GruMon: Fast and Accurate Group Monitoring for Heterogeneous Urban Spaces

Rijurekha Sen*
MPI-SWS
rijurekha@mpi-sws.org

Youngki Lee
School of Information Systems
Singapore Management University
youngkilee@smu.edu.sg

Kasthuri Jayarajah
School of Information Systems
Singapore Management University
kasthurij.2014@phdis.smu.edu.sg

Archan Misra
School of Information Systems
Singapore Management University
archanm@smu.edu.sg

Rajesh Krishna Balan
School of Information Systems
Singapore Management University
rajesh@smu.edu.sg

Abstract

Real-time monitoring of groups and their rich contexts will be a key building block for futuristic, *group-aware mobile services*. In this paper, we propose *GruMon*, a fast and accurate group monitoring system for dense and complex urban spaces. *GruMon* meets the performance criteria of *precise* group detection at *low latencies* by overcoming two critical challenges of practical urban spaces, namely (a) the high density of crowds, and (b) the imprecise location information available indoors. Using a host of novel features extracted from commodity smartphone sensors, *GruMon* can detect over 80% of the groups, with 97% precision, using 10 minutes latency windows, even in venues with limited or no location information. Moreover, in venues where location information is available, *GruMon* improves the detection latency by up to 20% using semantic information and additional sensors to complement traditional spatio-temporal clustering approaches. We evaluated *GruMon* on data collected from 258 shopping episodes from 154 real participants, in two large shopping complexes in Korea and Singapore. We also tested *GruMon* on a large-scale dataset from an international airport (containing ≈37K+ unlabelled location traces per day) and a live deployment at our university, and showed both *GruMon's* potential performance at scale and various scalability challenges for real-world dense environment deployments.

*This work was done when the author was a research engineer at Singapore Management University.

SenSys'14, November 3–6, 2014, Memphis, TN, USA.

http://dx.doi.org/10.1145/2668332.2668340

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]; J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms

Measurement, Human Factors, Experimentation

Keywords

context monitoring, social groups, smartphone sensors, indoor localization, clustering

1 Introduction

It has become increasingly important for retailers and businesses to be able to provide context-specific incentive to potential customers. Indeed, Gartner predicts that context-aware technologies are slated to affect $96 billion of annual consumer spending by the end of 2015 with 40% of the world's smartphone users opting in to context service providers to track their activities [1]. However, determining the right incentive to provide frequently requires knowledge of the group that the customer is traveling in. For example, larger groups would be more attracted to "Buy X get Y free" promotions over individuals.

Detecting groups requires solving two distinct yet related problems: (a) determining which individuals in a specified location are traveling together; i.e., they are in a group, and (b) identifying what the relationship is between the individuals in a group. i.e., what type (friend, family, etc.) of group are they. In this paper, we focus on providing an accurate and fast method for identifying which individuals are traveling together (known henceforth as the "group detection" problem). We defer identifying relationships to future work.

Group detection, at a very high level, requires analysis of the trajectories of each individual and then identifying the individuals whose trajectories are so similar that they must be in the same group (with a low error probability). This type of trajectory analysis [2, 3] has been performed in other domains such as vehicular networks and animal migration

patterns. However, there has been little research in identifying human groups in crowded urban environments (or even in profiling the characteristics of human group movement in such environments), such as shopping malls and airports - environments that could greatly benefit from better customer incentive mechanisms.

Unfortunately, group detection in crowded urban environments is non-trivial due to two key challenges: (1) the density ensures that at any location, a large number of people are moving together – intentionally or otherwise, (2) the location tracking systems in many indoor venues tend to be either non-existent or provide low accuracy (due to the crowd density and other reasons). The first challenge makes it difficult to build a low latency group detection system as many individuals will be co-located for prolonged periods of time. The second challenge implies that group detection cannot rely entirely on the availability of accurate location data.

In this paper, we present a solution, called *GruMon*, that overcomes both challenges by fusing location data (of different levels of accuracy) with additional data such as semantic labels and smartphone sensor data from the accelerometer, compass and barometer. In particular, we use the correlations between the sensor and semantic readings of different individuals to determine if they are in the same group. To test *GruMon*, we collected data, using real participants, from two different shopping malls, CoEX mall in Korea and Plaza Singapura in Singapore. In addition, we also tested *GruMon* using a 3 day dataset from a large international airport (containing 37K+ unlabeled location traces per day). Our results showed that *GruMon* is able to (a) detect over 80% of the known groups with 97% precision, within 10 minutes of observing a group of individuals, even in venues with poor or even no location data. (b) In venues where location information is available, *GruMon* improves the precision of traditional trajectory tracking algorithms through the use of semantic labels by up to *22%*, and (c) the use of inertial sensors with location data allows *GruMon* to improve the recall rate of group detection by up to *20%*. Moreover, based on testing *GruMon* over the airport dataset, we showed *GruMon's* potential performance at scale and discuss diverse scalability issues. Additionally, a live deployment of *GruMon* has been operational on our university campus and we report preliminary observations from the collected dataset.

The main contributions of our paper are threefold:

[a] Identify challenges of group detection: We provide a comprehensive overview of existing group detection methodologies and identify that fast and accurate group detection is not simple in urban spaces. We empirically show that traditional spatio-temporal or Bluetooth proximity based group-detection approaches do not work well in dense urban environments due to (a) unavailability of accurate location information, and (b) very high crowd densities.

[b] Fast and accurate group detection method: We design and implement a set of heuristics that achieve fast and accurate group detection in dense and complex urban spaces. In addition, we devise micro-activity correlation techniques for deployment environments that lack localization infrastructure. Finally, we improve both the precision and latency of traditional spatio-temporal clustering methods, using additional sensor information.

[c] Experiments in dense indoor urban spaces: We test our techniques at two dense indoor shopping complexes: *CoEX in Korea* and *Plaza Singapura in Singapore*, which both have 200+ stores each, and 140K and 56K visitors per day, respectively. Using the data collected from 258 separate shopping episodes (with 178 of the episodes involving groups of size 2 to 5+) from 154 distinct individuals, we show that *GruMon* achieves high accuracy while keeping latency and power consumption low. Note: To reduce experimental bias, we recruited regular mall visitors who were asked to visit the mall when they wished to do so and with their preferred companions, at their will. We find many examples of natural behavior; for example, in many cases, the groups did not spend all their time together (i.e., they exhibited individual behavior for some period of time). Moreover, we also test *GruMon* with a large-scale unlabeled location dataset from a commercial airport, to examine possible practical challenges that need to be addressed, for *GruMon* to be deployed at scale.

2 Motivating Scenarios

Detecting groups in multi-functional urban spaces can be of value to both venue operators as well as individual consumers. The following are some motivating application scenarios for group detection in shopping complexes.

Proactive group-aware promotions: Mall operators have traditionally used the marketing strategy of group or bulk promotions and discounts to influence customers' intentions and purchase decisions. Automated group detection can make such strategies practical. For example, a "buy 2 ice-creams, get 1 free" promotion may seem more attractive to a group of three friends than to an individual. The ability to detect groups can also assist in forming a larger flash group with the goal of availing bulk promotions together. For example, a group of three may be recommended to a couple, if both groups are waiting outside a cinema hall, to avail a promotion such as "Buy 4 movie tickets and get 1 free".

Mall resource planning: Group information could help significantly in resource planning and obviating related customer pain points during shopping excursions. For example, a large group, without a-priori reservations, usually has difficulties in finding tables where they can have a meal together. Identifying such large groups and also the current resource situation of the retailers could help customers find appropriate places to go to as well as help retailers not waste their resources. Accumulative knowledge on groups and their visit history could also help improve the mall layout.

Since lots of modern airports function as large retail complexes, they can also hugely benefit from the previous scenarios. Also, accumulative knowledge on groups can be especially useful in airports, for instance, for security surveillance and taxi dispatching.

3 GruMon Design

3.1 Scenario Requirements

From the practical usage scenarios, we identified four major requirements for a group detection system. *GruMon* satisfies all four requirements (details in Table 1) listed below:

Expt. Description	Evaluation Results
High Detection Accuracy	
Plaza Singapura Mall	Section 6.2, 6.3.1
CoEx Mall	Section 6.2, 6.3.2
Low Detection Latency	
All Venues	Section 6.4
Support Heterogeneous Environments	
Store-Level Locations (CoEx)	Section 6.3.2
Store-Level + Sensor Data (CoEx)	Section 6.3.2
Only Sensor Data (Plaza Singapura)	Section 6.3.1
Dense Location Traces (Airport)	Section 7
Energy-Efficient Data Collection	
All Venues	Section 6.5

Table 1. Summary of *GruMon*'s effectiveness

1. High Detection Accuracy. First of all, the system should detect groups accurately (e.g. precision values of $>$ 90%) to prevent group-tailored promotions or flash recommendations from spamming ineligible single users (which reduces customer satisfaction).

2. Low Detection Latency. The system must detect groups as quickly as possible (e.g. within 5 to 10 minutes). Otherwise, group-based promotions and recommendations may not be effective as the groups are detected after they have already left the "interesting" areas. However, achieving both low latency and high accuracy is not easy as high accuracy requires observing more data (over a longer time window) while low latency requires fast decision.

3. Support Heterogeneous Environments. For *GruMon* to be easily deployable, it should require minimal changes for each new environment. However, this is not easy in practice as each environment can provide a different set of data with different fidelities. For example, the Singapore mall was observed to have a very inaccurate location system while the Korean mall and the airport have accurate location systems. Hence, where necessary, to achieve sufficient accuracy, *GruMon* must be able to combine location with semantic data and/or smartphone sensing data.

4. Energy-Efficient Data Collection. The target scenario applications are likely to run in the background to deliver proactive suggestions. Thus, it is especially important that any data collection must be done in energy-efficient ways to avoid unnacceptably high energy drains.

3.2 Test Venues

To effectively design and test *GruMon*, we targeted three heterogeneous urban venues where *GruMon* can be tested with different inputs in terms of data availability, fidelity, and scale. The three venues included two gigantic shopping complexes (a single-storey shopping mall called CoEX in Seoul and a 9-storey shopping mall called Plaza Singapura in Singapore) and a large international airport. Hereafter, we refer to these venues as $Mall_1$, $Mall_2$, and *Airport*, respectively. From each venue, we collected different types of sensing and location data from both groups and individuals, and utilised them for the design and evaluation of our techniques. Table 2 summarises the venues and data used.

At all three venues, we collected heterogeneous location data at different scales and fidelity (accuracy and update rates). At both malls, we leveraged existing client-side Wi-Fi location systems (that required an application to be running) with $Mall_1$ providing store-level locations with 10 second update rates while $Mall_2$ only provided highly inaccurate, almost unusable locations (due to a big atrium in the center, few APs, and large crowds). On the other hand, we managed to obtain the server-side (from the APs directly) location traces for the *Airport*. This allowed us to collect medium accuracy (of between 15 to 25 metres with variable update rates of between 10 seconds to 12 minutes) location information for every device connected to the Wi-Fi network at *Airport*.

Since we needed to have an application running at the malls (for client-side location tracking), we also decided to collect various types of sensing data to improve the location tracking accuracy of each venue — low rate accelerometer data at $Mall_1$ (which had reasonable location accuracy already) and richer sensing data at $Mall_2$ (which had poor location accuracy). However, this was not possible at *Airport* as our data collection was directly from the back-end APs. Overall, both mall datasets were medium scale (O(100) traces each) with labelled ground truth information while the *Airport* dataset was much larger (O(10,000) traces) without ground truth information. We explain the details of how we collected each dataset next.

Characteristics	$Mall_1$	$Mall_2$	*Airport*
Total # of traces	183	75	37K+
No. of groups	47	26	N/A(unlabelled)
No. of individuals	70	-	N/A(unlabelled)
Actual Group sizes			
2 Members	41	12	
3 Members	4	9	
4 Members	1	1	
5+ Members	1	2	
Collection period	Oct'11 to Mar'12	Oct'13 to Nov'13	3rd Mar'14 to 5th Mar'14
Collected Data			
Location data	WiFi (every 10s)	WiFi (every 50ms)	WiFi (10s – 12min)
Inertial sensing	accel@5Hz	accel@100Hz compass@100Hz baro@25Hz	N/A

Table 2. Dataset description: All participants were members of the general public

[1] Labelled datasets from shopping malls: We collected datasets labelled with ground-truth group information from the two malls, by recruiting participants. For $Mall_1$, each participant was asked to install and run our data collection application on their own phones during the entire time of the mall visit. For $Mall_2$, each participant was asked to carry a provided Samsung Galaxy S III phone that runs our data collection software. After the experiment, the participants were asked to upload the dataset they collected, and to complete a survey to specify their demographics (age, gender etc.) and who they visited the mall with (to ascertain ground truth of the groups). We compensated participants with monetary incentives of 20,000 KRW for $Mall_1$ and 20 SGD for $Mall_2$ (equivalent to 18 USD and 16 USD), respectively.

We used the following steps to reduce experimental bias. First, previously unknown participants were recruited via social media campaigns, and they were not told the aim of the experiment. Second, they were asked to go to the mall on any date and time that they were comfortable with. Third, they were asked to bring zero, one, or more acquaintances of their own choice (group size distribution shown in Table 2). Note, we also collected data from all accompanying acquaintances. Fourth, they were not provided with any specific instructions about the mall visit. They were allowed to do anything and everything they so desired at the mall (visit any store, spend any amount of time in any location, separate from their acquaintances, carry the phones in any way, etc.).

We acknowledge that other bias could still exist even after our careful experiment design. High degree of user diversity and scale make it almost impractical to remove biases completely. However, we believe GruMon takes a meaningful step towards group detection in heterogeneous urban spaces. It adopts a number of robust features based on sensor data as well as location data. Such features can be readily tuned and combined to fit the characteristics of various venues and their visitors.

We analysed the data after all the experiments had been run and we found many examples of natural behaviour (lingering in stores, separating from their acquaintances, etc.). We observed that at least 13% of the groups at $Mall_1$ and 54% of the groups at $Mall_2$ exhibited some amount of dispersion behaviour (where parts of the group separated and had their own independent non-group related motion patterns for some period of time). We computed the dispersion duration as lasting, on average, 43% of the total mall visit duration at $Mall_1$ and 28% of the total mall visit duration at $Mall_2$. Overall, this amount of variability gives us better confidence that our user-study collected dataset is representative of real group behaviour.

Note: In this data-driven design, evaluation of accuracy in the presence of false positive data is a critical necessity. In spite of collecting data from a fairly large population of 154 distinct individuals in two urban mall settings, it was still logistically very difficult to conduct *simultaneous* experiments with several groups and individuals at a particular venue. This made it challenging to evaluate the accuracy of our group detection methods, in the presence of other groups and individuals present at the same venue at the same time. To handle this shortcoming, we mix all the data from a particular venue together so that the traces of all the individuals and groups in our data set (for that location) start at the same time while ensuring that the time order among group members is preserved. By doing this, we created a high-fidelity emulated dataset that has all groups and individuals at a particular venue starting their experiments simultaneously. Note: this actually biases against *GruMon* as we could be introducing more noise into the trace.

[2] Unlabelled dataset from an airport: The *Airport* dataset includes location updates from 37,000 devices (per day) in an airport terminal for three different days in March 2014. This data set is much larger than the mall data as the locations are obtained directly from the airport Wi-Fi access points; thus eliminating the need for any client-side data collection applications. However, because the dataset was automatically collected from the infrastructure and not from specific client devices, it is unlabelled; i.e., unlike the shopping malls, we do not have ground truth about the real groups. However, even without ground truth, the large amount of data helps us in understanding the practical challenges that *GruMon* might face when deployed at scale.

4 Feasibility of Previous Approaches

There have been a number of research work in multiple research domains for discovery of groups. Each work uses different definitions and assumptions about groups, and has attempted to achieve different system goals. In this section, we describe how we use and improve on prior works in our solution.

4.1 Spatio-temporal Clustering

In multiple research domains, including database and data mining, a lot of effort was spent to detect travelling companions such as flocking animals and humans based on their spatial trajectories over time [2, 4, 5, 3, 6, 7]. Acknowledging the reality that companions do not always stick together, another line of work discovers groups that form and disperse multiple times as time lapses [5, 3]. More recently, researchers have tried to study grouping behaviors in shoppers and students [6] and mobility patterns such as leading and following in humans [7].

Compared to *GruMon*, these works focus on the accuracy of detecting travelling companions through offline processing over long-term location traces; real-time processing and related metrics such as latency and energy-efficiency are not their major concerns. Recently, some researchers proposed a framework to detect groups over streaming location data [3]. However, it also does not consider crowded, dense scenarios in which low-latency detection is challenging due to the presence of multiple short indistinguishable trajectories. There have been some initial research in group detection for indoor spaces [6, 7]; however, they also assume that location information is available and given in the form of well-understood coordinates, which are not true for a number of indoor urban spaces.

Applying Spatio-temporal clustering. We examined the applicability of spatio-temporal clustering to our dataset and describe several practical challenges discovered.

[a] Inaccuracy of indoor location information: Inaccurate indoor location can make spatio-temporal clustering almost impossible. In our case, $Mall_2$ (with a low accuracy Wi-Fi location system) demonstrates this problem clearly. For example, Figure 1 shows the location landmarks returned by the localisation system of $Mall_2$ over time, for three members of the same group standing together in level 3 of the mall. Note that the y-axis is the landmark ID in treble figures; the hundredth place of the landmark values indicates the floor level and the last two figures indicate the landmark within the floor. From the figure, we observe that the three group members are localised to the same place only 25% of the times (marked by rectangles). They are localised to two, sometimes three different floors, and even up to five stories apart. This causes spatio-temporal clustering approaches to classify them as strangers. Such deviation among group

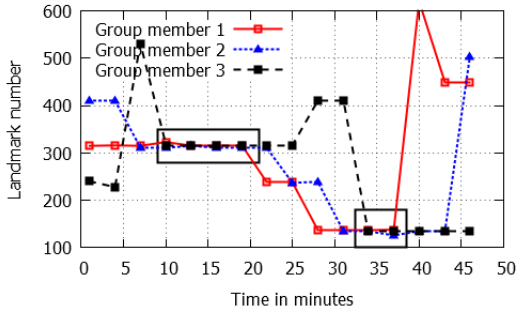

Figure 1. Example of location deviation within group

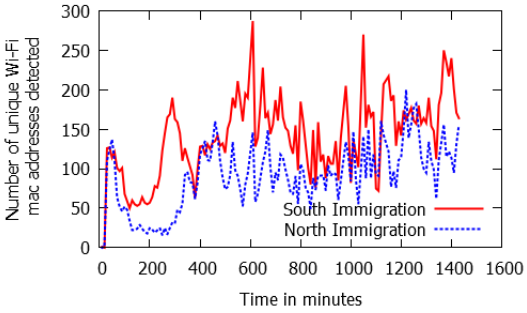

Figure 2. Co-located individuals at airport immigration

members is repeatedly observed in this dataset.

Such inaccuracy would be quite common in venues like $Mall_2$ for the following reasons: (a) intractable RF fingerprinting due to the large mall size, (b) temporal fluctuations in RSSI due to dynamic crowds, (c) few, sparsely-located APs[1], and (d) a mall layout with open atrium, all of which in combination causes traditional fingerprint matching algorithms like Radar [12] and Horus [13] to perform poorly.

[b] Precision and latency issues in dense environments: Even if accurate location is available, spatio-temporal analysis might give lots of false positives in group detection, if the crowd densities are high. Random individuals might be at the same location, at the same time, for elongated periods. Figure 2 shows the number of unique Wi-Fi MAC addresses detected at two immigration gates in the airport dataset, over 10 minute time windows, averaged over the 3 days of data. We observe that about 100 to 200 different individuals are co-located at each immigration gate, within the same 10 minute time windows. This indicates that even with medium fidelity location information (*Airport* accuracy is about 15-25 meters), it may not be possible to detect groups quickly if people stay in the same general area.

We understand that the spatio-temporal approach will be more powerful when highly accurate and precise location data are available at low latency. However, note that GruMon targets diverse urban venues in practice with different levels of localization capabilities.

4.2 Bluetooth-based Proximity Detection

Another line of research aims to understand social interactions such as meetings and watercooler conversations, in workplace and campus environments, using close co-location or proximity as an high-level indicator of such interactions. In such confined, non-crowded environments, Bluetooth scans have been used to detect nearby persons using a list of scanned devices over time [14, 15, 16].

Applicability to our environments. We found several potential limitations in applying Bluetooth scanning techniques to our environments. First, unlike workplace environments where people know and trust each other, in densely crowded malls and airports, with strangers all around, people might not allow Bluetooth discovery to be enabled to prevent spams and other security threats.

Second, we examined if Bluetooth-based techniques can scale to a crowded situation where possibly tens or hundreds of devices could be co-located within a radius of 30 feet which is the typical communication range of Bluetooth radios. We instrumented 16 different phones, running on Android 4.1 or higher, to *periodically and simultaneously scan every 15 seconds* to study the time taken for discovering neighbours, the proportion of neighbors discovered and the energy consumed. We make the following observations:

[1] By increasing the number of devices up to 15, the time taken to discover all possible neighbours increases to 10 seconds while the number of discovered devices drops to 80%. This indicates that having plenty of devices nearby (e.g., in a crowded mall) can result in (1) scans requiring longer time, (2) partial discovery of neighbors, affecting detection accuracy and latency of existing algorithms.

[2] More nearby devices also increases the power consumption for scanning: We measured the power consumption of the device while increasing the number of nearby devices that are also scanning simultaneously. Each scan cycle [17] consists of a inquiry scan ($\approx$ 160 mW), page scan ($\approx$ 210 mW) and no-scan ($\approx$ 110 mW, i.e. the time lapse between the end of a page scan and the next request for inquiry). We observed that with an increase of simultaneously scanning devices, the time taken for page scan increases resulting in increasing power consumption on average.

Bluetooth Low Energy (BLE) is designed to operate with lower power consumption, and some recent works like [18] have used BLE as a low power machanism to discover neighbors. However, on the smartphones we tested, due to backward compatibility reasons, BLE shares the same radio and antenna as regular Bluetooth resulting in minimal power savings. In our experiments with a Samsung Galaxy S4 running on Android 4.3, we observed a saving of only 15 - 20 mW with a low-energy scan.

Also, commodity handsets do not offer full BLE support yet. For example, Android smartphones (as of Android 4.4) can only function as clients that discover other devices — thus, they cannot be discovered by other BLE devices. We plan to investigate the scope and limitations of using BLE further, when suitable devices and APIs become available.

[1]Increase in localization accuracy with increase in AP density has been discussed in design manuals of indoor location systems [8, 9] and also empirically measured [10, 11]. However, the empirical measurements have been in outdoor or in controlled lab indoor environments. Understanding the localization performance with varying AP density and positioning, for challenging indoor venues like shopping malls and airports, is an orthogonal research direction.

4.3 Other Approaches

Acoustic sensing-based conversation group detection: Recently, approaches using conversations to detect daily social interactions [19, 20] have been proposed. They develop real-time speaker identification techniques using phone-embedded microphones in order to detect conversation groups [20] and infer more detailed contexts during conversations [19]. In the contexts of crowded urban spaces, however, it is challenging to simply leverage these techniques for group detection. This is because conversations within a shopping group occur in a much more sporadic manner, and is inter-weaved among multiple groups. For example, in clothing stores, people stand and move around freely to look at different items, and multiple groups talk together in a shared space. Moreover, in many indoor areas such as food court and event halls, the noise level might be too high due to dense crowds as well as temporary events, which makes sound-based approaches less robust.

Online social network-based social group detection: The use of Social Network Services (SNS) to detect social groups has been well-studied [21, 22]. Although well-studied, networked friends usually belongs to a much wider geographical span (sometimes across continents) that makes social network groups less relevant for in-person interactions such as shopping together. In addition, SNS groups frequently include people whom a) the user may not know, and b) do not spend time with doing daily activities such as shopping. However, while SNS data may not be useful to detect when the user is in a group (as the SNS data changes at much slower time scales than interactions in an indoor space), it might be useful to detect the relationship between a user and his group members (once a group and its members have been discovered). We plan to quantify the benefits of augmenting *GruMon* with SNS data in future work.

5 How GruMon Works

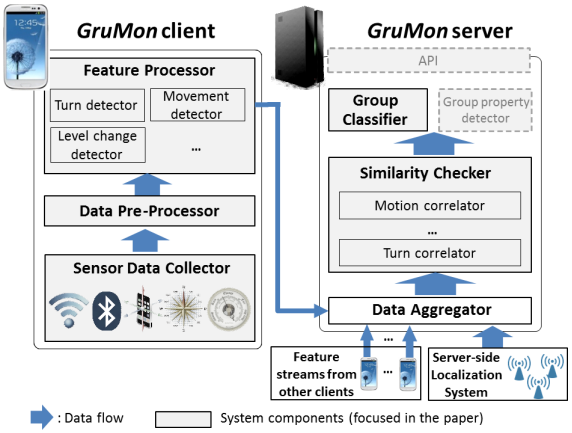

Figure 3. GruMon architecture

In this section, we describe how *GruMon* works in detail. The key intuition is that members of the same group exhibit similar micro-activity or mobility patterns (collected from diverse phone-embedded sensors), which distinguishes them from people who are in different groups, or from random individuals. Figure 3 shows the *GruMon* system architecture. It comprises a client that runs on multiple individual smartphones to collect diverse sensor data and extract relevant features and a server that detects groups by processing the data from the clients (or other data sources such as server-side location data from *Airport*); client may not be necessary when data is obtained from other sources.

Figure 4 shows the overall workflow of our group detection method. It consists of four major steps as follows:

Step [1]: Each *GruMon* client detects diverse micro-activity and location features using phone-embedded sensors. The features calculated on the clients are sent to *GruMon* server. (See Section 5.1 for details.)

Step [2]: The server first computes similarities between each pair of *GruMon* clients, using cross-correlation of the time-series of the computed features. (See Section 5.2.)

Step [3]: The server then passes the pairwise similarities through a supervised binary SVM classifier, which classifies each pairwise edge as positive or *group edge* vs. negative or *non-group edge*, based on a pre-trained classification model. (See Section 5.3.)

Step [4]: Finally, the server runs a clustering algorithm on the positive edges returned by the binary classifier, to output sets of individual clients as groups. (See Section 5.4.)

The main novelty lies in building an end-to-end working system combining all four steps to illustrate *GruMon* 's potential and evaluate its performance. Especially, we newly explore novel features for group detection for **Step[1]**. For other steps, we carefully adopted and tuned existing classification and clustering algorithms that suit the unique characteristics of our group detection problem.

Note that *GruMon* works on the assumption that all group members to be detected communicate with the *GruMon* server. If some members of a group communicate, while others do not, only partial groups will be detected. It is beyond the scope of this paper to address the possible security concerns, i.e., having the venue owner as a trusted entity to monitor willing visitors insides its premises.

5.1 Feature Computation

The features for group detection should be similar within a group and discriminative otherwise. The first step of group detection is to extract comparable features from raw sensor data streams. To calculate a feature, each *GruMon* client first accumulates raw sensor readings for a designated time window. Over the window, it pre-processes the sensor data to check the data integrity, and applies a corresponding feature detection method. Such features includes a location or spatial feature (denoted by F_S), motion features (F_M), turn features (F_C), and level change features (F_L). We next explain why these features are promising for detecting groups, and also describe how to calculate the features; note that feature calculation is only briefly explained due to space limitations.

[a] Spatial features (F_S): A distinctive feature of groups moving in an urban space, is their transition patterns from one semantic section to another. E.g., many strangers can be present at one semantic location, leading to false positives using spatio-temporal analysis, but only real group members would tend to make simultaneous or coordinated transitions between sections, repeated multiple times as the group

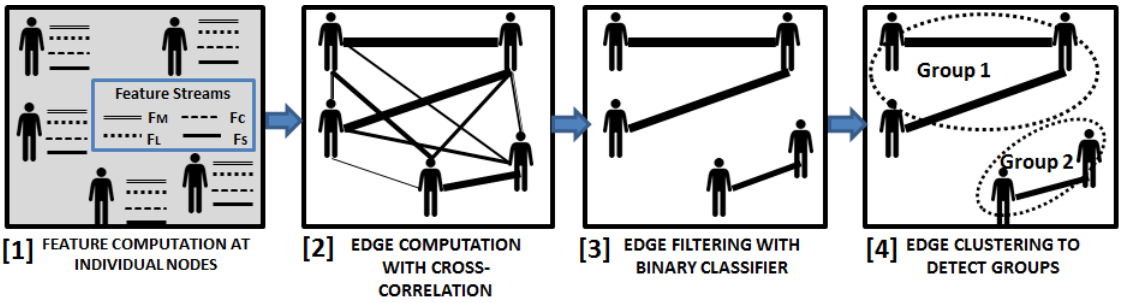

Figure 4. Workflow of group detection

moves through a venue. This helps filter spurious groups and overcome the low precision of spatio-temporal clustering.

To understand the viability of using semantic transitions, we observed the store transition patterns of the 47 groups in our $Mall_1$ dataset. Table 3 and Figure 5 show the results. From this, we noted that 1) group members made coordinated transitions, 2) people made multiple transitions during their stay, and 3) more transitions are made at the start of their visits. Table 3 shows the percentage frequency of groups versus the number of unique stores visited. From the table, 90% of the groups visited a minimum of five unique stores during their mall visit. Figure 5 shows the cumulative percentage of time, in minutes, taken to make consecutive store transitions. From the figure, nearly 60% of the groups (the vertical line in the figure) made their first store transitions within the first ten minutes of their mall visit. Overall, semantic transitions appear to be able to distinguish groups versus non-groups.

No. of stores	< 3	3-4	5-6	7-8	9-10	>10	Total
Frequency	0	10	32	22	28	8	100

Table 3. Percentage frequency of store visits by groups

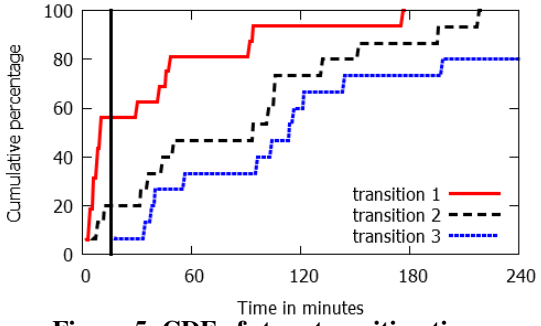

Figure 5. CDF of store transition times

Computing F_S: The clients periodically report the observed Wi-Fi access points (with associated received signal strength (RSS)) to the server. The server converts these RSS measurements to locations using a pre-collected fingerprint map and hyperbolic Radar [12]. For the *Airport* dataset, the server uses similar techniques to convert the RSS measurements obtained from the server-side APs to physical locations. The server then extracts semantic transitions and other spatial features from these computed location information. Movement from a common source location to a common destination location, by different individuals, within one minute interval of each other, is considered as a coordinated semantic transition in our implementation.

[b] Motion features (F_M): For group members, their mobility patterns (e.g., moving vs. stationary) tend to be matched as they naturally walk together at the similar pace. Figure 6 shows two friends' accelerometer magnitude on the y-axis versus time on the x-axis as they walk from the entrance of a shopping mall (on level 1) to a movie theater on level 7. The similarity in their mobility pattern is apparent where they both walk from the entrance to the first escalator, and then stop and walk intermittently as they take 6 flights of escalators to the seventh level, walk to the ticket counter, stand at the counter to collect tickets, walk to the movie theater and finally sit down to watch the movie.

Computing F_M: From the accelerometer streams, we extract motion features that indicate the stationary versus motion states of the individual. This feature is intentionally kept very simple to avoid propagating errors caused by the phones' placement position and device heterogeneity (both effects are known to affect accelerometer accuracy greatly) to the next stages of group detection. More specifically, the states we require can be inferred using well-known activity recognition tools. A decision-tree with features of (a) standard deviation of magnitude, and (b) standard deviation of difference in magnitude of consecutive samples, gives 98% accuracy to label 5-seconds accelerometer data sampled at 100 Hz frequency correctly as 'stationary' or 'motion'. Having a *small* time window (5s) for the motion feature is necessary to capture the intermittent stop and go motion of people.

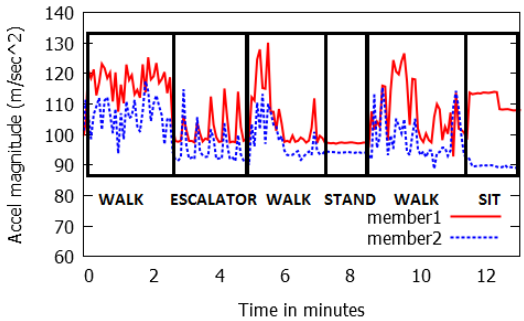

Figure 6. Motion similarity

[c] Turn features (F_C): Group members take certain turns together while walking based on the geometry of the urban space. Figure 7 shows the compass values (in degrees from north) on the y-axis versus time on the x-axis for three mem-

bers of a group in $Mall_2$. The simultaneous change in angles marked by four different rectangles, where the members take coordinated turns, can be observed from the figure.

An important point to note is that the absolute value of the angle is different across group members as this depends on the phone orientation (smartphone compasses measure the angle the y-axis of the phone makes with the North). Thus members with phones vertically placed in pockets or horizontally placed in hands or bags, will have different angle readings. However, even though the absolute values differ, the angle changes are similar across group members. Thus detecting the event of *change in angle* can be made orientation independent.

The comparison of turns *while walking continuously* is important, as we empirically see people take multiple arbitrary turns to look at things or to interact with other group members. A simple heuristic of only considering readings taken during continuous walking helps filter out some of these spurious turns.

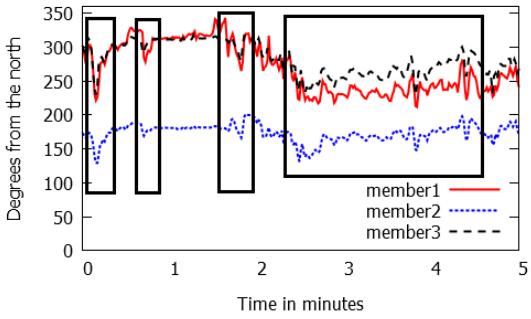

Figure 7. Turn similarity

Computing F_C: From compass data, we detect *turns* made by individuals. In more detail, turns are inferred by (a) computing the difference between smoothed compass readings N samples apart, and (b) detecting turns when the difference exceeds a threshold Th_C. In our implementation, we sampled the compass data at 100 Hz, smoothed the data stream with a one-second moving average window, and empirically fixed $N = 200$ and $Th_C = 15$. These settings were able to detect turns $> 45^\circ$ with 87% accuracy at $Mall_2$.

[d] Level features (F_L): Group members tend to transit between different floor levels in a coordinated way. In a multi-level urban space, these coordinated level transitions can be detected using smartphone barometers. Figure 8 shows the barometer readings on the y-axis versus time on the x-axis for four group members in $Mall_2$. The coordinated level changes among the group members, marked by a rectangle, are visually apparent from the figure.

As also shown by Kartik et. al [23], different phone calibrations result in different absolute pressure values across phones (for the same floor transitions). However, even though the absolute values differ, the *relative changes* in pressure across phones are coordinated. In addition, barometric readings are independent of phone orientation and user activity (when there is no change in their level) making it a very reliable level transition predictor.

Computing F_L: Similar to compass readings, the barometer readings are smoothed using a moving average. Level

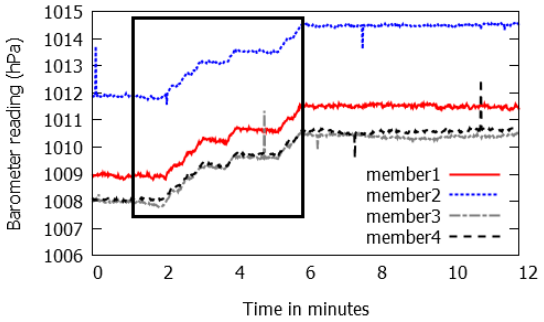

Figure 8. Level transition similarity

transitions are detected by (a) computing difference between smoothed readings N samples apart, and (b) detecting floor changes if the difference exceeds a threshold Th_L. In our implementation, we smoothed the barometer readings sampled at 25 Hz, by a moving average of 1 second, and empirically fixed $N = 250$ and $Th_L = 0.2$ (based on results from Kartik et. al [23]). Due to the simplicity of the feature and the differentiable barometer readings at different levels, we achieved 100% level change detection accuracy for our $Mall_2$ dataset.

It might be more sensible to collect mobility features only when store transitions are made by visitors. This is because, group members typically diverge and pursue their individual interests within individual stores. In this regard, there could be scope for further improvement since the current *GruMon* prototype calculates these features for user traces over the entire mall visit duration. Note, however, that we show *GruMon*'s current prototype still performs well in Section 6.

5.2 Edge Computation with Cross-correlation

This step computes the correlation between two feature streams belonging to a pair of individuals. As a first step, it first converts streams of features (F_M, F_C, F_L, F_S) updated from clients into a time series of binary values (indicating events detected from sensor data). Figure 9 shows an example of this time-series data. In more detail, binary event values are defined as (a) 0 for stationary and 1 for motion for the F_M feature, (b) 0 for no-turns and 1 for turns for the F_C feature, and (c) 0 for no level change and 1 for level change for the F_L feature. Note that each binary label is based on the features that are computed over 5s of buffered sensor data.

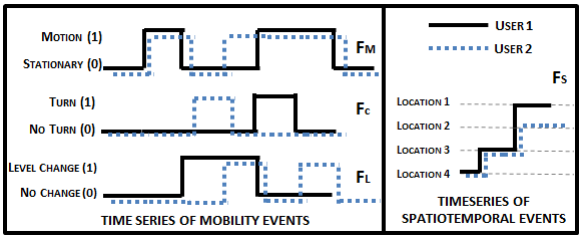

Figure 9. Pairwise feature correlation

Next, for all possible individual pair combinations, the *GruMon* server computes cross-correlations [24] between the converted time-series data. We carefully determine the weight of each feature based on our dataset-specific empirical studies. E.g., we assign higher weights for coordinated level changes, and lower weights for being stationary together. This step *boosts* the importance of more reliable

sensor events (like level transition using the stable barometer sensor) and *suppresses* the less reliable ones (like turns which can be spuriously measured by the noisy compass sensor). Similar to mobility events, we calculate correlations for the time-series of features F_S (store visits and transitions).

Note that in our current implementation, the number of correlation pairs to evaluate is quadratic in the number of individuals. However, we can reduce this number significantly by applying a computationally cheap but rough clustering of people; for example, comparing only the pairs of individuals within a certain location radius, when even reasonably coarse-grained location information is available.

Each individual can be assumed as a graph node, as shown in Figure 4[2]. At the end of this cross-correlation step, we assign an edge between each pair of nodes, where the edge weight is determined by the cross-correlation value. Higher weights signify more similarity between two individuals, as shown by thick lines in Figure 4[2], while the thin lines in the figure represent lower pairwise similarity.

5.3 Edge Filtering with Binary Classifier

This step pre-selects the pairs of individuals that are likely to be in the same group before applying a clustering algorithm to determine the final groups.

The weighted graph created in the previous step using pairwise cross-correlation, acts as an input to this step. Each weighted edge in the input graph, is classified by a binary SVM classifier, as a positive edge or *group edge* vs. a negative edge or *non-group edge*. As shown in Figure 4[3], the *non-group* edges are filtered out and only the *group* edges are retained. Thus the graph after this edge-filtering step, contains much fewer edges compared to the input graph, which had all possible pairwise edges.

The correlation values between pairs of individuals evolve over time. As members of the same group spend more time together, their pairwise correlation values gradually become more positive. For unrelated individuals, over time, the values gradually become more negative. Thus, as the feature values change, the SVM classification models are different for different time windows. In Section 6, we experiment with time windows of 5*mins*, 10*mins*, 15*mins* ... to 30*mins*. This time window can be configured depending on the accuracy and latency tradeoff that applications need to make.

The binary SVM classifier is trained to perform edge classification. Correlation values between pairs of individuals, who are part of same group, are used as positive training samples. Similarly, pairwise correlation values from random individuals or people belonging to different groups, are used as negative training samples. Inspite of this training overhead, this step dramatically enhances *GruMon's* precision compared to only running unsupervised edge clustering on all possible pairwise relations. We will show this performance enhancement empirically, in Section 6.

5.4 Edge Clustering to Detect Groups

The final step clusters the edges retained in the previous step, to output sets of individuals as groups. We utilized the Markov Cluster algorithm (MCL) [25], instead of other possibilities, for the following reasons: First, MCL does not need the number of clusters as an input parameter, unlike some other graph clustering algorithms [26]. Second, MCL is well suited for undirected input graphs which matches our graphs as the relationship between pairs of grouped individuals is symmetrical (and can thus be treated as undirected). Third, MCL interprets weights of edges as similarities, which is true for our input graph. Finally, MCL performs well for graphs in which the diameter of the natural clusters is not too large. This holds true for our input graphs, where compared to the size of the graph, i.e. the total number of individuals present, the group sizes are typically small (2-5 people). So the diameter of the clusters, i.e. the maximum path length in a cluster is small.

Venue	Features
$Mall_1$	time spent together, number of coordinated store transitions (F_S)
$Mall_2$	motion, turn and level-change similarities (F_M, F_C, F_L)

Table 4. Features for the two shopping malls

6 Evaluation

In this section, we present experimental results for *GruMon* for the two mall datasets which had ground truth. We use the *Airport* data in Section 7 to investigate practical issues with deploying *GruMon*. To recap, $Mall_1$ has dense Wi-Fi infrastructure, giving store-level location accuracies while $Mall_2$ has poor indoor location data but a potentially richer set of sensor-driven features due to a) having nine levels causing people to frequently use escalators and elevators, resulting in significant barometer signatures, b) high crowd levels, causing people to walk in a stop and go manner avoiding collisions, resulting in significant accelerometer signatures, and c) several corridors at right angles to each other, resulting in significant compass signatures. Table 4 summarizes the available and useful features at each venue. We used diverse combinations of these features, to understand their effect on *GruMon's* performance.

For this evaluation, we split the datasets from the two mall venues into disjoint training and test sets. For each venue, a SVM classifier (used in the step 3 of *GruMon*(Section 5)) was trained on the training data and we then measured the accuracy of *GruMon* using the test data. 10 out of the 47 groups and 10 out of the 70 individuals (Table 2) formed the training data for $Mall_1$ with 6 out of the 26 groups for $Mall_2$. We repeated each experiment 100 times and picked a random set of training data for each run (to minimize bias caused by the choice of training and test data). All the results presented in the rest of this section are thus an average value across these 100 separate runs. However, in all cases, we did not split a group across test and training sets to avoid data cross-contamination (i.e., any particular group (comprising 2 or more individual traces) was either 100% in the training set or in the test set, and never in between).

6.1 Evaluation Metrics

We used accuracy, latency, and energy-efficiency as the key evaluation metrics as they are key requirements of group detection in urban spaces (as stated in Section 3).

Latency is measured as the time taken to detect groups after the group members start to report their data.

Energy-efficiency is measured as the additional smartphone power consumption, measured by the Monsoon power

monitor[3], caused by our *GruMon* client functionality.

Accuracy is measured in terms of *precision* and *recall*. We define the precision and recall of *GruMon* as follows. *GruMon* outputs multiple sets of individuals identified as groups; Compared to the ground truth groups, these sets might be exactly the same, subsets, supersets, intersections, or completely disjoint. Precision is defined as $\frac{(number of ground truth groups detected)}{(total number of groups detected)}$. To define overall recall, we first define *per-group recall* as $\frac{(size of ground truth group)-(number of detected subgroups)+1}{(size of ground truth group)}$, if the number of detected subgroups is less than the size of ground truth group, and 0 otherwise. For example, a group of size 5 will have recall of 1 if the whole group is detected, $4/5$ if the group is split into two subgroups, $3/5$ is the group is split into three subgroups, $2/5$ if the group is split into four subgroups and 0 if all five members are detected separately. The overall recall is computed as the average per-group recalls for all the ground truth groups.

We also separately evaluated the accuracy of our edge filtering method that uses SVMs. For this separate evaluation, we used the standard recall and precision definitions. There are four categories of decisions that the SVM can make: (1) *False Positive* (FP) - relationship between two non-group members labeled as group (2) *False Negative* (FN) - relationship between two group members labeled as non-group (3) *True Positive* (TP) - relationship between two group members labeled as group, and (4) *True Negative* (TN) - relationship between two non-group members labeled as non-group. If N is the number of decisions that the SVM makes, precision is defined as $\frac{N_{TP}}{N_{TP}+N_{FP}}$ and recall is $\frac{N_{TP}}{N_{TP}+N_{FN}}$. Note that precision, or the correctness of detected positive classes, increases with the decrease of false positives. Recall, or the number of positive classes detected correctly, increases as the false negatives decrease.

6.2 Overall Accuracy

Our first result is the overall accuracy of *GruMon*. Table 5 shows the recall and precision in $Mall_1$ and $Mall_2$. These values are computed with classifiers outputting results at 10 minute windows (*GruMon* will output the detected groups at t=10 min, 20 min, 30 min, ...) and all results are averaged over 100 random allocations of training and test datasets. All the available features for both malls (in Table 4) were used. The results show that *GruMon* can detect above 80% of the groups with over 90% precision for both the malls with the entire system applied (SVM-based edge filtering followed by MCL-based group clustering).

This result shows that *GruMon* can detect groups (of varying sizes) accurately (even when many groups separated for some portion of the visit) in heterogeneous environments, and can even work well, by using sensor-driven features, in environments ($Mall_2$) *without any location data!* We show which sensor-driven features were more useful in Section 6.3.1.

We then evaluated the effect of SVM-based edge filtering by running MCL without the edge filtering by directly providing all possible pairwise relations as inputs to the clustering algorithm. As indicated in bold in Table 5, the precision is very low when only MCL is used. This is mainly because no supervision is provided on the expected degree of similarity between two group members versus the degree of dissimilarity between two non-group members. Thus the clustering algorithm cannot split the large clusters into smaller real groups resulting in moderate recall (as each individual is part of some large cluster) but very poor precision. Thus the edge filtering step is necessary for better performance.

Depending on the required group detection application, *GruMon* can make tradeoffs between recall and precision, by adjusting the *SVM loss function*, which determines the sensitivity of the filtering. For example, some advertising campaigns might care less for false positives but might want to reach to a large audience. This would require high recall with allowable low precision. On the other hand, some promotion campaigns with useful coupons being given away would need very high precision to reduce wasted expenses with possible low recall requirements.

Venue	Recall SVM+MCL	Precision SVM+MCL	Recall MCL	Precision MCL
$Mall_1$	83.33	91.18	87.88	**57.44**
$Mall_2$	80.56	97.47	86.11	**42.62**

Table 5. Overall precision and recall

6.3 Effect of Different Features

In this section, we investigate the effect of using different features and their combinations on *GruMon's* performance.

6.3.1 Experiments with $Mall_2$ Dataset

$Mall_2$, with multiple available features, provides interesting avenues to explore the effect of diverse features and their combinations on the precision and recall of *GruMon*. Figure 10[2] presents the precision and recall, using 10 minute output windows, for the individual features (motion, turn and level change features), computed with accelerometer, compass and barometer sensors, respectively, across all possible sensor combinations.

The lowest ellipse in the figure highlight the low recall of the barometer as group members who do not transit levels within the first 10 minutes will give false negative errors. However, the barometer has very high precision, as (a) the sensor is quite accurate in detecting level changes, and (b) coordinated floor change is a strong indicator of group membership. On the other hand, the compass and accelerometer have lower precision as the motion versus stationary behaviors or turns might vary slightly among group members. As the compass is a noisy sensor affected by magnetic environments, it also gives lower accuracy in turn detection. Hence, its precision is the lowest (marked by the highest ellipse). Combining the features results in the best recall (accelerometer-compass combination), and the best precision (combining all three).

We also looked into the accuracy of the edge filtering stage (shown in Figure 11) and made an interesting observation about how the overall recall can decrease with pre-

[3] http://www.msoon.com/LabEquipment/PowerMonitor/

[2] For both Figures 10 and 11, we used a line graph for clearer presentation and not to show any trends.

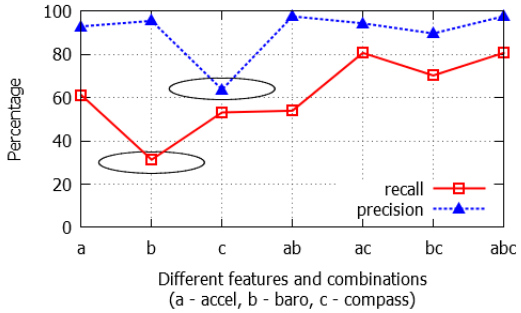

Figure 10. Overall accuracy vs. features at $Mall_2$

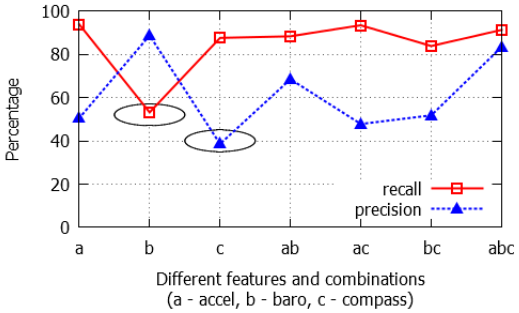

Figure 11. Edge filtering accuracy vs. features at $Mall_2$

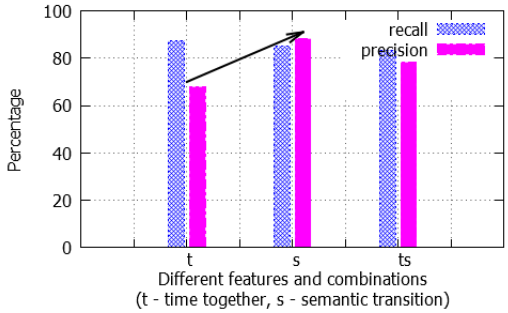

Figure 13. Accuracy vs. location features at $Mall_1$

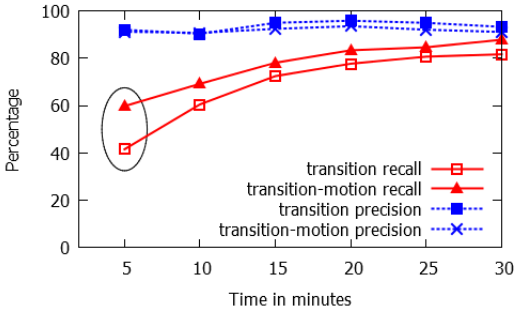

Figure 14. Recall increase with mobility features at $Mall_1$

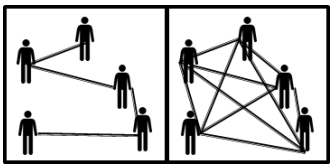

Figure 12. Sparse and dense clusters of individuals

cision increasing when using diverse feature combinations. For example, N individuals can be clustered into an extreme group where only $N-1$ edges are present among them, to the other extreme case of all $\binom{N}{2}$ possible edges present among the N individuals — with all combinations in between also possible. The two extreme cases are pictorially shown in Figure 12. Clustering nodes with less edges among themselves would give higher recall at lower precision, while nodes with more edges would have the opposite effect. This indicates that we can tradeoff between precision and recall of overall group detection, depending on application requirements, by tuning the SVM loss function of the edge filtering stage.

6.3.2 *Experiments with $Mall_1$ Dataset*

We also studied the effect of different feature combinations of coordinated store transitions and time spent together at $Mall_1$. Figure 13 shows the precision and recall using 10 minute windows for $Mall_1$. As shown by the arrow, semantic transition features achieve much higher precision compared to the time spent together feature as extra people co-located with actual groups get filtered out as the group makes coordinated semantic transitions while the extra people do not.

We also analyzed the combination of sensor-derived features and mobility features using a small subset of $Mall_1$ participants, for whom we have accelerometer data at 5Hz sampling frequency. This subset comprises of 36 people, with 19 people forming 9 groups and 17 individuals. We used 4 groups and 4 individuals as the training dataset, and repeated this experiment 50 times, each time training and testing a new SVM (with randomly selected test and training sets).

Figure 14 shows the recall and precision values with (a) transition features, and (b) transition-motion features, at different latencies. From the figure, adding the motion feature improves the recall at all latencies with only minimal precision impact. For example, recall improves by 20% at 5 minutes latency (marked with an ellipse) and 10% at 10 minutes. Some groups might not make enough semantic transitions at a particular latency to be detected by the corresponding SVM. These are groups who remain relatively close in space with other groups or unknown individuals for long times. However, some of these groups showed strong motion similarity and were detected by the SVM that uses this additional motion feature. This causes a decrease in false negatives, improving recall.

It is important to note that the 'time together' feature in Figure 13, is computed using the existing spatio-temporal clustering approaches [27, 2, 3], more specifically, the method outlined in [3]. We modified the method to adjust for the absence of accurate GPS locations and the clusters formed based on the *Haversine* distance between two GPS coordinates. Instead, each store in $Mall_1$ has been taken as a location cluster. In this light, Figure 13 and Figure 14 highlight the enhanced performance of *GruMon* over existing spatio-temporal approaches. They bring out the effects of using the novel features of coordinated transitions and mobility, in addition to spatio-temporal information.

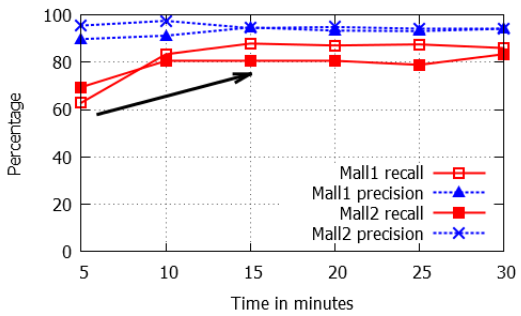

Figure 15. Accuracy vs. latency

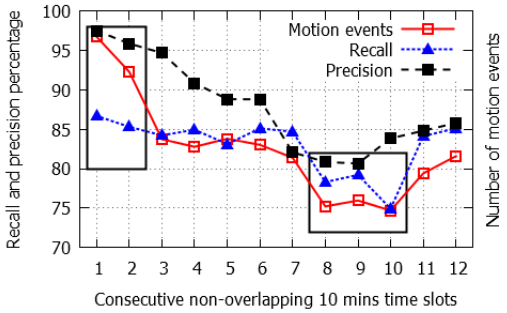

Figure 16. Accuracy vs. late start of data update at $Mall_2$

6.4 Accuracy-latency Tradeoff

Figure 15 shows the precision and recall values for $Mall_1$ and $Mall_2$ at different latencies. The precision remains almost constant with increasing time windows, but the recall increases by about 20% between 5 mins and 15 mins in both venues (marked by an arrow). This is because, over longer time periods, the accumulative nature of features such as (a) the number of $Mall_1$ stores transited together, or (b) mobility similarity between group members in $Mall_2$, would become more distinctive between groups and non-groups.

We next investigated if *"All sensing time windows are equivalent for group detection?"*. For example, would the first five-minute data trace right after the group enters the mall have better discriminative power than another five-minute data trace after the group has already spent 1 hour at the mall? This is of practical importance as *GruMon* users might start reporting their data to the server at different points during their visit to a venue.

Figure 16 shows the precision and recall detected in $Mall_2$, using different 10 minute windows from disjoint parts of the participants' traces. From the figure, the group detection recall dropped to 75% at the 9^{th} and the 10^{th} time slots, from 86% at the 1^{st} and 2^{nd} time slots (marked by rectangles). As each time slot is a consecutive 10 minute interval, this means that if the participants started uploading data at the 90^{th} minute of their stay at $Mall_2$, the recall of detecting them would be only 75%. Whereas if they had started uploading at the beginning of their mall visit, the recall would be 86%. The precision also dropped from 97% to 82% between the initial and later timeslots.

To understand the cause of this time-based variability, Figure 16 plots the avg. no. of mobility events (turns, level changes, transitions between moving and stopping) for each participant in the different 10 minutes time slots. The figure shows a strong correlation between performance metrics and mobility events as participants who remain mostly stationary (while dining, watching movies, etc.) would not generate useful sensor-driven events to compare and distinguish them.

An important experiments take-away is that the performance of the sensor-driven mobility features depends on the level of mobility, which in turn depends on the current context of the participant in the urban space. Empirically, we observed that participants had maximum mobility on entering the shopping mall possibly because (a) they had to traverse quite far to reach their intended destination from the mall entrance, or (b) they did not have a decided destination at the start of their visit and tended to explore more. Triggering the sensing to match situations where visitors are likely to be actively moving (such as when they first enter the mall together or separately, etc.) might thus increase system accuracies. Similarly, the spatio-temporal features at $Mall_1$ might be most discriminative in the early part of the mall visit, when people are undecided and explore more (generating more semantic transitions).

6.5 Power Consumption

We next investigated the power consumption of the *GruMon* client running on a Samsung Galaxy S-III. We first measured the power consumption for different sensors at various sampling frequencies. For reliable background sensor sampling in Android, the client needs to acquire a wakelock. This requires a base power consumption, even without any sensing, computation or communication, of about 94 mW on our test phone. We assume that background sensing will be common across multiple applications in the near future and thus do not include the wakelock power consumption in our results. Table 6 shows the power consumption for different sensors at different Android sensing frequencies and observed that power consumption decreases significantly at lower sampling rates. We observed that Wi-Fi scans on the client device consumes high energy; this shows that localization from the infrastructure-side could be important in terms of power consumption, which does not cause any additional battery drain on a client device.

We then investigated the impact of low sampling rates on *GruMon's* accuracy. Our micro-benchmarks with the slowest sampling rates showed that the motion and level change detection accuracies remained unaltered compared to using the fastest sampling rate. However, the turn detection accuracy dropped to 78% from 87% when using the slowest sampling rates. Even with this drop, we observed only a marginal difference (less than 1 %) in *GruMon's* accuracy. Thus, the *GruMon* client can run effectively even when sampling data at the lowest most energy-efficient rates.

Features	Fastest (100Hz)	Game (50Hz)	UI (16.7Hz)	Normal (5.5Hz)
Motion (accel)	122.79	114	84.61	29.28
Level change(baro)	41.89	41.22	30.55	12.92
Turns (compass)	164.92	135.9	86.82	35.22
Wi-Fi Scan	160.63 (scanned every 2.5 sec)			

Table 6. Power consumption (mW) vs. sampling rates

We then measured the total power consumption of the *GruMon* client which collects, processes, and transmits sen-

Sensor combination	Compute only (Wi-Fi OFF)	Compute and send (Wi-Fi ON)	Send all raw (Wi-Fi ON)
Accel	40.39	55.17	59.68
Baro	26.01	42.05	45.81
Accel+Baro	42.49	61.28	66.00
Accel+Compass	43.35	69.63	72.81
Accel+Baro+Compass	46.72	72.30	75.81

Table 7. Power consumption (mW) vs. client tasks

sor readings. Table 7 shows the results for the lowest sampling rates (as it does not affect the detection accuracy). First, turning on all the sensors and then computing corresponding features does not consume much more power compared to using just a single sensor and associated feature; e.g., in the table, the absolute power difference of the barometer is not very high. This suggests that *GruMon* can utilize all the features, without wasting too much energy, to improve detection accuracy. Second, local computation versus server offloading consumes similar amounts power. The third and fourth columns in Table 7 shows the comparison between (a) (sensing + feature computation + transmission of features over Wi-Fi) and (b) (sensing + transmission of raw data over Wi-Fi). Overall, the difference between local computation versus server offloading is less than 10%. Thus, the choice between local versus server computation can be guided by external factors such as whether the raw sensor data would be useful in the server for other applications.

7 Scaling Up GruMon

In this section, we discuss the lessons learned when we applied *GruMon* to a much denser data source. For this, we use the large-scale *Airport* dataset that comprises of location traces from 37K+ mobile devices per day (see Section 3.2 for details). For this large dataset, we were not able to collect ground truth of actual groups. However, we still obtained several useful real-world deployment insights by applying *GruMon* to this scale of data.

Groups detected in whole dataset: We first applied *GruMon* over the whole dataset and investigated the groups detected by the system. Note that unlike the $Mall_2$ dataset with store level semantic locations, the *Airport* dataset has coarse level semantic separation into two immigration gates, two departure gates, three retail sections, two skytrains and two transfer sections. So each semantic location is a large physical area, making the possible number of semantic transitions smaller (which hurts the performance of *GruMon*), delays between transitions larger, and higher population within each section. Note: there was no noticeable increase (everything finished in under a second) in the processing latency of the *GruMon* server even with the larger *Airport* dataset.

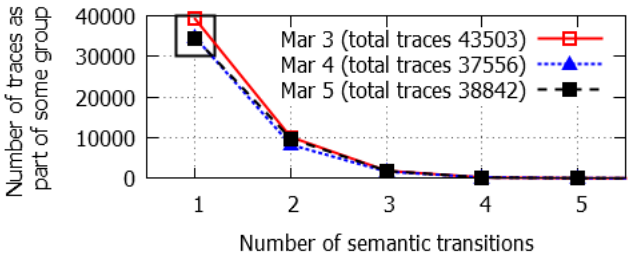

Figure 17. Number of traces detected as part of groups

Figure 17 shows the number of individuals detected as part of groups (of different sizes) on y-axis, with respect to the different threshold values to determine groups, i.e., the number of transitions made together. As marked by a rectangle, using low threshold value of one semantic transition, classifies too many individuals into groups. This indicates the low precision induced by many co-located strangers in dense environments. On the other hand, using high threshold values of 4 or more transitions detects very few groups, possibly increasing false negatives and reducing recall. Also for very high threshold values, the group detection latency increases. From the results, 2-3 semantic transitions, appear to give a good accuracy versus latency tradeoff.

Groups detected in partial dataset: To better classify *GruMon's* accuracy, we detected groups among the location traces which ended at one set of departure gates (Gates Z) and which were seen earlier elsewhere in the airport. These are likely to be departing passengers from these gates. From the online available flight schedules, we identified the period between 1130 to 1330 hours, during which three flights departed from these gates, with no flights departing either 1.5 hours before or after this time slot. Thus, it is quite likely that gate traces from this two hour time slot will contain mostly passengers who departed in one of the three flights.

Date	Detected devices	Devices in groups	Group size 2	Group size 3	Group size 4 or more
Mar 3	679	210	91	8	1
Mar 4	584	197	76	15	0
Mar 5	498	170	76	6	0

Table 8. Groups detected at Gates Z between 11:30-13:30

We extracted the traces for the selection location traces and applied *GruMon* to those traces; using 15–30 minutes together with 2 semantic transitions classifier. Table 8 shows the number of devices detected by the location system (column 2), devices detected by *GruMon* as part of some group (column 3), and the number of groups of different sizes (columns 4 to 6). From the table, we see that *GruMon* detected about 80–90 couples, about 10 groups of three individuals, and hardly any groups of size four or more. Given these were passengers on morning flights in the first three working days of the week, these numbers are possibly reasonable (as large families or groups of friends usually leave for vacations on weekends).

Effect of adding false positive data: Next, we examined the effect of funneling (i.e., large crowds following the same path) caused by high people densities. To do this, we used the set of Mar 3 traces used in Table 8 and assumed that the groups detected within the traces (as depicted in Table 8) represented ground truth. We then mixed other traces with this original set, re-ran *GruMon*, and observed the effect of this mixing on the recall and precision of detected groups using the assumed ground truth as a baseline.

We mixed three different additional traces with the original data: a) traces which ended at gates Z across all possible times of the day (likely to be departing passengers taking flights from the same gate at different times of the day), b) traces which end at gates Y between 1130 to 1330 hours (likely departing passengers leaving at the same time but

from a different gate on the same side of the airport as gates Z), and c) all traces which have some overlap in time with 1130 to 1330 hours (these are all devices in the airport terminal during that time period). We call the original set as *base*, and the sets with different additional traces as $mixed_{time\_gate}$, $mixed_{gate}$, and $mixed_{time\_all}$ respectively.

Here, recall is defined in the same way as for the edge clustering recall in Section 6, i.e., for each group in the *base*, the recall is computed as the fraction of members detected together in the mixed sets, and these values are then averaged over all *base* groups. Precision, or the fraction of correct groups among detected groups in the mixed sets, is computed slightly differently as we changed the denominator to the detected groups with at least one overlapping member with any *base* group. This was to avoid incorrect results by including the many possible groups detected within the background traces themselves.

Table 9 shows the values of the performance metrics, together with the number of traces added in each mixed set. The *base* contains 679 traces. From the table, time separation ($mixed_{time\_gate}$) helps to retain the original groups, giving both high recall and precision. However, $mixed_{gate}$, inspite of adding the least number of additional traces, sees a significant drop in precision. This might be reasonable, as all these traces belong to passengers departing in the same time window. So some traces might have had similar spatio-temporal characteristics at immigration and retail areas, with only the very end of the traces becoming divergent near the separate departure gates. This increases false positives and reduces precision. Adding all traces overlapping with the time window ($mixed_{time\_all}$), increases the number of traces, but does not degrade the metrics significantly further — possibly due to different movement characteristics for arriving passengers, passengers departing at other gates, or staff.

Parameter/ Metric	$mixed_{time\_gate}$	$mixed_{gate}$	$mixed_{time\_all}$
Traces added	6014	774	7749
Recall	98.49	93.31	90.29
Precision	96.82	84.35	82.41

Table 9. Effect of people density

Some other observations: *GruMon's* output will also be affected by the fraction of devices that communicate with the *GruMon* system. To examine this aspect, we again segregate the passengers departing from gates Z. From the online available flight schedules, we divided the day into five time slots, separated by times during which no flights departed from this gate. Within each time slot, multiple flights departed, which would have resulted in arriving and boarding passengers for different flights intermingling at the gate. Using available online information about each aircraft and seating capacities [28], we estimated the maximum number of people who might depart from gates Z during each time slot.

Table 10 shows the number of departing passengers as functions of the maximum capacity, and the corresponding number of detected devices each day, for the five time slots. We observed that the number of detected devices varies between days – either because the flights were not very full, or because less people connected to Wi-Fi at these gates. Overall, the *Airport* dataset exhibits strong temporal data sparsity

Time of day	Maximum capacity	Mar 3 detections	Mar 4 detections	Mar 5 detections
00:00-03:00	819	136	98	65
05:30-10:00	2291	1938	1949	1569
11:30-13:30	731	679	584	498
14:30-19:30	2029	1912	1702	1369
19:30-24:00	1718	820	520	892

Table 10. Sensing coverage

patterns. In addition, it is also possible for the sensor readings from client applications to exhibit sparseness issues due to power management, network connectivity, and other reasons. We plan to investigate and mitigate the impact of these types of data sparseness issues on *GruMon* in future work.

8 GruMon Live Deployment

We have been rolling out the live deployment of *GruMon* across all schools at the Singapore Management University, in two phases. We have completed the first phase, where GruMon solely uses the location-based features including *coordinated transitions* on continuously streamed location feeds. Location data are updated for all devices connected to the campus WiFi, at the server-side, without the involvement of the client devices. Currently, we are working on the second phase to integrate various sensor-based features. We plan to deploy this version to the participants of LiveLabs mobile testbed [29] that includes 2K+ signed participants who can provide sensory data in real time.

Figure 18 shows screenshot of based group-based analytics from the live dashboard on a particular day during term break which plots the number of groups detected by GruMon aggregated at 15 minute intervals. We see dynamic groups starting to form from around 7.30 AM, and peaking during midday through early evening (e.g. 11 AM to 6 PM) after which we see the number of groups falling as students start leaving the campus. Although the student population on-campus during the break is much lower than the term time, we see consistency in the number of groups detected, and the times over which they are detected. We also observe notable differences between the student visit population between weekdays vs. weekends. The number of groups detected before 7.30 AM are in fact devices that are co-located (such as phones, laptops, etc. lying around in the labs and offices), and are not actual students, which serves as our base case (about 50 groups). We also notice abrupt drops in the number of groups detected throughout the day – we're investigating this further as the reason behind this is not fully clear to us currently.

9 Conclusion and Future Work

In this paper, we presented *GruMon*, a group detection system for dense, urban spaces. *GruMon* can achieve good results even when location information is unavailable, using novel sensor-derived micro-activity features. Also, when location information is present, it further improves detection precision and latency by using semantic transitions. We conducted extensive evaluations using real datasets from two shopping malls and an airport, and showed that *GruMon* can detect most of the groups ($> 80\%$) with high precision (97%) even with a) many groups (e.g., 54% of the groups for $Mall_2$) separating at some period during the data collection period, and b) varying levels of location accuracy.

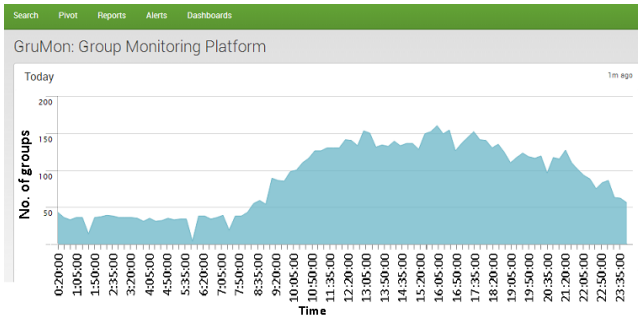

Figure 18. Screenshot from the GruMon live deployment at the SMU campus for a particular day.

In the future, we plan to explore several interesting directions to extend the current work. First, we plan to enhance *GruMon* with the ability to detect the relationships between members of detected groups. This will be a key enabler for useful group-aware promotion together with *GruMon*'s group detection capability. Second, it will be interesting to enhance the system with upcoming wearable devices like smart glasses and watches. These could help to visually capture groups with members who do not carry smartphones such as children and elderlies.

In addition, in the shorter term, we plan to deploy and evaluate GruMon in more diverse urban venues. Especially, we are considering challenging venues like fairs, exhibitions and museums, where lots of visitors follow a pre-designed walking flow and exhibit similar movement patterns even though they are not in groups. Lastly, we also plan to expand the coverage of *GruMon* beyond Android users. In the current design, the users of closed platforms like Windows phones and iOS, can be only traced using server-side localization, as they do not allow applications to access raw Wi-Fi scan values. To support those users, native implementations at the OS level, would be necessary.

10 Acknowledgments

We would like to thank the anonymous reviewers and the shepherd for providing constructive feedback to improve the paper. We also thank all the participants for making the design and empirical evaluation of our system feasible.

This work is supported by the National Research Foundation, Prime Minister's Office, Singapore, under IDM Futures Funding Initiative and International Research Centre @ Singapore Funding Initiative, and administered by the Interactive & Digital Media Program Office, Media Development Authority, and by the Singapore Ministry of Education Academic Research Fund Tier 2 under research grant MOE2011-T2-1001. All findings and recommendations are those of the authors and do not necessarily reflect the views of the granting agency, or Singapore Management University.

11 References

[1] News. `http://www.gartner.com/newsroom/id/1827614`.

[2] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: A partition-and-group framework. In *SIGMOD*, Beijing, 2007.

[3] Lu-An Tang, Yu Zheng, Jing Yuan, Jiawei Han, Alice Leung, Wen-Chih Peng, and Thomas La Porta. A framework of traveling companion discovery on trajectory data streams. In *ACM Transactions on Intelligent Systems and Technology*, 2013.

[4] Hoyoung Jeung, Heng Tao Shen, and Xiaofang Zhou. Convoy queries in spatio-temporal databases. In *ICDE*, Cancun, Mexico, 2008.

[5] Zhenhui Li, Bolin Ding, Jiawei Han, and Roland Kays. Swarm: Mining relaxed temporal moving object clusters. *PVLDB*, 3(1), 2010.

[6] Siyuan Liu, Shuhui Wang, Kasthuri Jayarajah, Archan Misra, and Ramayya Krishnan. Todmis: Mining communities from trajectories. In *CIKM*, San Francisco, USA, 2013.

[7] Mikkel Baun Kjargaard, Henrik Blunck, Markus Wustenberg, Kaj Gronbask, Martin Wirz, Daniel Roggen, and Gerhard Troster. Time-lag method for detecting following and leadership behavior of pedestrians from mobile sensing data. In *Percom*, San Diego, USA, 2013.

[8] Cisco lbs design guide. `http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/WiFiLBS-DG.pdf`.

[9] Ruckus indoor location. `http://www.digitalairwireless.com/wireless-blog/recent/ruckus-wireless-spot-location-based-best-practices-part-two.html`.

[10] U-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm. Accuracy characterization for metropolitan-scale wi-fi localization. In *MobiSys*, Seattle, USA, 2005.

[11] Empirical Evaluation of the Limits on Localization Using Signal Strength. Gayathri chandrasekaran and mesut ali ergin and jie yang and song liu and yingying chen and marco gruteser and richard p. martin. In *SECON*, New Orleans, USA, 2009.

[12] P. Bahl and V.N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM*, Tel Aviv, Israel, 2000.

[13] Moustafa Youssef and Ashok Agrawala. The horus wlan location determination system. In *MobiSys*, Seattle, USA, 2005.

[14] Nathan Eagle and Alex (Sandy) Pentland. Reality mining: Sensing complex social systems. *Springer Personal Ubiquitous Computing Journal*, 10(4), 2006.

[15] Kiran K. Rachuri, Cecilia Mascolo, Mirco Musolesi, and Peter J. Rentfrow. Sociablesense: Exploring the trade-offs of adaptive sampling and computation offloading for social sensing. In *MobiCom*, Las Vegas, USA, 2011.

[16] N. Banerjee, S. Agarwal, V. Bahl, R. Chandra, A. Wolman, and M. D. Corner. Virtual compass: relative positioning to sense mobile social interactions. In *Pervasive*, Helsinki, Finland, 2010.

[17] Android developer reference - bluetooth adapter. `http://developer.android.com/reference/android/bluetooth/`.

[18] P. Aditya, V. Erdélyi, M. Lentz, E. Shi, B. Bhattacharjee, and P. Druschel. Encore: Private, context-based communication for mobile social apps. In *MobiSys*, Bretton Woods, USA, 2014.

[19] Y. Lee, C. Min, C. Hwang, J. Lee, I. Hwang, Y. Ju, C. Yoo, M. Moon, U. Lee, and J. Song. Sociophone: Everyday face-to-face interaction monitoring platform using multi-phone sensor fusion. In *MobiSys*, Taipei, Taiwan, 2013.

[20] Chengwen Luo and Mun Choon Chan. Socialweaver: Collaborative inference of human conversation networks using smartphones. In *SenSys*, Rome, Italy, 2013.

[21] J. J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, Lake Tahoe, Nevada, USA, 2012.

[22] Yuchen Zhao, Guan Wang, Philip S. Yu, Shaobo Liu, and Simon Zhang. Inferring social roles and statuses in social networks. In *SIGKDD*, Chicago, USA, 2013.

[23] Kartik Muralidharan, Azeem Javed Khan, Archan Misra, Rajesh Krishna Balan, and Sharad Agarwal. Barometric phone sensors – more hype than hope! In *HotMobile*, Santa Barbara, USA, 2014.

[24] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer.

[25] Markov clustering algorithm. `http://micans.org/mcl/`.

[26] C. Lim, S. Bohacek, J. Hespanha, and K. Obraczka. Hierarchical max-flow routing. In *GLOBECOM*, St. Louis, USA, 2005.

[27] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hong Cheng. Mining discriminative patterns for classifying trajectories on road networks. *IEEE Transactions on Knowledge and Data Engineering*, 2011.

[28] Seatguru: Airline seat maps. `http://www.seatguru.com/`.

[29] Rajesh Krishna Balan, Archan Misra, and Youngki Lee. LiveLabs: Building an in-situ real-time mobile experimentation testbed. In *HotMobile*, Santa Barbara, USA, 2014.