Research Article

Catch You as I Can: Indoor Localization via Ambient Sound Signature and Human Behavior

Junzhao Du, Wenjing Chen, Yuewei Liu, Yawen Gu, and Hui Liu

Software School, Xidian University, Xi’an 710071, China

Correspondence should be addressed to Junzhao Du; dujz@xidian.edu.cn

Received 11 July 2013; Accepted 13 September 2013

1. Introduction

Mobile phones are gradually becoming powerful platforms for people-centric computing. As the sensing device has become steadily sophisticated, strong perception capability has been employed on the mobile phone, which ultimately makes it easier to comprehend the location and offer meaningful upper layer service for the users. A variety of applications are on the rise, many of which utilize location information on the phone [1–3]. Localization has been widely adopted in our life and work, and through it we can get shopping lists on a mobile phone when the phone detects the nearby Wal-Mart [4]. We found that all these applications could provide location results without any other information. In addition, they cannot deeply sense the relation between the user’s location and their surroundings. Thus, our goal is to design an efficient surrounding-aware platform in order to provide an accurate recognition of the status for mobile phone users, and at the same time the monitor (such as the boss, teacher, and parents, who have gotten the permission from users) can obtain the active state information by our surrounding-aware platform.

Intuitively, the problem of location recognition could be solved by taking advantage of the intensively investigated localization technology based on the wireless signal, such as Wi-Fi, GSM, and GPS. As far as outdoor localization is concerned, GPS has provided ideal recognizing accuracy. However, no current technique has offered a perfect solution to the problem of indoor localization. Limited by the complex structure and intensive EMI inside buildings, the wireless signal strength changes irregularly so that the sensing device cannot give a reasonable prediction of the location. It has been proved by related experiences [5] that, by location techniques based on the GPS/GSM, it is infeasible indoors by containing two larger location errors to distinguish two relatively closer indoor locations. Alternative Wi-Fi-based schemes (RADAR, Place Lab, SkyHook, etc., [6–8]) offer a better accuracy for indoor localization, but generally, these schemes only supply a user’s physical positions; however,
the user's surroundings information and the status of user cannot be obtained.

So far, there have been some researches on the localization method integrated with the mobile phone's sensing device. SoundSense [9] collects sound data by the microphone and uses time-domain and frequency-domain feature of the sound, but its main use is to detect the sound event. Batphone [10] brings forward a kind of sound fingerprints to conduct localization, which results in an accuracy of 70% in real situations. Peer assisted localization approach [11] obtains accurate acoustic ranging estimates [12] among peer phones, and then maps their locations jointly against the Wi-Fi signature map subjecting to ranging constraints. Besides, there are other researches which employ the location technique of combining Wi-Fi with multiple sensors, for instance, SurroundSense [5]. However, SurroundSense's location result is which store you are in, such as determining if it is a bookstore or a coffee house where a user is located. The accuracy will be greatly decreased when two adjacent stores are relatively similar.

Above all, we consider that when a user is supposed to appear in one place at some specific time, our system cannot only acquire the user's logical location, but also acquire what the user is doing by sensing user's surroundings information. Also the fact that the user's location is known by others will be considered as invasion of privacy, but in certain conditions, people's location can be shared with others for improving efficiency and safety. For instance, teachers concern about if their students attend class punctually; employers care about if their staff members start work punctually.

Our system is to achieve accurate indoor localization and infer the user's status through deep perception. Of course, translating this idea-sketch into a functional system entails a number of system design challenges presented by mobile phones.

(1) Indoor or Outdoor Detection. Because most of the users are located indoors (the outdoor users are not ruled out), the system should locate the user in general by outdoor localization techniques, and then detect if the users stay indoors using the Wi-Fi localization techniques. When the users stay outdoors, the GPS can position the location exactly; when the users stay indoors, we should use outdoor localization techniques first to return the approximate location of users and then combined with indoor localization to conduct accurate indoor localization.

(2) Local Position Pinpoint. We need to carefully identify the correct position result with the ambient information. It is necessary to consider the effect of the room's environmental feature and the variable factors on localization, such as different marks of people's sound information.

(3) Chat Keyword Sensing. On the basis of localization, we gather sound information through the microphone and take advantage of the voiceprint feature contained in sound information to infer the person talking with the user and perceive and mark what people said depending on the keywords in the sound information, and then deduce what the user does in current indoor environment.

(4) Sensing and Localization Assembling. We use different localization techniques and different sensing devices to collect various environmental information, an effective information assembling strategy is required to solve the puzzle of combining different pieces of information from multiple environment fingerprints to obtain the user's right information on location and the perception information, thus deducing where the user is and what he is doing.

(5) Reporting and Sharing. When the monitor needs to supervise a user's location and behavior in some period, we need to put the schedule of the user in the backend server. Once the sensing and deduced results returned by the system match the schedule, it is indicated that the user has completed the regulated thing as prescribed.

In this paper, we develop practical solutions to deal with such challenges. In particular, we extract unique identifiable fingerprints of indoor environment and utilize the GSM/Wi-Fi signal to determine the user's location. When the rough result obtained by the users is an indoor scene, we need to have further work on how the microphone in the mobile phone collects the indoor ambient environment and the talker's sound. When collecting the sound information on the current scene, it is necessary to consider how to conduct blind source separation for sound information, that is, to separate ambient voice from sound for identifying the ambient scene afterwards. Based on the sound collected by the microphone, the indoor fingerprint, voiceprint features, and keywords of current environment can be obtained. In separation of the sound signal, we not only need to focus on how to use the mobile phone to collect sound files to make sure that the size of the audio file is decreased as much as possible and not only impact the recognition accuracy as much as possible, but also choose the lightweight feature extracting method to fit the mobile phone's low computing capacity and small storage area. We try to exploit the sparseness in speech to extract frequency-domain acoustic features inside a smartphone, when the sampling rate is as high as 8 KHz. We propose an efficient and robust AdaBoost ensemble learning algorithm for having a comprehensive classification of outdoor localization information, voice information, the talker's voiceprint information, and keywords obtained from environment. In order to supervise if a user's behavior is as prescribed at the regulated time and location, we need to save the user's schedule to the backend server in advance. When the system returns location perception information, we need to match localization results with the schedule by using the matching method. Finally, we are then able to utilize the ensemble learning algorithm to accurately get the user's location information and determination information on whether the user reaches the particular location punctually.

We consolidate the techniques above, and then implement the architecture of the prototype system and algorithms with the Android platform using three types of mobile phones (Samsung Galaxy S2 i9100, HTC One S, and HTC Sensation G14). We also profile resource consumption in terms of the CPU and memory usage and evaluate the performance of the algorithms using real world ambient data sets through our system. We collect training data at 61 different sites and evaluate the accuracy with 100 students over a two-month period of experiments. As a result, the mobile phone scheme...
can find location with a detection accuracy of 92% and identify the classroom with an accuracy of up to 90%.

This paper is structured as follows. In Section 2 we provide the impacts of phone context and a target of our architecture. Section 3 describes the design of our prototype system, particularly in the architecture of our system. Section 4 evaluates the software system and analyses the results achieved from simulation. We present related work in Section 5 and conclude the paper in Section 6.

2. Preliminary and Motivation

In this section, we first review the impacts of the phone context, which serves as the basic challenge and the reason why the microphone is used as the basic sensor instead of others. Then, we highlight the target and the main challenges entailed to turn the principles into a practical system.

Mobile phones are pervasive and qualified for identifying the sound events around us in daily lives. However, phones are primarily designed for voice communication and present a number of practical limitations. We carry phones in different ways, for example, in the pocket, on a belt, in a purse, or in a bag. The location of a phone with respect to the user's body where a phone is used, and the conditions under which it is used, are collectively referred to as the phone context [9].

The phone context presents a number of challenges to building a robust location tracking system using appropriate sensory information. Using the microphone as the sensor of the system alone is mainly because sound is not easily affected by the phone context change compared with other sensors. IODetector [13] detects the indoor/outdoor environment by light sensors. But, we find that light sensors are easily affected by the phone context. For example, in the same office, the light intensity values are different when the phone is in the hand or in a pocket. The sound as a prevalent signal in nature, not only can reflect the characteristics of places and speech but also is robust to the phone context. Acoustic signals could be obtained at any moment when the system is functioning in spite of challenging external conditions such as poor lighting or visual information. Besides, they are relatively cheaper to store and compute than visual signals. Figure 1 is the experiment which can prove our argument.

Figure 1 shows the sensing effects of different context of the user's phone. In case A, the mobile phone face to the source, while in case B the source is away from the phone. In case C, the phone is in the hand of the user, while in case D, the user puts the phone in the pocket. From Figure 1, we can know that the light intensity changes a lot in different contexts. On the contrary, audio signals change little. As shown in Figure 1, the intensity of the light differs greatly when the mobile phone is in hand or in the pocket, whereas voice behaves distinctively. So, it's easy to conclude that the influence of context on light is far greater than on sound. Therefore, we choose the phone's microphone as the primary sensor to collect environmental information.

From the experiment, we see that there are linear relations between the signals from different phones at the same place, and the sound quality characteristics and spectrum are extremely similar. Therefore, by identifying the linear relationship, we can design a robust sensing system dealing with complicated phone contexts.

3. System Design

Though our idea is intuitive, the design of such a system in practice entails substantial challenges. In this section, we first
present the main components of the system, and then we characterize the challenges in the design and implementation and introduce several techniques for dealing with them.

3.1. System Overview. Figure 2 is the architecture in the broad design space. We shall describe the high level flow of information through this architecture, and will present the internal details later.

Data Collector. This is a collection of modules, each of which determines the fingerprint of the current location by acquiring data from necessary sensors and by using the necessary inference algorithm. As depicted in Figure 2 (left top), a user collects and uploads the sensing information to the backend server by the smartphone. When the user wants to know his/her location, the smartphone will collect the surrounding sound information and the GSM/Wi-Fi signal which will be fed back to the server for processing. Since the location of the smartphone user is not limited in a certain area, the system should determine the user’s coarse location. If the general location is far from the location the user should be in, then the system will determine that the user does not follow with the schedule. Ideally, the mobile phone of the user automatically performs the data collection and transmission.

We have measured energy consumed by the Wi-Fi and the microphone in an Android phone. Table 1 shows the energy consumption of the combination of different sensors. The key observation, which we expect to hold for other platforms and sensors implementations as well, is that energy consumptions of various sensors varied by an order of magnitude. This alludes to the potential savings in our system when we choose as energy-efficient a sensor as possible and make the accuracy high. The length of acoustic sample is a very important parameter which can influence the tradeoff between recognition accuracy rate and system energy consumption. If the sample becomes longer, the system will cost more energy to sample recording, feature extraction, and classification. If we can choose a reasonable length of the audio sample, we will save much more energy and achieve a higher recognition accuracy at the same time. Hence, we divide the sample into different lengths, 5 s, 10 s, 15 s, 20 s, 25 s, and 30 s, and use them to recognize the scene. As shown in Figure 3, the accuracy of recognition obtained by using 15 s and 25 s samples is almost the same. Because using 25 s samples will consume more energy, we conclude that the recording time should be 15 s which can make a tradeoff between the energy-efficiency and accuracy by experiment.

Backend Server. We transfer most of the computation burden to the backend server where the uploaded information from the user is processed and the requests from the monitor are replied.

In order to run the system, we need to build a database to store environmental sound information, voice information, and Wi-Fi information. When the system is operating, the background server will process Wi-Fi, GSM, voice, and environment noise information uploaded from the user.
After receiving the information, the system will analyze Wi-Fi/GSM information data first, and judge whether the user is within a geographic range, and then estimate which room the user is in according to the voice information. Besides, we want to get the schedule of the user in advance, so that the system will compare the result with the schedule of the user, and obtain the user's status. Moreover, since the system uses the multichannel to identify the location of the user when recording attendance data, it will decrease the possibilities of cheating and errors in recognition procedure tremendously.

The server is composed of several modules. A Data Collector gathers the data from the user's smartphone and undertakes preprocessing tasks. The preprocessed data is forwarded to the Fingerprinting Factory, which is located roughly with the Wi-Fi/GSM and then utilizes voiceprint recognition and text categorization to realize the scene classification and behavior inference of the user. The features are then compared with the Data Warehouse which stores the training samples and the schedule, and finally gets the result of tracking the location for the user to query.

**Monitor User.** The monitor user can send the request to query the whole day activities of a user. Besides, they also have the authority to acquire the user's current location information and to know whether the user is at the place following the schedule.

3.2. Data Preprocessing. Data preprocessing is mainly done in the data collector module. The first step of our system is collecting and recording ambient sound, speech, and Wi-Fi/GSM. To further understand the information revealed in these samples, we should preprocess the sound data and extract the ambient sound and speech from the acoustic sample. The goal of preprocessing is to reduce the data volume that needs to be transmitted.

Sound processing usually starts by segmenting the audio stream from the microphone into frames of the same duration. Features of classification are extracted during processing either from an individual frame or from a window which has $n$ frames. In our system, classification is performed with respect to the complete $n$ frame window, not simply on any individual frame. Not all frames are considered for processing. Spectral entropy and energy measurements are used to filter frames that are silent or are too hard to conduct classification accurately due to the context (e.g., far away from the source or muffled in backpack).

The first step of our system is to record an audio sample of a predefined length. To further understand the information revealed in these samples, we segment the audio stream into frames of the same duration. Segmenting the audio stream into uniform frames is a common practice for feature extraction and classification. The frame width (i.e., duration) is a key system parameter that needs to be optimized because it should be short enough so that there are not drastic changes in audio content, and meanwhile long enough so that the characteristics signature of the sound can be captured. Existing researches usually exploit frames that overlap with each other so as to capture subtle changes in the sounds more precisely. However, this may cause the overlapping pieces of audio data to be processed multiple times. Given the resource constraints of the phone we use independent non-overlapping frames consisting of 256 sampling points, that is, about 32 ms, at an input sampling rate of 8000 Hz. After segmenting frames, we multiply each frame by a window function vector, which reduces the signal magnitude near the frame boundary.

As the sound signal we collected contains environmental ambient noise and voice, we have to separate background noise from voice. The separated noise is used for the identification of the environment and voice for the speaker's voiceprint recognition and keyword recognition. During the speech processing, we need to determine whether the audio data contains voice, and only in the case with voice can we continue voiceprint recognition and speech identification.

In our daily life, the place people frequently visits will have its own unique background sound. In a classroom there exists a teacher's voice; in a lab, the keyboard or other sounds; in a plant, the roaring sound of the machines; and in a shop, its own specific background music and so on. During supervision, our system will periodically record voice segments and analyze the specific place from which they come. More than that, users will encounter certain people in their everyday lives or do a specific thing with others. By analyzing the speaker's identity or the content of his/her speech, the system can infer the user's current location. However, not all sound clips contain clear voices, but our system will filter the voice according to a certain rule, namely, the Voice Activity Detection (VAD). Some papers propose to use multiple features in combination with some modeling algorithms such as CART [14] or ANN [15]. However, these algorithms add up with the complexity of the VAD itself. Some papers put up noise estimation and adaptation methods for improving VAD robustness [16], but these methods are computationally expensive. Since our VAD is implemented on the mobile phone, we need to take into account the mobile's low processing capacity and our demand for quickly getting the VAD results, so we use the typical method.
Figure 4 shows three audio files, Figure 4(a) being the noise sound; Figure 4(b), the sound that contains the clear vocal sound in classroom; and Figure 4(c), the sound with extremely high noise. We noticed that in most cases, the ambient noise volume level does not exceed 0.3 (noise volume level ∈ [0, 1]), while the clear vocal sound often exceeds 0.4. In our system, we count up the samples whose volume level exceeds a certain threshold to determine if the voice segment has a clear vocal sound, and then determine whether it needs voice recognition. If there exists a certain vocal sound and the sample level is not up to the threshold, it is indicated that the human voice is not very clear and should be ignored. Conversely, in some extreme conditions such as that in Figure 4(c) which is recorded at a bar party, loud music has overshadowed any voice, so most of the sampling volume levels are beyond 0.4. If calculated by the previous method, such an audio can also be classified as a clear vocal audio. Considering this extreme case, there are two solutions. One is to simply ignore this extreme situation, since in this case we cannot obtain any useful results after speech recognition. Without recognizing the text, the system does not carry out the subsequent text categorization. Since it will not affect the positioning accuracy, we can simply ignore this situation right away. However, considering the demand that the system in the phone has to enhance operational efficiency as much as possible and reduce unnecessary steps, the first situation is somewhat irresponsible. As we all know, there will be pauses in a human speech, and it will be reflected in the audio signal despite of the short pause time, as shown in Figure 4(b). But, the audio signal filled with noise is generally continuous and is composed of numerous signals, so these signal samples of the high energy level is very dense, as shown in Figure 4(c). To prove the above point of view, we count the signal level density respectively in Figures 4(b) and 4(c). In Figure 5, we can easily observe that the sampled sound clip level in noisy environment is concentrated mostly in the high level portion.
3.3. Fingerprint Extraction. The Fingerprinting Factory receives the type of processed data (Wi-Fi, accelerometer) and extracts the fingerprints. The fingerprints are distributed to the Behavior Identifying module and the Scene Classify module. This module performs a set of appropriate operations, including computation of the RSSI value of Wi-Fi APs, ambient sound feature selection, and voiceprint selection.

3.3.1. Treatment of the Wi-Fi/GSM: Which Room Are You in? In the Fingerprint Factory, it is inevitable to compare the information successively in the large database regardless of using the pure Wi-Fi or pure microphone to find location. Since many of them are redundant, we need a simple approach to finding the general location which will decrease the time of comparisons and thus reduce the complexity. The collected data includes the phone's (GSM-based) physical coordinate, \( L_{\text{GSM}} \). The \( L_{\text{GSM}} \) is a (latitude, longitude) tuple accurate to around 150 m. GSM base stations are utilized for localization which can get the latitudinal and longitudinal information. Although the accuracy of the GSM is not high, a shortlist of location information can be provided. The system can pick out the suitable scenes from the database for further localization.

The backend server needs to maintain a database that stores the Wi-Fi and audio signal fingerprints that are collected from different rooms in our building. War-driving along one hour in a classroom, the mobile phone normally captures a group of Wi-Fi and audio signals at one time. But even if collected from the same place, signals might be different from time to time. To improve the robustness of our system, we combine the Wi-Fi with the ambient sound for making up the disadvantage. We use the RSSI values of Wi-Fi APs as feature [17]. For the system to remain robust under signal fluctuations (which alters the set of overheard APs), we only consider APs that are stronger than a threshold RSSI.

We compute similarity of two locations, \( L_1 \) and \( L_2 \) as follows. We denote the sets of Wi-Fi APs as \( A \). Let \( f_i(a) \) represent the RSSI of AP \( a \) at location \( L_i \), \( a \in A \). If \( a \) does not cover \( L_i \), then \( f_i(a) = 0 \). We now define \( S \) as similarity. If we want to know the similarity between locations \( L_1 \) and \( L_2 \), then we have

\[
S = \frac{1}{|A|} \sum_{a \in A} \min \left( \frac{f_1(a)}{f_2(a)} \right) \max \left( \frac{f_1(a)}{f_2(a)} \right).
\]  

If the Wi-Fi signals covering \( L_1 \) and \( L_2 \) vary greatly, it indicates that the Wi-Fi APs could not be received at either place or the signal in one of the places is too weak to be received. The sum of the results from (I) must be small, which implies that the similarity of the two locations is low. After acquiring the location similarity, the system makes a comparison between the calculated value and its preset threshold. If and only if the similarity is smaller than the preset value, the place can then be deemed to be recognizable and be added into the system landmark as the additional one. Figure 7 shows this tradeoff using traces from two buildings. We observed that 0.4 was a reasonable threshold, balancing the quality and quantity of Wi-Fi APs.

To manifest our standpoint, several tests were conducted on the floor of our laboratory. We collected the Wi-Fi information from three rooms, in the order of A, B, C, and back to A. There are 24 samples in each room, respectively. Figure 6 illustrates the specific locations of three places.
Figure 6: The positions of APs and three places (A, B, and C) of collecting Wi-Fi signals are shown on the floor plan.

Figure 7: Tradeoff between similarity threshold and number of Wi-Fi landmarks.

Figure 8: The similarity of Wi-Fi samples which are collected from A, B, and C shown in Figure 6.

Figure 8 displays the similarity of Wi-Fi signals collected from three rooms. The darker the color is, the higher the similarity of the Wi-Fi samples corresponding to the horizontal ordinate is. It can be clearly seen from the chart that the similarity among samples is extremely high in the same group. However, the similarity between samples 1–12 and 73–96 is not expected; although the samples are collected in the same room A, they are collected at different time periods. Moreover, the samples taken from different places vary greatly from each other in most situations. Nevertheless there are similar ones which appeared from time to time. For instance, samples 40–44 gathered from room B shows great similarity to the samples taken from room A. Given the above situation, we add audio indicators for fine positioning in order to acquire a higher accuracy.

3.3.2. Treatment of Sound: Who Is Talking? Where Is She/He Talking?

In this section, we extract the features of the ambient sound and speech in order to explore audio data to detect who is talking and what he/she is talking about. With this information, we can know where the user is and what he/she is doing with whom.

When we have used VAD to successfully detect the presence of voice in the audio, the separation of noise and speech can be conducted. We separate the ambient sound and speech based on the frequency-domain convolutive signal blind source algorithm. The observation signal is structured by the wavelet multiresolution analysis first, and then the separation is achieved between ambient sound and speech through the frequency domain independent component analysis [18–20]. We can extract the features respectively for ambient sound and speech after the separation.

In what follows, we discuss the spectrogram representation for the ambient sound and voice sound we use in our system.

Acoustic Background Spectrum (ABS) [10]. The ABS is a new ambient sound fingerprint. The first step is to record an audio sample of length $t_{samp}$. In the next several steps, computing...
the power spectrum of each frame which involves applying a Fast Fourier transform (FFT) of resolution $2 \times (n_{\text{spec}} - 1)$, throwing away the redundant second half of the result, leaving $n_{\text{spec}}$ elements, and multiplying the result elements by their complex conjugates, giving the power. $n_{\text{spec}}$ is the spectral resolution. After the operations, this ambient sound signal is transformed into a time-frequency representation called a power spectrogram. After the spectrogram is computed, we filter out the frequency band of interest by simply isolating the appropriate rows and apply a new method for extracting the noise-robust spectrogram summary vector. We accomplish this by choosing one of the smallest values observed for each frequency in the sampling window. We choose a value near the minimum, the 5th-percentile value ($p05$). Choosing the $p05$ value involves either sorting or using a linear-time selection algorithm (such as quick select) in each of the spectrogram rows. The final step in computing the ABS is to compute the logarithm of the spectrogram summary vector.

Sparse Mel Frequency Cepstrum Coefficient (sMFCC) [21]. The idea of the sparse MFCC algorithm is to compute a sparse approximation of MFCC features from a given frame of discrete time-domain signals $x_n$ of length $n$. The algorithm uses a modified version of the Sparse Fast Fourier transform (sFFT) [22, 23] as a subroutine. Like the sFFT to FFT, sMFCC is an approximation to MFCC. In this algorithm, the sparseness parameter $k$ is computed, which is one of the key parameters of the sFFT algorithm. Once the Fourier coefficients are obtained, we follow the standard procedure of MFCC [24]. Experiments show that sMFCC is up to 5.84 times faster than MFCC while its error is within 1.1%–3.9% that of MFCC.

In order to realize that our system can be adopted on any smartphone, we use Samsung Galaxy S2 i9100 and HTC One S to collect data samples. Considering that we need to transfer the samples to the backend server on the web and reduce the complexity of processing sound at the same time, we choose the 8000 Hz sampling rate to record voices. Figure 4(b) plots the raw audio signal segment in the time domain when the user is in a classroom. After we have separated the background noise from voice, we extract ABS as the feature vectors of our background noise. If there are voice signals in the recorded data, we also need to extract sMFCC as its feature vector. After extracting sMFCC feature vectors of audio data in classrooms and labs, respectively, the Vector Quantization then compares and analyzes these feature vectors, leading to the final answer.

3.4. System Classification Algorithm for Environmental Perception and Persons

3.4.1. Scene Classification Based on the Background Noise. When the user is in the period of being tracked, the mobile phone samples Wi-Fi and audio data and reports the information to the backend server. The GSM/Wi-Fi offers the location accuracy to around 150 m [5] to decrease the number of scenes. Then, the scene the user is in can be classified with audio data. In Section 3.3.2, we introduced how to extract the features of ambient sound and human speech. Additionally, the features should be analyzed by a classifier model and, after that, the location where the sound is collected can be determined. The scene classifying module is mainly responsible for the scene classification.

After the ABS room fingerprint is calculated, it can be compared with the previously-observed fingerprints to determine the location. We solve this classification problem via supervised learning. We assume that a database of room fingerprints is available with each fingerprint labeled with a room identifier. The problem at hand is to label the currently-observed room fingerprint. To do this, we choose the Multi-class Support Vector Machine (M-SVM), Naive Bayes Model (NBC), and Probabilistic Neural Network (PNN) to be the candidates and they all have their own characteristics. The M-SVM has good recognition ability for the non-linear input (e.g., linear characteristics will be destroyed by the mixture of many emergency events with environment). The SVM contains many kernel functions: Polynomial kernel function, Gaussian Radial Basis function, and Hyperbolic tangent Function. By choosing the reasonable kernel function, it can conduct classification in linear time [25], which provides the basis for efficient mobile phone classification. We try to use different kernel functions for sound classification. Through the experiment, we find that using the Gaussian Radial Basis function can lead to the highest classification rate. The advantage of the SVM method is to use a small training set; the disadvantages are that the theory only really covers the determination of the parameters for a given value of the regularization and kernel parameters and the choice of the kernel. Besides, the complexity of the SVM method is high. The Naive Bayes is suitable for the case where users will only provide a few labeled training samples. Therefore, the NBC is a simple and effective method for text categorization. For the PNN we can complete the training phase very fast, and after getting new data it does not need a repeated training [26, 27].

We choose the simple and fast Neural Network algorithm for comparing room fingerprints. In particular, we use the Probabilistic Neural Network for classification. Because our system has run online training, using the PNN can make the system rapidly respond to environmental changes.

The PNN based on statistical theory is of equal function to the optimal Bayes classifier, but it does not need the BP algorithm to compute reversal error propagation as the traditional multilayer feedback neural network does. Instead, the PNN computes the data forward totally, so it has the advantages of short training time, simple topological structure, easy algorithm design, strong fault tolerance, and so forth. The PNN is widely used in the field of modeling recognition and classification.

3.4.2. Speaker Recognition and Behavior Inference. When audio information is included in the acquisition sound data, we can extract the ambient noise from audio. We test the audio indication-based scene classifying method in various scenarios, and experiments show encouraging results for scene classification. We can apply this technique in any environment as long as noise exists. Through voiceprint
recognition for human speech in our records, we can determine if it is “mark people” who are related to the user currently. If a user, Alice, is talking with somebody, our system can recognize this if Alice is contacting specific people at regulated time according to the audio recognition of the people Alice is talking with, which is helpful to deduce what Alice is doing. For instance, if Alice is talking with her teacher, by using voiceprint recognition, our system can recognize that the teacher is doing is teaching the operation system course, and from the keywords of what the teacher is talking about through text classification method, it can be concluded that the teacher is having the class instead of chatting. The work of activity recognition through voiceprint recognition and text classification is completed mainly in the Behavior Identify module.

Voiceprint Recognition (VPR). In the backend server, sMFCC is extracted from speech sound data. The feature vectors are then sent to a classifier for classification and further analysis. Feature matching is referred to as the classification of the extracted features from individual speakers. The feature matching techniques used in speaker recognition include the Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), and Vector Quantization (VQ) [28].

Text Categorization. Text categorization techniques are used to classify text documents into categories or classes. For instance, in our system, if the customer says, “Yes, this algorithm is efficient,” we want the system to recognize the keywords and to infer the customer’s behavior with other information accordingly. In this paper, we use a machine-learning technique called Naive Bayes for the problem of text categorization. The Naive Bayes has the stable recognition rate and needs only small amounts of training samples for the estimated parameters of the classifier.

3.5. Algorithm for Comprehensive Localization Deduction. As mentioned above, different sensors of the smart phone collect sample data (Wi-Fi/GSM, ambient sound, human speech and text keywords) at a user configured rate and infer the location of user, respectively. The system aggregates the results from each sensor and then decides comprehensively the user’s ultimate estimation location. Further, we can determine if the user appears at the regulated place and time according to the audio recognition of the user. The system aggregates the results from the phone’s different classification modules, and observation results are sent to a classifier for classification and further analysis. During each iteration, the observation weights are updated based on training process, greater weights for an observation represent a greater classification difficulty.

Aggregated data from the phone’s different classification modules is fed into the Inference Module in each aggregation interval to make an inference decision. We use the AdaBoost [8], an ensemble learning algorithm, as our inference location classifier which resides entirely in the backend server. The AdaBoost combines an ensemble of weak classifiers together to form a single, more robust classifier. With this approach, we are able to train weak classifiers for each classification module’s results in our deployment and combine them together to infer location.

Using the AdaBoost, we incrementally build an ensemble of computationally inexpensive weak classifiers, each of which is trained from the labeled training observations of a single module’s results. Weak classifiers need only to make classification decisions that are slightly correlated with the ground truth; their capabilities are combined to form a single accurate classifier. The completed ensemble may contain multiple weak classifiers for the same module’s result; some module’s results may not have trained classifiers in the ensemble at all. The AdaBoost incrementally creates such result-based weak classifiers by emphasizing the training observations misclassified by previous classifiers, thus ensuring that the training accuracy is maximized.

We describe the AdaBoost training as follows. We define a set of locations \( A = \{a_1, \ldots, a_n\} \), different classification modules \( S = \{s_1, s_2, s_3, \ldots, s_m\} \), and observation results \( O_j \) for each module \( s_j \in S \), where each module has \( n \) training observations. The training output is an ensemble of weak classifiers \( H = \{h_1, \ldots, h_T\} \), where \( h_i \in H \) represents the weak classifier chosen in the \( i \)th iteration. We initialize a set of equal weights \( D_1 \) for each training observation, where during the training process, greater weights for an observation represent a greater classification difficulty.

During each iteration \( t \), we train a weak classifier \( h_{t,j} \) for each module \( s_j \in S \) using observations \( O_j \) and weights \( D_t \). We then compute the weighted classifier error \( e_{t,j} \) for each trained module classifier, adding only the module classifier to \( H \) which has the lowest weighted error. Before the next iteration, the observation weights \( D_t \) are updated based on the current weights and the misclassifications made by the selected classifier.

Given an observation \( o \), each weak classifier returns a probability vector \([0,1]^m\) with each scalar representing the probability that the current location is \( a_i \). To train a weak classifier \( h_{t,j} \) for each classification module \( s_j \in S \), we use a Naive Bayes model. With a weak classifier chosen for each iteration, the output of the AdaBoost classifier for each new observation \( O \) during the runtime is defined as

\[
h(o) = \arg\max_{a_i \in A} \sum_{t=1}^T \left( \log \frac{1}{e_t} \right) h_t(o, a_i).
\]
In (2), the activity probability vector for each weak classifier \( h_t \) is weighted by the inverse of its error \( \varepsilon_t \). Thus, the weak classifiers with the lowest training error have the largest weight in making classification decisions. To put it another way, the AdaBoost chooses the classification modules with weak classifiers that minimize the weighted training error, achieving a maximum training accuracy for all locations.

The data warehouse is mainly used for storing the trained data and schedule. When inference modules complete ensemble classification according to the training result and schedule, the monitors are capable of querying the information on the user in order to track the user.

4. Evaluation

In this section, we discuss the evaluation of our system. We implement a prototype system on the Android platform with different types of mobile phone. We collect audio data from 61 different sites over an 8-week period of experiments. Following this, we present the experimental methodology and detailed performance.

4.1. Experimental Methodology

**Mobile Device.** In the first half of the experiment, we implement the application on the Android platform and test its performance using three different types of mobile phones (Samsung Galaxy S2 i9100, HTC Desire S, and HTC Sensation G14). The Samsung Galaxy S2 i9100 has a 1GB RAM and dual-core 1.2 GHz Cortex-A9 processor, the HTC Desire S has a 768 MB RAM and 1 GHz Scorpion processor, and the HTC Sensation G14 has a 768 MB RAM and dual-core 1.2 GHz Scorpion processor. Application is independent of platforms. We believe that the proposed system can be simply implanted to other mobile computing platforms, such as Apple iOS and Windows Phone. In the later stage of the experiment, students in our class installed our application to help us test our system.

**Backend Server.** We implement the backend server in Java running on the HP z400 workstation with 6 GB memory and Intel Xeon W3503 processor. The scene classifying service can be deployed in a computing cloud for dynamic and scalable resource provisioning as well.

**Experiment Environment.** The campus has various indoor scenes including the dormitory, restaurant, classroom, laboratory, library, supermarket, bar, and even hospital. Such complex indoor environment is enough to make our system finish the whole experiment and get convincing experimental data. We set up a database, which contains the fingerprints (Wi-Fi signal and ambient sound) of 61 locations and voice. The demarcation between different scenes is mainly composed of the physical boundaries like walls, doors, and so forth. But some places which cover a large area like the supermarket or corridors do not have distinct physical boundaries, so we set a virtual boundary in this kind of place by the size of area or special environmental characteristics.

| Table 2: Performance of the vocal detection algorithm. |
|-----------------------------|-----------------|---------------|
|                            | Accuracy | Comprehensive accuracy |
| Quiet With vocal           | 97%      | 94%            |
| Quiet Without vocal        | 98%      |                |
| Noisy With vocal           | 89%      |                |
| Noisy Without vocal        | 95%      |                |

Experimental location scenes can be roughly divided into three classes: quiet scenes (laboratory, library, hospital, etc.), noisy scenes (dormitory, restaurant, market, bar, etc.), and scenes which have vocals from specific people (classroom, meeting room, etc.). All the scenarios are distributed in every zone of the campus or outside the university. The structure of the buildings which have these scenes is very complicated. Some environmental sounds from some places are similar while those from other places are entirely different. With the help of these characteristics, these indoor scenes can simulate every indoor localization scene in our real life, and thus detect the localization accuracy of the system.

We collect data from every scenario, get localization result, and calculate the localization accuracy. Synthesizing the results of each experiment, we can get an accuracy of approximately 95%.

4.2. Mobile System Performance

4.2.1. The Accuracy of Detecting People’s Sounds. During the 8-week experiments, we collected hundreds of audio files with 8000 samplings. Due to the uneven lengths of audio files, we divide the files into 15s audio fragments, which unconsciously expands the set of experimental data. In these data, some contain speech, while some are very noisy. If we use noisy audio data to automatic speech or voiceprint recognition, the system cannot get the results. In this case, the system must determine whether the audio data have vocal, no matter what we can hear from the audio files.

We test the vocal detection algorithm mentioned in Section 3.2. Audio samples can be divided into four categories: quiet with human sounds, quiet without human sounds, noisy with human sounds and noisy without human sounds. We conduct a survey about the accuracy of the system and detect human sounds in four categories, respectively. In Table 2, the algorithm can achieve a high recognition accuracy in quiet conditions. In noisy conditions, if audio samples do not have human sounds, the detection accuracy of the algorithm is still satisfactory. Only in the case of the noisy audio sample with voice, is the recognition accuracy slightly lower than that for other three cases. This is because the energy level of human sounds is just on the boundary where the algorithm can detect voice in audio samples. In this condition, ASR (Automatic Speech Recognition) and VPR (Voice Recognition) cannot always get right results. The audio sample is falsely recognized because it thinks the samples do not contain voice. This kind of audio samples accounts for a small part of all samples, and the fuzzy voice in this kind of audio will influence the subsequent processes. Hence, making
mistakes by our algorithm in this case would not influence the localization accuracy of the whole system.

The experimental results suggest that the audio-based method can effectively detect the voice when the audio signal is recorded by the mobile, no matter what the audio signal looks like.

4.3. Backend System Performance

4.3.1. Ambient Sound Recognition Performance. The key algorithm for indoor localization is the scene recognition through the environmental sound, whose accuracy directly impacts that of the system results. The audio data mentioned in Section 4.2.1 are used to measure the scene recognition accuracy. In ABS, we conclude that the algorithm can give a high scene recognition accuracy. Considering the nature of the system, each time we classify the scenes, the number of scenes that need recognition is five to eight. Classifying the scenes by the algorithm can lead to a high recognition accuracy, namely, more than ninety percent. However, the algorithm is sensitive to the sounds of people's words in the audio, and once a large number of people's sound signals occur, the recognition rate will drop accordingly. In order to make the system capable of classifying all the types of audio, blind source separation is used to deal with the audio files, and then the audio can decrease the effect of people's sounds on the recognition algorithm to a large extent. We have compared the capacities of system classification before and after separation through experiment.

In the experiment, we classified the sound samples into human sounds and nonhuman sounds and chose a number of samples from the two types as the test set. We can observe the influence of the human sound on the system by controlling the proportion of human sample sounds in the test set. As shown in Figure 9, the longitudinal coordinate indicates the accuracy of the system recognition while the abscissa shows the proportion of human sample sounds. When there is no human sample in the test set, the recognition rate effect of the algorithm can be quite remarkable, but with the increase of the proportion of human sample sounds in the test set, the accuracy decreases. When every sample contains the human sound, the recognition will drop to nearly zero. If we add the human sound peeling algorithm to the system, the recognition rate will remain high, which implies that eliminating human sounds is helpful to the system scene classification.

4.3.2. The Performance of the Voiceprint Recognition and Voice Recognition in Noisy Environment. Due to the fact that the condition under which the mobile phone collects the audio cannot be restricted, the recording sound will not be clear. Usually, the human sound audio contains noises either large or small. We have tested the system’s capability to recognize the audio voiceprints and the voices containing noise.

As illustrated in Figure 10, the abscissa is the dimension of noise in the audio, the longitudinal coordinate on the left is the recognition accuracy, and the right one stands for the sample number. The bars in Figure 10 mean the number of samples in each noise level and the lines mean the accuracy in each noise level. It is obvious that the recognition accuracy decreases with the increase of the audio noisy degree. When the audio is quite noisy, namely, more than 0.4, the accuracy of recognition is only 60% to 70%, which is unacceptable. At the same time, it can be seen that voice recognition has a still higher demand on the audio. When the noise exceeds 0.3, the recognition rate would decrease in an accelerating way. So by the use of the algorithm mentioned in Section 3.2 in the pre-control process, the system detects excessive noise in the audio, and then determine that the audio is incapable of recognizing voice or voiceprint.

Not only do we conduct a survey of system localization accuracy, but also of the number of samples of each noise level. The bar in Figure 10 shows audio sample distribution according to normal distribution. The noise level of most audio samples is lower than 0.3, with high noisy samples accounting for an extremely small part in the sample set. This result suggests that most audio samples can be used to the ASR or VPR for assisting system localization.

In Section 4.3.1, we mentioned that the clear voice in the sample would blight the system localization accuracy.
So we peeled the voice from the audio sample. However in this section we will make use of the voice to help localization, which seems contradictory, but in fact is not. Audio samples with human sounds would be handled by two methods. One is to peel the voice by the scene module; the other is to use the VPR and ASR by the Behavior Identifying module. The two methods are processed in parallel and without interference so that a higher efficiency can be obtained, and the voice and environmental sounds can be used to serve our system at the same time.

4.3.3. Text Categorization Algorithm Performance. By selecting the text categorization algorithm, we compare the classification accuracies of three kinds of popular algorithm with different corpus. The categorization algorithms we chose are the Naive Bayes algorithm, SVM algorithm, and KNN algorithm. The corpus we select are normally used to evaluate the text categorization algorithm. They includes the following.

(1) 20NewsGroup, which is a collection of 20,000 messages from 20 different newsgroups, with one thousand messages from each newsgroup;
(2) Industry Sector, which is a collection of about 10000 web pages of companies from various economic sectors. (There are 105 categories);
(3) WebKB, which contains about 8000 web pages collected from computer science departments of various universities in Jan. 1997. We use the version with the following seven categories: "student", "faculty", "staff", "department", "course", "project", and "other";
(4) IBMweb, which contains about 7000 web pages collected from http://www.ibm.com/, categorized into 42 classes.

Table 3 obviously shows that using the SVM algorithm can lead to a higher accuracy than using other algorithms. The system needs to render services in real time for the user, so the algorithm must calculate the result rapidly. In these three text-categorization algorithms, the Naive Bayes and SVM algorithm take the longest time (10s) and the shortest time (2s), respectively. Time consumption of the SVM algorithm can be accepted. In practical applications, it is easy to expand the training set by the SVM algorithm, which does not require lots of verbose and repetitive operations.

| Algorithm      | WebKB | IBMweb | 20NewsGroup | Industry sector |
|----------------|-------|--------|-------------|----------------|
| Naive Bayes    | 70.1% | 75.2%  | 91.8%       | 86.9%          |
| SVM            | 89.7% | 87.0%  | 94.3%       | 94.6%          |
| KNN            | 83.2% | 85.3%  | 92.5%       | 91.7%          |

Table 3: Performance of three kinds of text categorization algorithms with different corpuses.

4.3.4. Comprehensive Localization Performance. We integrate the results from each module and use the AdaBoost classifier to locate the scene. Our experiment focuses mainly on two aspects. One is whether the system can get a more accurate result by integrating the results from each module. The other is whether the system can keep a high localization accuracy, when there are more and more scenes in the database which need to be located.

The students in our class installed our system on their mobile phones. They helped us to test the performance of our system in the real life and we obtained the statistics. According to this statistic data, with the laboratory taken for example, we drew Figure 11 about system localization results. In Figure 11, the colorized lines in the floor plan mean the location areas we delimited, each color representing a location area. In other words, these location areas are the exact results which should be located by our system. The three colorized bars behind the floor plan show the system result during the user’s walk in the laboratory. From top to bottom are respectively expressed the correct results that the system locate, the result located only being located falsely. But our system can be used to locate each position precisely. Only at the boundary between two areas of corridors can the system get some wrong results. It is because there is no physical boundary between two areas at all and the areas near this boundary are very similar. If the users get close to this boundary, the system will locate the position in the opposite area. Because of similarity, both the results given by the system should be considered as correct for the users.

In a real system, more and more scenes will be added to the scene recognition library. Therefore, faced with the increasing number of scenes, that is, the increasing scale of the problem, our system is still able to maintain a high level of recognition. As shown in Figure 12, we compared the positioning accuracies of the system, the SurroundSense as well as the case using the Wi-Fi and sound alone. When the number of scenes increases, the recognition accuracy remains stable by using our system, the SurroundSense and the Wi-Fi, respectively. And, it will tend to go downward if we only use voice features for positioning.

The reason is that when the number of scenes increases, if we only use voice features to locate the position, the system has to match the characteristics of all scenes. The more the scenes are, the greater the probability of similar scenes is, and the recognition rate will certainly decrease. By using the Wi-Fi positioning alone, the recognition rate will decrease until the number of scenes reach a certain order of magnitude. It is because Wi-Fi positioning is based on the calculated similarity of the Wi-Fi signal, and the scenes that have high similarity are concentrated in several scenes within a certain range. After all scenes within the range are added to the database, adding other scenes will not affect the algorithm recognition rate. However, due to imprecise characteristics of the Wi-Fi positioning algorithm, the correct rate we obtained would decrease.
14 International Journal of Distributed Sensor Networks

Figure 11: An experiment trace in laboratory building.

![Figure 12: Recognition accuracy versus problem size](image)

**5. Related Work**

More recently, several sensor-aided localization approaches have been proposed. By making use of the accelerometer and compass sensor, Constandache et al. [29] first mapped a predicted user trajectory onto a pre-downloaded map, and then inferred the user’s current position. The main disadvantage of such a method is that small errors are accumulated over time, resulting in a significant position drift. Besides, we observe that adjacent stores typically offering different services leading to different background features such as lighting and background music. Azizyan et al. [5] developed the SurroundSense system to differentiate neighboring stores in a shopping center. In their system, ambient-sound is exploited to identify a store whereas its use is straightforward, that is, authors exploit the sound amplitude distribution to compute the loudness characteristics. Our follow-up preliminary experiments show that such a simple manner neglects valuable features of the ambient sound that could lead to better localization. A more complex ambient sound feature MFCC is explored by SoundSense [9], which used the sound sample of the microphone sensor in the iPhone to classify sound events. The SoundSense ignores silent sound samples because the authors claim that these samples cannot represent a sound event. However, in the context of indoor-localization, a quiet background setting is meaningful in inferring the user’s position, for example, in a library or in a self-study room. Hence, we investigate the effectiveness of MFCC in localizing users since typical sound events are also tightly associated with a specific place. Chu et al. [30] proposed to use the Matching Pursuit (MP) technique to enhance the performance of the MFCC in differentiating scenarios. However, their experiments were conducted in an ideal setting, where the sound recorder can capture a large range of the frequency (0–24 K) of the background sound while the typical microphone sensor can only capture a relatively small range (0–8 K). The difference of sound recorders significantly changes the pattern of a sound sample, which could affect during our test is only about 70% and thus that algorithm cannot be used alone.

Both our system and SurroundSense [5] can get a satisfactory result, and it is because we adopted a certain method to define the scope in the preliminary stage to exclude the interference of the algorithm caused by extra scenes. In SurroundSense, several locations are artificially set to a cluster. Positioning in this cluster is not a self-optimization process. Meanwhile, during the positioning, SurroundSense used other sensors’ data except sound, such as color and acceleration data, and in the case with the Wi-Fi and MIC the accuracy rate is only about 70%. In our system, we used the GSM positioning and Wi-Fi positioning technologies to screen several candidate scenarios, so that we would deal only with 5–8 scenes when we conducted indoor locating with sound which can lead to a very accurate and stable positioning result of about 90%.
the localization accuracy achieved by subsequent processing methods.

6. Conclusion

In this paper, we presented the idea of taking advantage of the Wi-Fi/GSM and the microphone to perceive and recognize indoor and outdoor environments and tried to infer the behavior of users. At the same time, the monitor can track the location and behavior according to the schedule in the backend server. The main idea is to fingerprint a location based on its ambient sound, human voiceprint, and speech. This fingerprint is then used to identify the user’s location and behavior. We did a large number of experiments by using the Android smartphone and taking account of some kinds of classifiers as well as testing multiple methods such as voiceprint, speech recognition, text categorization, and so forth. Last but not least, we got their performance in precision. As a result of accuracy experiments, we believe that our system is an early step towards a long-standing research problem in indoor localization. Further research on fingerprinting techniques, sophisticated classification, and better energy management schemes could make our system a viable solution in the future.

Acknowledgment

This work is supported by the NSFC under Grants nos. 60803152, 60903206, and 61100075, the Fundamental Research Fund for the Central Universities (K50510230004), the Open Funds of ISN National Lab under Grant no. ISN-9-09, the Research on Key Technologies of Electromagnetic Spectrum Monitoring Based on Wireless Sensor Network under Grant no. 2010ZX03006-002-04.

References

[1] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt, “Mi- croblog: sharing and querying content through mobile phones and social participation,” in Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys ’08), pp. 174–186, 2008.
[2] S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, G. -S. Ahn, and A. T. Campbell, “Metrosense project: people-centric sensing at scale,” in Workshop on World-Sensor-Web (WSW ’06), 2006.
[3] P. Mohan, V. Padmanabhan, and R. Ramjee, “Nericell: rich moni-toring of road and traffic conditions using mobile smartphones,” in Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys ’08), pp. 323–336, November 2008.
[4] T. Sohn, K. A. Li, G. Lee, I. E. Smith, J. Scott, and W. G. Griswold, “Place-its: a study of location-based reminders on mobile phones,” in Proceedings of the 7th International Conference on Ubiquitous Computing (ACM Sensys ’05), pp. 232–250, Tokyo, Japan, September 2005.
[5] M. Azizyan, I. Constandache, and R. R. Choudhury, “Surround-Sense: mobile phone localization via ambience fingerprinting,” in Proceedings of the 15th Annual ACM International Conference on Mobile Computing and Networking (MobiCom ’09), pp. 261–272, September 2009.
[6] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM ’00), pp. 775–784, March 2000.
[7] M. Y. Chen, T. Sohn, D. Chmelev et al., “Practical metropolitan-scale positioning for GSM phones,” Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp ’06), vol. 4206, pp. 225–242, September 2006.
[8] Y.-C. Cheng, Y. Chawathe, A. Lamara, and J. Krumm, “Accu-racy characterization for metropolitan-scale Wi-Fi localization,” in Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys ’05), pp. 233–245, June 2005.
[9] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, “SoundSense: scalable sound sensing for people-centric applications on mobile phones,” in Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys ’09), pp. 165–178, June 2009.
[10] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, “Indoor localization without infrastructure using the acoustic background spectrum,” in Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys ’11), pp. 155–168, July 2011.
[11] H. Liu, Y. Gan, J. Yang et al., “Push the limit of WiFi based localization for smartphones,” in Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (Mobicom ’12), pp. 305–316, 2012.
[12] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, “BeepBeep: a high accuracy acoustic ranging system using COTS mobile devices,” in Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys ’07), pp. 1–14, November 2007.
[13] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen, “IODetector: a fihcic service for indoor outdoor detection,” in Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys ’12).
[14] W.-H. Shin, B.-S. Lee, Y.-K. Lee, and J.-S. Lee, “Speech/non-speech classification using multiple features for robust endpoint detection,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’00), pp. 1399–1402, June 2000.
[15] G.-D. Wu and C.-T. Lin, “Word boundary detection with mel-scale frequency bank in noisy environment,” IEEE Transactions on Speech and Audio Processing, vol. 8, no. 5, pp. 541–553, 2000.
[16] B. Lee and M. Hasegawa-Johnson, “Minimum mean squared error a posteriiori estimation of high variance vehicular noise,” in Proceedings of the Biennial on DSP for In-Vehicle and Mobile Systems, Istanbul, Turkey, June 2007.
[17] H. Wang, S. Sen, A. Elgohary, M. Farid, and M. Youssef, “Romit roy choudhury. No need to war-drive: unsupervised indoor localization,” in Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys ’12), pp. 197–210.
[18] T. Mei, J. Xi, F. Yin et al., “Blind source separation based on time-domain optimization of a frequency-domain independence criterion,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 6, pp. 2075–2085, 2006.
[19] S. Araki, R. Mukai, Makinos et al., “The fundamental limitation of frequency domain blind source separation for convolution
mixtures of speech,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’01), vol. 5, pp. 2737–2740, 2003.

[20] S. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang, “Multi-channel blind deconvolution and equalization using the natural gradient,” in Proceedings of the 1st IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications (SPAWC ’97), pp. 101–104, April 1997.

[21] S. Nirjon, R. Dickerson, I. Stankovic, G. Shen, and X. Jiang, “sMFCC: exploiting sparseness in speech for fast acoustic feature extraction on mobile devices- a feasibility study,” in Proceedings of the 14th Workshop on Mobile Computing Systems and Applications (ACM HotMobile ’13), 2013.

[22] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, “Simple and practical algorithm for sparse fourier transform,” in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA ’12), pp. 1183–1194, January 2012.

[23] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, “Simple and practical algorithm for sparse fourier transform,” in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA ’12), pp. 1183–1194, January 2012.

[24] S. B. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. Acoustics, Speech and Signal Processing,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 4, pp. 357–366, 1980.

[25] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121–167, 1998.

[26] D. F. Specht, “Probabilistic neural networks for classification, mapping, or associative memory,” in IEEE International Conference on Neural Networks, vol. I, pp. 525–532, July 1998.

[27] D. F. Specht, “Enhancements to probabilistic neural networks,” in International Joint Conference on Neural Networks, vol. I, pp. 761–768, June 1992.

[28] J. P. Campbell, “Speaker recognition: a tutorial,” Proceedings of the IEEE, vol. 85, no. 9, pp. 1437–1462, 1997.

[29] I. Constandache, R. R. Choudhury, and I. Rhee, “Towards mobile phone localization without war-driving,” in Proceedings of the International Conference on Computer Communications (INFOCOM ’10), pp. 2321–2329, March 2010.

[30] S. Chu, S. Narayanan, and C.-C. J. Kuo, “Environmental sound recognition with time Frequency audio features,” IEEE Transactions on Audio, Speech and Language Processing, vol. 17, no. 6, pp. 1142–1158, 2009.