

SenseMe: A System for Continuous, On-Device, and Multi-dimensional Context and Activity Recognition

Preeti Bhargava
Department of Computer
Science
University of Maryland,
College Park
prbharga@cs.umd.edu

Nick Gramsky
Department of Computer
Science
University of Maryland,
College Park
ngramsky@cs.umd.edu

Ashok Agrawala
Department of Computer
Science
University of Maryland,
College Park
agrawala@cs.umd.edu

ABSTRACT

In order to make context-aware systems more effective and provide timely, personalized and relevant information to a user, the context or situation of the user must be clearly defined along several dimensions. To this end, the system needs to simultaneously recognize multiple dimensions of the user's situation such as location, physical activity etc. in an automated and unobtrusive manner. In this paper, we present SenseMe - a system that leverages a user's smartphone and its multiple sensors in order to perform continuous, on-device, and multi-dimensional context and activity recognition. It recognizes five dimensions of a user's situation in a robust, automated, scalable, power efficient and non-invasive manner to paint a context-rich picture of the user. We evaluate SenseMe against several metrics with the aid of 2 two-week long live deployments involving 15 participants. We demonstrate improved or comparable accuracy with respect to existing systems without requiring any user calibration or input.

Keywords

Context-aware computing, Context and Activity Recognition, Mobile Phone Sensing, Mobile Systems and Applications

Categories and Subject Descriptors

C.3.2 [[Special-Purpose and Application-based Systems]]: Real-time and embedded systems, Ubiquitous computing; H.5.2 [Information interfaces and presentation (e.g., HCI)]

General Terms

Design, Experimentation, Human Factors, Performance

1. INTRODUCTION

In order to make context-aware systems more effective and provide timely, personalized and relevant information to the user, the context or situation of the user must be clearly defined along several dimensions. Though a user's situation can have multiple dimensions or aspects at any given instant of time, some of the most

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MOBIQUITOUS 2014, December 02-05, London, Great Britain
Copyright © 2014 ICST 978-1-63190-039-6
DOI 10.4108/icst.mobiqutous.2014.257654

important ones are: where he is (location), what he is doing (activity), and when (time) i.e. "Who, What, Where and When?" [6]. These aspects form the W4 model proposed by Castelli et al. [13] to represent contextual information about physical world objects, which can be employed by both users and context-aware systems. An additional aspect - whom the user is with, proposed by Schilit et al. [24], can be added to augment this model. Taken together, these five dimensions paint a context-rich picture of the user.

To determine these aspects of a user's situation, the context-aware system needs to simultaneously recognize multi-dimensional contextual information of a user at a given instant of time. Moreover, for large-scale adoption by users, this information should be acquired in a non-invasive manner, without placing undue burden on them. As a result, automated sensing ability is highly desirable for any information system powered by context-aware computing. In addition, this ability to sense users should be embedded in devices that they can carry around without effort and the user interaction should be minimal so that it is unobtrusive.

The ubiquitous smartphone, with its multitude of sensors and capabilities, has become the best choice for this purpose. Today's smart phones come equipped with an increasing range of sensing, computational, storage and communication capabilities. This has enabled sensing and tracking applications to emerge across a wide variety of applications areas such as location based services, personal healthcare, and social networking etc. A key challenge of mobile phone sensing is to process raw data from multiple sensors in order to infer higher level activities and context in real-time and in a robust, generic and energy efficient manner.

In this paper, we present SenseMe - a system that leverages the smartphone and its multiple sensors in order to perform **continuous, on-device, and multi-dimensional context and activity recognition** for a user. It achieves this in a robust, automated, scalable, power efficient and non-invasive manner. Table 1 summarizes the set of five dimensions of a user's situation (and their possible discrete values) that SenseMe recognizes, with a temporal resolution of a minute¹, to determine the aspects "Who, What, Where, When, and Who you are with?". The dimensions are:

1. *Environmental context* - This dimension represents whether the user is Outdoors (outside a building), Indoors (inside a building), or Indoor-Outdoors (close to or in a semi-open building or inside a building but near a door or large window). It is significant as it enables context-aware localization(explained next).
2. *Context-aware Location* - This dimension represents location of the user in both indoor and outdoor environments. If he is 'outdoors', SenseMe uses GPS for localization and reverse geocoding in order to resolve the logical address of the user's location. If he is 'indoors' or in the 'indoor-outdoor'

¹This time slice duration is long enough to be discriminative and short enough to provide high accuracy labeling results.

Situation Dimension	Possible Values
Environmental Context	{Indoor, Outdoor, Indoor-Outdoor}
Physical Activity	{Stationary, Walking, Running, In-vehicle}
Context-aware Location	Set of locations determined by Wi-Fi (indoors) or reverse geocoding(outdoors)
Device Activity	Task the user is engaged in on the device such as phone call or messaging
Social Context	Number of people around the user at any given instant of time

Table 1: Situation dimensions being captured by SenseMe at any instant of time

Device	Range		Resolution	
	Magnetic Field (μ T)	Light (lux)	Magnetic Field(μ T)	Light (lux)
Google Nexus	9830.0	65528.0	0.15	0.2
Motorola Xoom	2000.0	208076.8	0.0625	0.05

Table 2: Comparison of Range and Resolution for Magnetic Field and Light sensors on two different devices

area, SenseMe performs indoor localization using Locus [12] which is a Wi-Fi based indoor localization system for multi-story buildings.

Since the location should be in a human understandable format, SenseMe determines a high-level logical location in addition to a low-level location in a raw format. It further identifies a location type (restaurant, academic building, student center etc.) for each resolved location. This enables tracking of a user’s location history at fine and coarse grained levels throughout the day and also helps in identifying the places where they spend a significant amount of their time.

3. *Physical Activity* - This dimension represents the physical activity of the user such as walking, running, stationary or in a vehicle (car, bus, bike etc.).
4. *Device Activity* - This dimension represents the task the user is currently engaged in on his/her smart device (checking mail or phone call). It is equally important as the physical activity due to the growing proliferation of mobile devices and their increasing usage as opposed to desktop and laptop computers. Most people carry their smart devices everywhere and perform a substantial set of their everyday activities such as web browsing on it. As a result, we believe that it forms a significant dimension of the user’s situation.
5. *Social context* - This dimension represents the social activity of users i.e. how much time they spent interacting or being around people.

Thus, at any instant of time t , SenseMe represents the user’s situation S as a feature vector in a multi-dimensional *Situation Space*. For example, $S(t)$ for a user is: <Indoor; Stationary; Phone Call; A.V. Williams Building - College Academic Building; With 4 people>.

SenseMe has been completely implemented on the Android platform and runs on off-the-shelf Android smartphones and tablets. We evaluate SenseMe extensively against several qualitative and quantitative metrics, with the aid of 2 two-week long live deployments involving 15 participants. We demonstrate improved or comparable accuracy with respect to existing systems without requiring any user calibration or input.

The rest of the paper is organized as follows: In Section 2, we

discuss existing related work in the field of Context and Activity Recognition and highlight their shortcomings and differences with our approach. Following that, we explain the key contributions of our work in Section 3 and training data collection in Section 4. Section 5 describes the SenseMe system. We present the evaluation of SenseMe in Section 6 and briefly describe the SenseMe API in Section 7. Finally, we conclude by summarizing our contributions.

2. RELATED WORK

There have been several recent efforts in the field of context and activity recognition. We discuss some notable and relevant examples here including those whose goals are similar to ours i.e. they use off-the-shelf devices such as mobile phones rather than proprietary hardware or sensors. We also highlight their limitations and differences with our approach. Moreover, most of these efforts have been isolated and capture a single dimension of context or activity as opposed to multi-dimensional context and activity recognition.

2.1 Environmental Context Recognition

IODetector[27] is a sensing service that runs on the mobile phone and uses light and magnetic field sensors, and cell tower signals in order to detect whether the device is outdoors, indoors, or semi-outdoors². However, sensors such as the magnetic field and light sensors often depend on device manufacturer. As shown in Table 2, the range and resolution of these sensors vary with each device. The output of these sensors also varies with time of the day and weather. Hence, extensive calibration and hand tuning as done in IODetector[27] is not a robust and accurate method. In addition, they use the cell tower signal strength but many smart devices such as tablets do not come equipped with the cellular radio. Overall, it has an average accuracy of 88%.

The unavailability of a GPS fix has been used by Ravindranath et al. [23] to infer that the user is in an indoor environment. However, just the availability or unavailability of the GPS fix is not a robust parameter and can lead to many false positives. It is possible to have a GPS fix indoors even if its weak.

TempIO [19] determines environmental context by comparing the environment temperature, measured using proprietary hardware, with the current outdoor temperature obtained from a web service or external thermometer. A major limitation of this work is that web services usually provide temperatures at a coarse granularity of a city or locality rather than an exact fine grained location. Moreover, users have to carry the external hardware along with them as temperature sensors may not be available on all devices.

On the other hand, SenseMe performs environmental context recognition using NMEA 0183 data[5], obtained from GPS, which is a standard data specification used for communication between electronic devices such as GPS receivers and other types of instruments. This makes it robust and independent of time, weather, and device manufacturer.

2.2 Physical Activity Recognition

CenceMe [21] is mobile phone system which uses accelerometer, GPS, audio and bluetooth to infer human activities such as ‘Walking’, ‘Standing’, ‘Running’, ‘Sitting’ and ‘Vehicle’. It runs on the Nokia N95 with components written in both JME and Symbian C++. To preserve phone resources, certain computations are split between the phone and a back end desktop server. The accuracy of the classifier varies with the activity being classified - high (94%) for ‘Walking’ but low(<80%) for the others. CenceMe uses kMeans clustering to identify significant locations inhabited by users. It injects the users’ presence and current activity on a social network which can be privacy invasive. Moreover, there are

²We use the same definitions in SenseMe for Indoor, Outdoor and Indoor-Outdoor respectively.

latency challenges in splitting the computation between the phone and a backend server as this slows down the response time of the system and requires network connectivity at all times. In addition, there are privacy concerns and other costs associated with uploading personal data of users to a server or cloud.

Jigsaw [20] is an application with a similar premise but different implementation. It uses three different pipelines for accelerometer, GPS and microphone and runs entirely on the phone. The classifier accuracies for the same set of activities as CenceMe is about 94%. The microphone pipeline detects activities such as brushing, showering, typing, vacuuming etc. and its accuracy ranges from 84% to 88%. While the ability to detect higher number of activities is definitely an advantage, the tradeoff between accuracy, utility and energy consumption must be maintained. The microphone is a power hungry sensor and hence, using it for activity recognition poses significant challenges. Additionally, this raises privacy concerns [18] and can have legal implications as recording audio in any form may require users' permission.

Using accelerometer for activity recognition suffers from several limitations which make it a non-robust method:

- High likelihood of false positives - if a user shakes his phone, it is often labeled as a physical activity.
- Dependency on the gait of a user,
- Dependency on placement of the phone - whether it is placed on the body or if its in a bag or a purse.
- User calibration is required in order to make it independent of orientation and body position[20].

To address these limitations, SenseMe recognizes a user's physical activity based on the speed of the device obtained from the GPS, which makes it agnostic to gait, body position and orientation.

2.3 Localization

There exists a wide spectrum of research in indoor and outdoor localization using off the shelf devices as well as instrumented setups. All of them focus mainly on either indoor localization (Wi-Fi based systems such as RADAR[9], Horus[26], Active Campus[14] and Locus[12]) or outdoor localization (GPS based systems such as EnTracked[17] and StarTrack[8]). However, our aim in SenseMe is to enable context-aware localization i.e. localization in both indoor and outdoor environments through technologies that are readily available through the smartphone - GPS and Wi-Fi. This aids in capturing all of the user's locations in an unobtrusive manner using a ubiquitous device.

2.4 Social Context Recognition

CenceMe [21] determines social context by scanning a user's environment for recognized bluetooth devices and displaying the number of 'CenceMe buddies' (other CenceMe users) that are around. However, this requires location sharing which can be privacy-invasive. Hence, in SenseMe, we address the general problem of determining how many people are around the user, irrespective of whether they use SenseMe or not. Moreover, many users may not consent to sharing their data especially location with other users and as a result, we do not support that in the current system. Instead, we use bluetooth to recognize a user's social context.

2.5 Device Activity Recognition

There are several commercially available smart phone applications that track the application usage of a user and organize the applications based on the Most Frequently/Recently used application. On the other hand, in SenseMe, we add a temporal aspect to the application usage and observe it in tandem with other recognized dimensions.

2.6 Logging raw sensory information

In the FunF project [7], Aharon et al. log a variety of sensory

Classifier	Environmental Context(%)	Physical Activity(%)	Training time (s)
C4.5	96.87	93.18	0.1
kNN	92.74	83.77	0.2
Random Forest	92.65	94.79	34

Table 3: Comparison of classifier accuracies(%) and average training time(s) for the training datasets

information from the devices of 55 users with a maximum temporal resolution of 6 minutes. Wagner et al. [25] undertake the challenge of large scale smartphone usage data collection from Android devices of 21,350 users over a period of 2 years. The data collected includes accelerometer readings, call logs, cell tower scans etc. However, these approaches log low level data without inferring any high-level context or activities from it.

3. KEY CONTRIBUTIONS

Our key contributions in this paper are:

1. We present a robust, generic and scalable technique for performing environmental context recognition which is independent of time, weather, and device manufacturer.
2. We present a robust, generic and scalable technique for performing activity recognition, for select physical activities, which is independent of gait, body position and orientation.
3. We utilize the user's environmental context and physical activity to perform opportunistic context-aware localization using existing technologies that are generic and easily scalable.
4. We capture the user's device activity as well as social context to augment his multi-dimensional situation.
5. We implement the aforementioned techniques as part of a generic system, SenseMe that runs entirely on the smart device and is completely non-invasive.
6. We demonstrate improved or comparable accuracy with respect to other existing systems without requiring any user calibration or input.

Since SenseMe uses GPS and several other sensors, managing power consumption is crucial. To this end, we have implemented a resource efficient duty cycle that employs power conservation techniques to control GPS usage without sacrificing accuracy.

4. TRAINING DATA COLLECTION

To implement the SenseMe system, we first collected NMEA 0183 and speed data over a period of one month in different environments (such as university, office buildings, high rises, apartment complexes) and in different road conditions (such as highways, downtown, city and local roads).

The NMEA training data samples include specific NMEA sentences received in a time span of a minute. NMEA sentences consist of several words separated by a ',' and the first word, also called the data type, defines the interpretation of the rest of the sentence. The two sentences that interest us the most are the GGA and GSA sentences. These sentences contain meta-data about the GPS fix such as number of visible satellites and the Dilution of Precision (DOP) [4]. The DOP is the relative accuracy of horizontal (HDOP) or vertical (VDOP) position as the case may be. It is a number where a smaller value means a higher level of accuracy. Once an NMEA listener is enabled on the device, it starts receiving these sentences every second irrespective of a GPS fix being achieved and independent of the GPS sampling rate. The speed training data samples consist of raw speed values of the device received in a time span of a minute sampled at a minimum interval of 10 seconds.

The data was collected by 4 members of our lab (including the authors) at several times in a day as well as in different weather

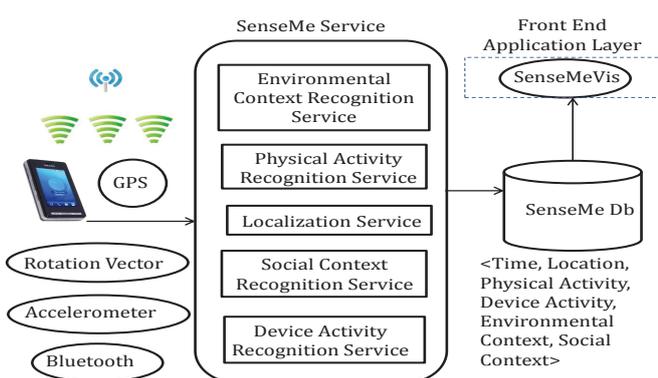


Figure 1: Architecture of the SenseMe system

conditions, to study their effects (if any) on the data. All the participants who collected the data also annotated it carefully to provide ground truth values for each dimension being captured. We computed Pearson’s linear correlation coefficient on the collected NMEA data and determined that the weather specifically outlook had a strong correlation with it. Hence, we removed it from our set of features. The total data collected was approximately equivalent to a continuous run of 168 hours.

We experimented with 3 classifiers on both the NMEA and speed training datasets: C4.5, kNN (k=3) and Random Forest (10 trees). The Weka[15] implementation of each classifier was run on these sets with 10 fold cross validation. Table 3 shows a comparison of the classifiers’ accuracies as well as average training time for environmental context and physical activity recognition. C4.5 proved to be faster, more accurate and efficient than both kNN and Random Forest for environmental context recognition. For physical activity recognition, it proved to be more accurate than kNN and slightly less accurate than Random Forest but faster than both.

Since smart devices have memory, CPU and power constraints, it is most effective and efficient to use a fast, accurate and light-weight classifier. The C4.5 decision tree is a light-weight classifier as opposed to the kNN classifier (which is an instance based method for classification and hence requires in-memory storage of training instances) and Random Forest (which is an ensemble classifier). As a result, we selected the C4.5 decision tree on the basis of its performance and also because it is fast and not computationally intensive. We implemented it in SenseMe for environmental context and physical activity recognition. There are more advanced techniques like Support Vector Machines which generate a confidence measure with each classification label. This can then be used as an input to a Hidden Markov Model for smoothing. However, in SenseMe we attempt to balance resource usage efficiency with accuracy on a resource constrained platform and the C4.5 classifier achieves it superbly.

5. THE SENSEME SYSTEM

We now describe the SenseMe system in detail.

5.1 System Design and Architecture

Figure 1 shows the architecture of the SenseMe system. It has been completely implemented on the Android platform. It consists of a background Android service called SenseMeService, which consists of 5 individual services (one for each dimension), a SQLite database called SenseMeDb and a foreground proof of concept visualization called SenseMeVis. The temporal context and activity information recognized by each service is stored in SenseMeDb. SenseMe can run on the user’s phone as an application which can

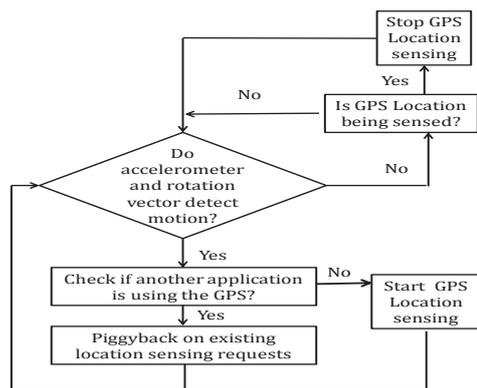


Figure 2: GPS Duty Cycle

be easily pushed to the background where it continuously functions, collects and processes data. Whenever the application is brought to the foreground, SenseMeVis retrieves the information from SenseMeDb in order to render it on the device display.

5.2 GPS Duty Cycle

Zhuang et al. [28] propose several techniques for preserving energy consumption of location based applications specifically on the Android platform. These include: (i) Substitution (replacing a more accurate but energy intensive provider such as GPS with a less accurate but efficient provider such as Network), (ii) Suppression (using low power sensors to suppress the usage of GPS), (iii) Piggybacking (synchronizing the location sensing requests with existing requests) and (iv) Adaptation (adapting the system-wide sensing parameters such as time and distance, when battery level is low). Some of these techniques such as Substitution and Adaptation preserve the battery at the cost of location accuracy.

In SenseMe, we utilize the GPS in an energy and resource efficient manner without sacrificing accuracy. Figure 2 shows the workflow of the GPS Duty Cycle in SenseMe. It employs the following techniques to conserve power usage:

5.2.1 Suppression

Several systems such as SenseLess [10] and Nericell [22] have been proposed to utilize the accelerometer as a control for the GPS. On a similar line, SenseMe uses the linear accelerometer as well as the rotation vector sensor as a means to control or suppress the GPS usage. Since they are light-weight sensors and consume very little power, they can be used as an effective trigger for turning the GPS on when motion is detected and off otherwise.

We calculate the ℓ_2 norm root of the linear acceleration and rotation vectors to get the overall linear acceleration and rotation of the device. Some of the samples can be noisy and hence we do not rely on raw values. Instead, we calculate the average and variance of the acceleration and rotation values obtained from 150 samples to obtain four parameters, termed as μ_{acc} , σ_{acc} , μ_{rot} and σ_{rot} respectively. These samples can be spread over 10 to 30 seconds according to the sampling rate, as specified in the Android API[2].

We empirically studied the linear acceleration and rotation vector values obtained from different Android devices when a user was stationary and when he/she was in motion. We determined suitable thresholds for all the parameters. If all the parameters are greater than their respective threshold values, it implies that the device is in motion. In all other cases, it implies that the device is not moving.

5.2.2 Piggybacking

When the linear acceleration and rotation vector sensors detect

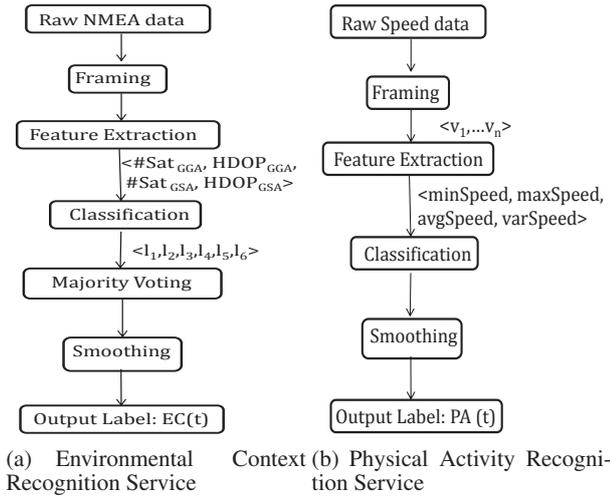


Figure 3: Environmental Context and Physical Activity Recognition Services

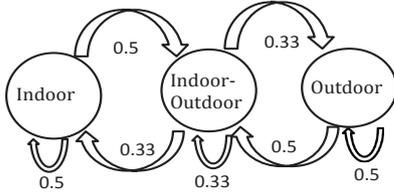


Figure 4: Indoor-Outdoor transition Markov Chain

motion, the system checks if another application is already using the GPS. If yes, it piggybacks location sensing on the existing requests. If not, it explicitly starts the GPS for location sensing.

5.2.3 Sensing Adaptation

We attempt to balance the trade-off between accuracy and power consumption, by employing an optimum sensing interval (in terms of time and distance) for location updates[1]. This is based on the intensity of the physical activity (walking as opposed to driving) of the user and further aids in preserving battery life and consumption.

5.3 The SenseMe Service

The SenseMe service consists of the following individual services:

5.3.1 The Environmental Context Recognition Service

This service uses NMEA 0183 data from the GPS to recognize the environmental context for a user. Figure 3(a) shows the pipeline for it. It has the following stages:

- Framing - Since NMEA sentences are received every second, we operate on frames of these sentences where each frame consists of $\tau=10$ seconds.
- Feature Extraction - For each frame, we average the number of satellites and the HDOP for both the GSA and GGA sentences to create a feature vector of the form $\langle \#Sat_{GGA}, HDOP_{GGA}, \#Sat_{GSA}, HDOP_{GSA} \rangle$.
- Classification - This feature vector is then used as input to a C4.5 classifier, to generate one of the following labels - 'Indoor', 'Outdoor', 'Indoor-Outdoor'. Thus, we have a vec-

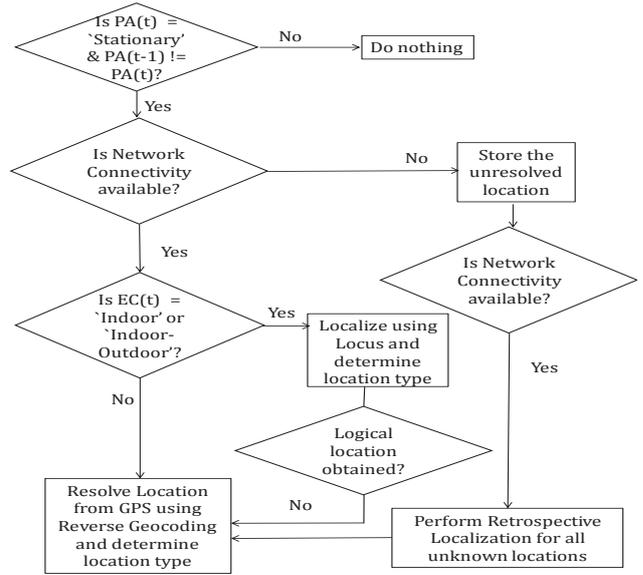


Figure 5: Opportunistic context-aware Localization performed by the Localization service

tor of six labels $\langle l_1, l_2, l_3, l_4, l_5, l_6 \rangle$ for every minute.

- Majority Voting - We then perform majority voting on this vector to generate an environmental context value, $EC(t)$, for each minute.
- Temporal Smoothing - For smoothing outliers, we have implemented this service as a stateful service and modeled the environmental context as a 1st order Markov Chain, where the current state i.e. environmental context at time t , $EC(t)$, is dependent only on the previous state at time $t-1$ i.e. $EC(t-1)$. Figure 4 shows the transition probabilities for the Markov Chain and is derived from the fact that any person when moving from 'indoor' to 'outdoor' state or vice versa will always go via the 'indoor-outdoor' state.

5.3.2 The Physical Activity Recognition Service

This service uses raw speed data from the GPS to recognize the physical activity of a user. Figure 3(b) shows the pipeline for it. It has the following stages:

- Framing - According to the duty cycle, the service receives speed data at a minimum interval of 10 seconds and a maximum interval of 60 seconds depending on the motion of the device. Each frame consists of $\tau = 60$ seconds and for each frame, we get a raw speed vector $S = \langle v_1, \dots, v_n \rangle$, where $n \in [1,6]$.
- Feature Extraction - We extract statistical features from this raw speed vector to generate a feature vector of the form $\langle \minSpeed, \maxSpeed, \text{avgSpeed}, \text{varSpeed} \rangle$ where \minSpeed is the minimum, \maxSpeed is the maximum, avgSpeed is the average and varSpeed is the variance, of all the speed values in S .
- Classification - This feature vector is used as input to a C4.5 classifier, to generate one of the following labels - 'Stationary', 'Walking', 'Running' and 'In-vehicle'. Thus, we get a Physical Activity value, $PA(t)$, for the user for every minute.
- Temporal Smoothing - This service is a stateful service and we store the values $PA(t-1)$ and $PA(t-2)$ at any given instant of time. We use a sliding window smoother of size 3 to smoothen any outliers out.

5.3.3 The Localization Service

Although SenseMe senses the raw location of the user in accordance with the GPS duty cycle, we resolve the location to a logical address only when the user is ‘Stationary’. This serves three purposes: (i) It reduces irrelevant contextual information in the form of multiple logical locations that the user could be passing through. For instance, if the user is driving on a highway, he/she will pass through multiple locations within a few seconds. (ii) It prevents unnecessary network bandwidth usage. (iii) It can help determine the user’s mobility patterns such as daily commute routes and travel paths. It also capture the places, where they spend a significant amount of their time (such as home, work, restaurants and coffee shops), with fine-grained accuracy.

Figure 5 shows the workflow of the Localization service. As shown, the Localization Service performs context-aware localization on the basis of the user’s current environmental context i.e. EC(t) and the current physical activity i.e. PA(t). Thus, if PA(t) is determined to be ‘Stationary’, after a transition from another physical activity, SenseMe localizes the user to a logical indoor or outdoor address. If EC(t) is ‘outdoor’, it resolves the location obtained from the GPS (which is in latitude and longitude format) to provide a logical address using the Android Reverse Geocoding API. If EC(t) is ‘indoor’ or ‘indoor-outdoor’, it employs the Locus [12] system to determine the Room #, Floor #, and Building the user is in. The main benefits of Locus are that it is a calibration-free, readily deployable, scalable and robust system for floor as well as location determination in multi-story buildings. It relies on existing infrastructure and off-the-shelf mobile device capabilities, and requires no proprietary hardware to be installed.

If an indoor location could not be obtained using Locus (due Wi-Fi being off or disconnected), SenseMe resolves the last sensed outdoor location so that a coarse-grained location for the user can be obtained. This is only to ensure that the user is always localized whenever he/she is stationary and there are no unknown locations in the user’s location history. If a network or data connection is not available at any time, SenseMe performs what we term as *Retrospective Localization*. It stores every unresolved location where the user was stationary and localization could not be performed. As soon as a data or network connection is available, SenseMe performs opportunistic localization of all unknown locations.

Once localization has been done and a logical address is obtained for the location, the Localization Service also determines the location type if available. If the user is outdoors, the location type is a Foursquare category obtained via the FourSquare Venues API³ (such as “College Academic Building”) or a Google Place type via the Google Places API⁴. If the user is indoors, the location type has two fields: a coarse-grained “Building Type” which refers to the type of the Building or establishment the user is in and a fine-grained “Room Use” which specifies the category of the room the user is in, for example, “Research Laboratory”. This is useful meta information that can enable semantic place prediction.

5.3.4 The Device Activity Recognition Service

This service determines the task that the user is engaged in on his smart device, for instance, a phone call, web browsing, or using a navigation application such as Maps. It polls the device OS every minute to determine which application is running in the foreground while the screen is on and active. If a media file such as an audio or video file is being played, SenseMe also records its metadata (track, artist, album name etc.).

5.3.5 The Social Context Recognition Service

This service uses bluetooth to scan the user’s environment every 2 minutes to determine a *social meter* i.e how many people

³<https://developer.foursquare.com/overview/venues.htm>

⁴<https://developers.google.com/places/>

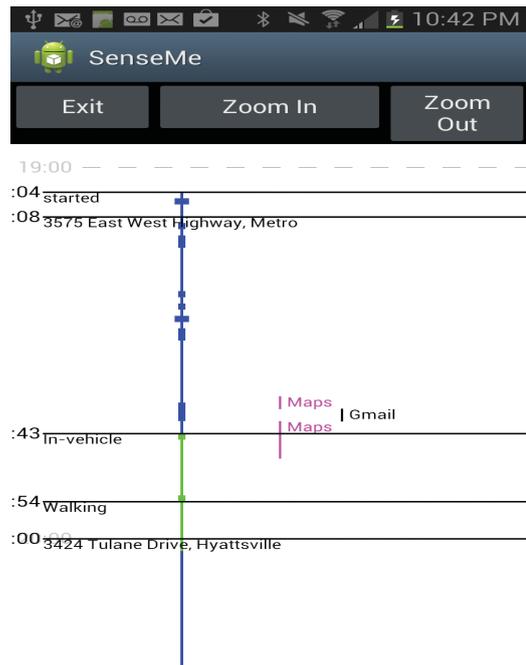


Figure 6: Screenshot of the SenseMeVis timeline (best viewed in color)

and/or their devices are around the user in any given time interval of 2 minutes. It obtains the device class[3] of all the devices it hears and filters them on the basis of their type. Thus, portable devices such as phones, laptops, hand held PDAs and wearable devices are counted, while other devices are not, as these devices are more likely to be carried around by people. Since bluetooth scans take about 60 seconds on an average to complete [16], we have set a threshold interval of 60 seconds between successive scans in order to balance power consumption with accuracy.

5.4 Proof of Concept Visualization - SenseMeVis

As mentioned earlier, SenseMe continuously operates as a background service. Whenever the application is resumed or brought to the foreground, the multi-dimensional context and activity information recognized by it is rendered on the device display via a foreground proof of concept visualization called SenseMeVis. The information is visualized for a maximum period of 24 hours in retrospect from the current time instant.

Figure 6 shows a screenshot of SenseMeVis visualizing the multi-dimensional contextual information recognized by SenseMe while it was running on a user’s device. It is a scrollable timeline that runs from top to bottom, showing the multiple dimensions of context and activities. Time units are fixed, thus, allowing the user to compare the information without needing to interpret how long something took place. The time and date at which SenseMe was started is at the top of the visualization. Markers on the left hand side of the visual along with dashed lines partition the hours to help temporally anchor the rendered information across the screen.

There are three types of visualizations that portray the 5 dimensions of context and activity that SenseMe recognizes:

1. The vertical central bar of varying thickness (known as the *Context bar*) represents environmental and social context. The environmental context is conveyed through the color of the bar - a blue segment indicates the user was indoors, green symbolizes outdoors and red is the indoor-outdoor state. The thickness of the bar conveys the social context. The wider

the bar, the more the number of people and portable devices the user is surrounded by. A very skinny bar indicates the user is not near anyone.

2. The horizontal solid black lines and subsequent labels partition the timeline based on location and physical activity. As mentioned earlier, the localization service performs context-aware localization only when the user is ‘Stationary’ after a transition from another activity. Similarly, for clarity and legibility of the visualization, we display either a location (if the physical activity is not ‘Stationary’) or a physical activity. Thus, if the user were stationary, SenseMe displays his location but if the user were in motion, SenseMe displays his physical activity. This simplification prevents the rendering of multiple locations in a short period of time. For example, if a user is in a car traveling at 60 mph, rather than displaying all the locations he might pass through, we summarize that timeframe with his physical activity - ‘In vehicle’.
3. The vertical lines of varying colors, on the right side of the context bar, represent the device activity of the user. Each activity is assigned a column and a color. This ensures that all entries for a given activity always fall within the same column and are colored the same to facilitate ease in understanding of the visualization. As there is no limit to the number of unique activities a user could be doing, activities may share a column but the color will be different allowing the user to distinguish between them easily.

As shown, the current run of SenseMe is started around 7:04 pm. It recognizes that the user is indoors and stationary so it localizes her and resolves a logical address for her location. She is indoors for about 35 minutes and is surrounded by varying number of people during this time period. She is also performing different activities on her device during this period such as checking her email and accessing a navigation application. She then goes outdoors and is driving alone for about 11 minutes. Finally, she parks her vehicle, starts waking and gets home around 8 pm.

6. EVALUATION

6.1 Evaluation Metrics

SenseMe is a versatile system for continuous, on-device, and multi-dimensional context and activity recognition. Hence, it needs to meet the following qualitative and quantitative requirements:

- High accuracy and scalability - As a generic system that can be used by multiple users on different devices, it should be highly accurate when recognizing context and activities. It should also be scalable to a large and varied set of users.
- Generality and Robustness - It should be general and robust enough to be used at any time in any environment.
- Universal Applicability - It should be applicable to any device and independent of device manufacturer.
- Minimum latency and Robustness to network failure - It should provide responses in real time even when there is sparse or no network connectivity.
- Non-invasive with minimum user calibration required - It should be non-invasive, capable of operating in the background and require minimum user input or calibration.
- Privacy preserving - Since a user’s context history has privacy implications, the data should be kept secure, confidential and shared only with the user.
- Energy efficiency - Being a system running on mobile devices with constrained energy budgets, it should use resource efficient methods and duty cycles.

We now evaluate SenseMe against each of these requirements.

6.2 Methodology

To evaluate SenseMe, we conducted two live deployments of 2

Ground Truth	SenseMe		
	indoor	outdoor	indoor-outdoor
indoor	0.984	0.016	0.0
outdoor	0.07	0.93	0.0
indoor-outdoor	0.11	0.07	0.82

Table 4: Confusion matrix for environmental context recognition for test data

Ground Truth	SenseMe			
	Stationary	Walking	Running	In-vehicle
Stationary	1.0	0.0	0.0	0.0
Walking	0.0	0.93	0.04	0.03
Running	0.0	0.0	0.95	0.05
In-vehicle	0.0	0.01	0.039	0.951

Table 5: Confusion matrix for physical activity recognition for test data

SenseMe service	Overall Accuracy (%)
Environmental Context Recognition	91.23
Physical Activity Recognition	95.75
Context-aware Localization	93.12
Device Activity Recognition	99.1
Social Context Recognition	87.5

Table 6: Accuracy of SenseMe Services (%)

Battery usage metric	Google Nexus	Samsung Galaxy	Motorola Xoom
Maximum consumption (%)	18	28	40
Average consumption (%)	16	27	37.4
Average runtime (hrs)	24	24	24

Table 7: Comparison of Battery consumption and runtime during a 24 hour continuous run of SenseMe

weeks each. In the first deployment, we recruited 8 subjects - 7 of whom were male Computer Science graduate students while one was a male post doctoral associate. We incorporated the feedback from them and fixed minor bugs in the system. We then conducted a second deployment, for which we recruited 7 subjects (5 male and 2 female) who were professionals working across USA. None of these test subjects had participated in training data collection. In both the deployments, we installed SenseMe on the subjects’ personal devices as giving temporary devices to them for a study may affect their interaction and usage of the device and bias results.

All the subjects were asked to run SenseMe on their devices, in the background, for a period of 2 weeks while going about their daily life. They were also asked to keep a journal of their physical activities, locations, environments and number of people around them throughout the day. As an incentive, the subjects were provided monetary compensation. At the end of the two-week deployment period, the subjects were asked to submit their journals as well as the SenseMeDb databases, which contained the context and activity recognition information recognized by SenseMe. This data was used as a test dataset to determine the accuracy of the different SenseMe services. The estimated value for each dimension, as determined by SenseMe, was compared with the Ground Truth values provided by the subjects as part of their journals.

We used this methodology as we did not want SenseMe to be obtrusive and disrupt their daily life. Moreover, this methodology allowed the context and activity information to be captured in a real life practical scenario, thus, making the evaluation more effective.

6.3 Accuracy Results

Since Environmental Context and Physical Activity Recognition services employ multi-label classification, we define accuracy for them as:

$$a = \frac{T \cap P}{T \cup P}$$

where T is the set of ground truth and P is the set of predicted labels for all instances.

For Localization and Device Activity services, we measure the mean absolute prediction error and define accuracy as

$$a = \frac{\# \text{ of instances where estimated value} = \text{ground truth}}{\text{Total \# of instances}}$$

For Social Context Recognition service, we measure the mean relative prediction error and define accuracy as

$$a = 1 - \sum_{i=1}^N \frac{|\text{estimated value} - \text{ground truth}|}{\text{ground truth}}$$

Table 6 shows the overall accuracy of all the services.

6.3.1 Environmental Context and Physical Activity Recognition Services

As shown, the environmental context recognition accuracy is higher than IODetector[27] which reports an overall accuracy of 88%. The physical activity recognition accuracy is more than CenceMe [21] for all the common activities detected. It is comparable to Jigsaw [20] though it detects a slightly higher number of activities than SenseMe. Tables 4 and 5 show the confusion matrices for environmental context and physical activity recognition on test data. We believe these results can be improved even further by using more sophisticated classification and smoothing techniques.

6.3.2 Device Activity Recognition Service

The number of tasks that a user performs on a smart device during the day can be huge and maintaining a detailed journal for each of them can be quite burdensome. Moreover, because our temporal resolution is a minute, we do not capture device activities that last less than that. Hence, we did not ask the subjects to maintain detailed logs for every task that they performed on their device. Instead, we asked them to look at the visualization periodically to check the accuracy of the device activity and log that. Based on their logs, it was extremely accurate.

6.3.3 Localization Service

For all the subjects, the locations recognized by SenseMe were same as or close to their actual locations in both indoor and outdoor environments. This is mainly because the accuracy of this service is directly dependent on the technique being used for localization. Reverse geocoding is usually not 100% accurate as it is often difficult to resolve every latitude/longitude to a logical address. For Locus, the floor accuracy is approximately 99% and it achieves room level accuracy on the floor.

6.3.4 Social Context Recognition Service

Most users logged an estimate of the number of people around them and results from this service matched that reasonably. However, a limitation of determining social context using bluetooth is that it is constrained by the distance over which bluetooth operates. As a corner case, one of the subjects mentioned that he was at a wedding with several people in a large hall and SenseMe recognized the people on his table only. Though co-location might be a better alternative for determining social context, it requires location sharing by the users which they may not consent to. Hence, it is essential that a trade-off between accuracy and privacy is main-

tained. Since bluetooth performs reasonably well (being off by a small margin only) we have used it in our current implementation.

6.4 Qualitative Results

6.4.1 General, scalable and universally applicable

SenseMe has been tested on 15 subjects with varied schedules and mobility patterns. All the subjects carried devices made by different manufacturers such as Samsung, HTC, LG etc. and running different versions of Android OS ranging from 4.0 to 4.2. SenseMe ran without any major errors on most of the devices as it is based on techniques that are independent of device or manufacturer.

6.4.2 Minimum latency, robustness to network failure, and privacy preserving

In SenseMe, all computation and processing is carried out on the device and it does not require an external server. As a result, there is a minimum latency of a minute (which is the smallest granularity of computation). Only localization needs a network or data connection, but the localization service pipeline ensures that the system carries out opportunistic localization. Moreover, the user's data is kept private and confidential on the mobile device and is visible only to him/her, thus, mitigating privacy concerns.

6.4.3 Non-invasive and calibration-free

SenseMe is non-invasive i.e. it can easily run in the background in order to collect and process user's data without the need for any intervention. Also, it performs context and activity recognition using techniques that are agnostic to orientation, body position, time or weather, and hence, no calibration by the user is required.

6.5 Resource Utilization Results

We used two methods to determine SenseMe's battery consumption⁵:

6.5.1 Measuring the Average and Maximum % of battery consumed

For the average case, we measured the battery consumption of SenseMe while it was running in the background continuously for 24 hours on a user's device while he went about his daily activities.

To determine the maximum limit, we measured the battery consumption of using the GPS alone, which can be the most power consuming component of our system, in the worst case scenario. This, of course, varies with each user since the mobility and usage patterns can be quite different for everyone. Hence, we first determined the fraction of time the GPS was sensed during the entire run of SenseMe. Our analysis indicated that as a worst case, the GPS was actively sensed for 15% of the total time for which SenseMe ran, mainly due to an effective duty cycle that uses three different power conservation techniques. Thus, we estimated the maximum energy consumption of SenseMe by sampling the GPS for 15% time of a continuous run of 24 hours i.e. 3.6 hours, on battery without charging.

6.5.2 Measuring the average runtime of battery without charging

We tested the average runtime of the device battery (without the need for recharge) in a day when SenseMe was running in the background on it. The screen was set to a low brightness level since the screen display can consume a major chunk of the battery.

Table 7 summarizes the results of battery consumption and runtime on three different devices. All the devices were used moderately while these experiments were conducted. As shown, the average and maximum battery consumption were at most 40% and

⁵The actual battery consumption often depends on device usage and its age.

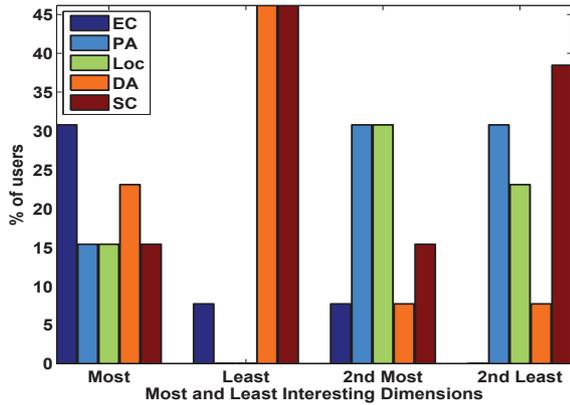


Figure 7: Dimensions users found most and least interesting

the battery lasted for more than 24 hours during a continuous run of SenseMe without requiring a recharge.

6.6 User feedback and Survey

On conclusion of each of the live deployment periods, we held a short interview with each subject to discuss which dimensions they found useful and interesting and to evaluate their user experience with SenseMe. We also wanted to uncover any issues with the system and ask them for open-ended valuable feedback.

6.6.1 Most and Least Interesting Dimensions

Figure 7 shows the plot for the dimensions users found most and least interesting (EC = Environmental Context, PA = Physical Activity, Loc = Context-aware Location, DA = Device Activity and SC = Social Context). As shown, a significant proportion found the Environmental Context dimension to be most interesting. Two of the subjects even referred to this dimension as “Intriguing” and “Insightful”. A majority of the users found the Device Activity and Social Context dimensions to be least interesting.

6.6.2 Perceived accuracy

In order to quantify the subjects’ perceived accuracy of the system, we asked them to look at the visualization at least once a day, during the deployment period, in addition to keeping a journal of their day. This was done in order to help them gauge whether the captured information represented their daily life log accurately. During the interview, the subjects rated SenseMe’s accuracy on a Likert scale from 1 (Very Accurate) to 10 (Very Inaccurate). The average score across the 15 subjects was 2.2 suggesting that the system was highly accurate in capturing their daily life log.

6.6.3 Feedback about benefits and insight

One of the subjects acknowledged that the multi-dimensional information could “*help me identify patterns which I didn’t realize before*”, indicating the benefit of its temporal aspects and long-term continuous usage. Another subject wanted to employ the location and physical activity dimensions for increasing his productivity as they could “*help me determine if I followed my schedule.*”

Some subjects expressed the desire to use the system in a more reflective manner. One subject mentioned that she would like to use the environmental and social context dimensions in a *persuasive* manner to help ensure she gets outside and interacts with people in times of heavy and prolonged workloads. Reflecting on the information captured during her two-week user study period, she noted “*Man, I live in a box...*” Another subject, emphasizing the reflective value of the device activity dimension, stated, “*It could*

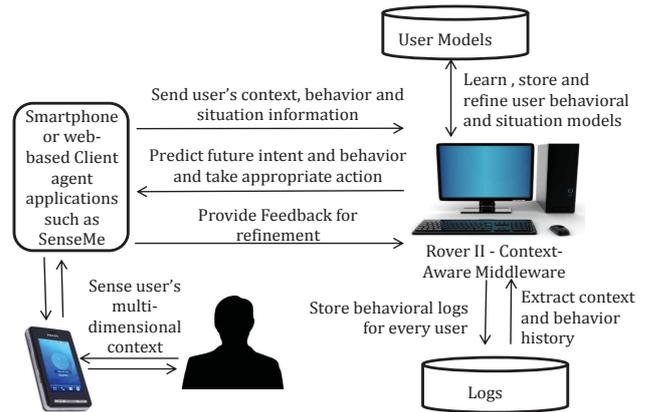


Figure 8: SenseMe acting as a client agent to a context-aware middleware in the future

help me find the apps I use way too much and also the amount of time that I wasted.”

7. SENSEME API

We have created a SenseMe API which will be made available to developers for building context-aware applications using the SenseMe system library. The API allows developers to obtain the values of one or more of the context and activity dimensions within a window of time, listen for context changes, and set up rules to trigger actions on context changes.

8. CONCLUSION AND FUTURE WORK

In this paper, we presented SenseMe - a system that leverages the smartphone and its sensors in order to perform continuous, on-device, and multi-dimensional context and activity recognition for a user. It achieves this in a robust, automated, scalable, power efficient and non-invasive manner. The system recognizes five dimensions of a user’s situation at any time instant - environmental context, physical activity, location, device activity and social context, to paint a context-rich picture of the user. The system also displays the recognized situation dimensions to the user via a proof of concept visualization called SenseMeVis. We evaluated the system extensively against several quantitative and qualitative metrics with the aid of 2 two-week long live deployments involving 15 participants. We demonstrated improved or comparable accuracy with respect to existing systems without the need for any user calibration or input.

We now plan to employ more sophisticated methods for recognition and smoothing and also implement the features suggested by our users. Additionally, we plan to enhance the system’s ability to detect other context and activities and with a finer granularity. In the near future, we would work on displaying periodic context and activity summaries to users.

In our broader vision for the future, the SenseMe system would function as a client agent application for a context-aware middleware such as Rover II [11]. As shown in Figure 8, a client agent application such as SenseMe would sense the user’s temporal and multi-dimensional context and activity information. This would be aggregated over a period of time in order to generate the user’s contextual history. The context-aware middleware would utilize this history to learn and store the user’s behavioral and situation models for predicting future behavior. This would further enable the context-aware system to act proactively on the user’s behalf in anticipation of the user’s future goals and intentions without explicit

requests. The system would also refine these user models periodically based on user feedback.

Some of the potential usecases include modeling correlations between the different dimensions being captured (for instance, between environmental context as well as physical activities etc.) This, in turn, can further enhance physical activity recognition by making it context-aware based on the type of environment (indoor/outdoor). Another potential user model could be determining the most frequent locations and travel paths of a user in order to predict user's transportation patterns. The device activity recognition can be further utilized to infer and predict the categories of tasks (gaming, media etc.) that users engage in most frequently and in specific contexts. For instance, watching Youtube when they are indoors and relaxing, as opposed to using Google Maps when they are outdoors and driving. Higher level context such as a user is 'Sleeping', 'In a meeting' or 'In a party' can be inferred based on location, co-location (location shared with other users) as well as social context. In the long run, such systems can have a profound impact on research areas such as proactive context-aware computing, persuasive computing, pervasive healthcare etc.

9. REFERENCES

- [1] Android location strategies. <http://developer.android.com/guide/topics/location/strategies.html>.
- [2] Android sensors overview. http://developer.android.com/guide/topics/sensors/sensors_overview.html.
- [3] Bluetooth baseband. <https://www.bluetooth.org/en-us/specification/assigned-numbers/baseband>.
- [4] Dilution of precision. http://en.wikipedia.org/wiki/Dilution_of_precision_%28GPS%29.
- [5] Nmea. <http://www.gpsinformation.org/dale/nmea.htm>.
- [6] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.
- [7] N. Aharony, W. Pan, C. Ip, I. Khayal, and A. Pentland. Social fmri: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing*, 7(6):643–659, 2011.
- [8] G. Ananthanarayanan, M. Haridasan, I. Mohamed, D. Terry, and C. A. Thekkath. Startrack: a framework for enabling track-based applications. In *Proceedings of the 7th ACM international conference on Mobile systems, applications, and services*, 2009.
- [9] P. Bahl and V. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 2000.
- [10] F. Ben Abdesslem, A. Phillips, and T. Henderson. Less is more: energy-efficient mobile sensing with senseless. In *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*, 2009.
- [11] P. Bhargava, S. Krishnamoorthy, and A. Agrawala. An ontological context model for representing a situation and the design of an intelligent context-aware middleware. In *Proceedings of the ACM Conference on Ubiquitous Computing*, 2012.
- [12] P. Bhargava, S. Krishnamoorthy, A. K. Nakshathri, M. Mah, and A. Agrawala. Locus: An indoor localization, tracking and navigation system for multi-story buildings using heuristics derived from wi-fi signal strength. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 212–223. Springer, 2013.
- [13] G. Castelli, A. Rosi, M. Mamei, and F. Zambonelli. A simple model and infrastructure for context-aware browsing of the world. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications*, 2007.
- [14] W. Griswold, R. Boyer, S. Brown, T. Truong, E. Bhasker, G. Jay, and R. Shapiro. Activecampus-sustaining educational communities through mobile technology. *University of California, San Diego, Department of Computer Science and Engineering, Technical Report*, 2002.
- [15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.
- [16] J.-R. Jiang, B.-R. Lin, and Y.-C. Tseng. Analysis of bluetooth device discovery and some speedup mechanisms. *Journal of the Institute of Electrical Engineering*, 11(4):301–310, 2004.
- [17] M. B. Kjærgaard, J. Langdal, T. Godsk, and T. Toftkjær. Entracked: energy-efficient robust position tracking for mobile devices. In *Proceedings of the 2009 ACM international conference on Mobile systems, applications, and services*.
- [18] P. Klasnja, S. Consolvo, T. Choudhury, R. Beckwith, and J. Hightower. Exploring privacy concerns about personal sensing. In *Pervasive Computing*, pages 176–183. Springer, 2009.
- [19] J. Krumm and R. Hariharan. Tempio: inside/outside classification with temperature. In *Second International Workshop on Man-Machine Symbiotic Systems*, 2004.
- [20] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, 2010.
- [21] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008.
- [22] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008.
- [23] L. Ravindranath, C. Newport, H. Balakrishnan, and S. Madden. Improving wireless network performance using sensor hints. In *Proceedings of USENIX conference on Networked systems design and implementation*, 2011.
- [24] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *IEEE First Workshop on Mobile Computing Systems and Applications (WMCSA)*, 1994.
- [25] D. Wagner, A. Rice, and A. Beresford. Device analyzer: Understanding smartphone usage. In *10th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Tokyo, Japan*, 2013.
- [26] M. Youssef, A. Agrawala, and A. Udaya Shankar. Wlan location determination via clustering and probability distributions. In *Proceedings of the 2003 IEEE International Conference on Pervasive Computing and Communications*.
- [27] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen. Iodetector: A generic service for indoor outdoor detection. *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems*, 2012.
- [28] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th international ACM conference on Mobile systems, applications, and services*, 2010.