2016.
- [19] L. Zhang, M. Li, and H.-J. Zhang, "Boosting image orientation detection with indoor vs. outdoor classification," in *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*. IEEE, 2002, pp. 95–99.
- [20] W. Tahir, A. Majeed, and T. Rehman, "Indoor/outdoor image classification using gist image features and neural network classifiers," in *2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET)*. IEEE, 2015, pp. 1–5.







Note:  
actual distance measurement  
are performed from the mean  
center location of the cluster.

