Indoor Localization without Infrastructure using the Acoustic Background Spectrum

Stephen P. Tarzia*   Peter A. Dinda*   Robert P. Dick†   Gokhan Memik*

*Northwestern University   †University of Michigan
EECS Department   EECS Department
Evanston, IL, USA   Ann Arbor, MI, USA
{spt175,pdinda,memik}@eecs.northwestern.edu   dickrp@eecs.umich.edu

ABSTRACT

We introduce a new technique for determining a mobile phone's indoor location even when Wi-Fi infrastructure is unavailable or sparse. Our technique is based on a new ambient sound fingerprint called the Acoustic Background Spectrum (ABS). An ABS serves well as a room fingerprint because it is compact, easily computed, robust to transient sounds, and surprisingly distinctive. As with other fingerprint-based localization techniques, location is determined by measuring the current fingerprint and then choosing the "closest" fingerprint from a database. An experiment involving 33 rooms yielded 69% correct fingerprint matches meaning that, in the majority of observations, the fingerprint was closer to a previous visit's fingerprint than to any fingerprints from the other 32 rooms. An implementation of ABS-localization called Batphone is publicly available for Apple iPhones. We used Batphone to show the benefit of using ABS-localization together with a commercial Wi-Fi-based localization method. In this second experiment, adding ABS improved room-level localization accuracy from 30% (Wi-Fi only) to 69% (Wi-Fi and ABS). While Wi-Fi-based localization has difficulty distinguishing nearby rooms, Batphone performs just as well with nearby rooms; it can distinguish pairs of adjacent rooms with 92% accuracy.

Categories and Subject Descriptors

H.3.4 [**Information storage and retrieval**]: Systems and software.

General Terms

Algorithms, experimentation, measurement, performance

Keywords

Localization, mobile systems, sound, fingerprinting

1. INTRODUCTION

The goal of our work is to allow a basic mobile device, such as a smartphone, to cheaply and quickly determine its location to the resolution of a single room. The mobile device may not have any specialized hardware, and there should be minimal, if any, dependence on computational or network infrastructure. More specifically, we seek a system that computes a

$$fingerprint \rightarrow roomlabel$$

mapping where *fingerprint* represents a measurement of the room the mobile device is currently in, and *roomlabel* represents a semantically-rich representation of the room. Fingerprint acquisition and mapping to room labels must be rapid. Fingerprints and room labels must be small so that it is possible to cache or prefetch them despite the device's limited memory. It must be possible to acquire a fingerprint and do the computation (using cached fingerprints) on the mobile device even if wireless communication is not presently possible. Above all, the fingerprint computation must be reliable in the presence of noise and environmental variations. In summary, a location fingerprint should be **DECENT**:

- **D**istinctive .... (see Section 5.1)
- r**E**sponsive .... (see Section 5.2)
- **C**ompact .... (see Section 5.2)
- **E**fficiently-computable .... (see Section 8.3)
- **N**oise-robust .... (see Section 5.3)
- **T**ime-invariant .... (see Section 5.4)

The Acoustic Background Spectrum (ABS) technique we describe here meets these requirements. The intuition behind ABS-based localization is that modern life is full of noises: computers whirr, lights buzz, and air conditioners blows. The field of architectural acoustics tells us that a room's geometry and furnishings strongly affect what is heard [18]. That is, the persistent acoustic drivers of a room, and the room's impulse response combine to form a distinct "sound" of the room. While it is obvious that a lecture hall sounds different than an office, we show that two otherwise similar-looking rooms are also very likely to have different persistent acoustic characteristics, even if the spaces may sound similar to a human listener.

Using ABS, which is implemented in our publicly available Batphone application[1], an Apple iPod is able to determine what room (and building, etc.) it is in using a compact $1.3\,kB$ representation of the room's acoustic spectrum as a fingerprint. The fingerprint is acquired within 10 seconds using the built-in microphone and by the eleventh second a list of room descriptions appears, which is ranked by likelihood of being the current room. Users can easily contribute new (*fingerprint, roomlabel*) pairs from the application.

The problem of indoor localization has been well-studied, but no ideal, all-purpose solution exists. Techniques based on observations of signals from radio beacons (such as Wi-Fi/cellular base stations or GPS satellites) have shown the most promise. However, these techniques are hampered by the fundamental inconsistency of indoor radio wave propagation caused by changes in the physical environment and interference from other electronic devices. Even the number of users connected to a base station affects the observed signal strength. Various studies have shown that Wi-Fi signal strengths vary significantly throughout the day and that these variations are not uniform (the base stations vary independently) [23, Figure 1] [13, Figure 7]. Signals from cellular base stations seem to be much more temporally-stable than Wi-Fi, however they are much less densely spaced, leading to localization difficulties.

Despite these observation uncertainties, researchers have built Wi-Fi-localization systems capable of 95% room-level accuracy in ideal, static environments [13, 32]. However, real usage environments generally make the problem harder in one of two ways: access point density may be low or occupancy variations may cause long-term signal variations. In either case, the reported Wi-Fi localization accuracy falls below 70% [13, Figures 6 & 8]. In addition, developing countries typically lack Wi-Fi infrastructure, but low-end mobile phones are widespread. Such devices have only a cellular radio and a microphone with which to sense the environment.

In contrast to radio-based techniques (Wi-Fi or cellular), our ABS technique works even if no radio signals are available if a sufficient collection of fingerprints is cached on the mobile device. Furthermore, the errors of ABS-based localization are different from those of radio-based localization, and thus the techniques can be combined for better performance than either one alone. In our experiments, the Batphone implementation of ABS improves the accuracy of the commercial Skyhook Wi-Fi-based localization service used by the iPod from 30% to 69%. The nature of the errors for radio-based localization is different from that for ABS-based localization: radio-localization errors tend to confuse nearby rooms, while ABS-based localization errors are generally *not* geospatially clustered. For example, using ABS alone allows us to distinguish adjacent rooms with 92% accuracy. The consequence is that user knowledge of general location is likely to increase the probability of selecting the correct room from a top-k list when ABS-based localization is used.

There is prior work on using acoustics for localization; a detailed comparison is given in Section 2. SurroundSense is the closest to our work [5]. It combines multiple sensors, including acoustic ambience measurement, visual ambience, and cellular-based localization to determine location. Sur-

roundSense's acoustic fingerprint is a compact histogram of a sound recording from the time domain, while ABS uses a compact *spectral* representation. Furthermore, our focus is on exclusively sound-based localization, and the ABS technique provides dramatically higher performance than the histogram technique when only the microphone is available. Another work, SoundSense [19], uses spectral features to classify sounds observed on a mobile device. Our work differs in that it targets localization rather than activity detection. Accordingly, we focus on background sounds while SoundSense focuses on transient sounds.

In summary, our contributions are as follows:

- We introduce the Acoustic Background Spectrum (ABS) fingerprint, a new acoustic ambience fingerprint that is compact, easily computed, robust to transient sounds, and able to distinguish remarkably similar rooms. ABS allows us to build the first accurate localization system based solely on microphone sensing.
- We introduce the Linear Combination Distance for combining localization fingerprints which, when applied to ABS and Wi-Fi, dramatically improves indoor localization accuracy over each individual method.
- We describe a trace-based simulation system that allows us to study and evaluate different acoustic localization techniques, and different parameters for an acoustic localization technique, using the same set of real acoustic traces (recordings). ABS was developed using this system and a collection of 264 acoustic traces, which we have made publicly available [1].
- We describe and make publicly available Batphone, an Apple iPhone implementation of ABS-based localization that demonstrates the technique's real-world performance and its low overhead. Batphone also includes the option to use Wi-Fi-based localization and combined ABS/Wi-Fi localization.

The rest of this paper is structured as follows. Section 2 describes related work. Section 3 describes the new Acoustic Background Spectrum (ABS) room fingerprinting technique. The ABS technique requires that we select values for certain parameters. We did so through trace collection and simulation. Section 4 describes our traces and simulator, while Section 5 describes our study to determine appropriate ABS parameters. This section also provides an initial evaluation of the performance of ABS and sensitivity to parameter variation. Section 6 describes the implementation of ABS within Batphone, including its integration with Wi-Fi-based localization. Section 8 evaluates the performance of Batphone. Section 9 concludes the paper.

2. RELATED WORK

Indoor localization has been well-studied in the mobile computing community. A user may be carrying a sensing device (such as a mobile phone) or may be moving within a field of fixed sensors. In either case, knowledge of the user's position can be useful for several applications. Generally, applications that depend on user location are called location-based services (LBS). Past applications have included environment interaction [3], reminders [11, 20], targeted advertising [5], tour guides [2, 8], navigation aids, and social networking (e.g., Foursquare and Gowalla).

Outdoor localization is well solved by GPS, but indoor localization remains a challenge in many cases. A wide variety

---

[1]Users of iPhones, iPads, and iPod Touch models with microphones can search for "Batphone" in Apple's app store. iOS version ≥ 4.0 is required.

of solutions have been proposed. The early work of Want et al. [29] and several subsequent works [30, 3, 25, 6, 27] require that sensors or beacons be installed throughout the environment. This requirement is onerous, but the resulting localization accuracy and reliability are generally excellent.

More recent work has focused on using existing cellular and/or Wi-Fi base stations as radio beacons for localization. In these systems [13, 32, 15], location is determined by observing the signal strength of Wi-Fi access points that are within range. Several efforts have been made to identify semantically-meaningful "places" rather than using simple room labels or coordinates. Kim et al. [17] give a recent example; they use the appearance and disappearance of radio base stations to indicate entrance into and departure from a place. Other place identification works cluster a time series of location coordinates (obtained by GPS or radio) to identify places, [16]. The Wi-Fi localization approach of Teller et al. eliminates the need to perform building surveys to build the location database [24]. There is even a commercial effort that uses wardriving to build a localization system[2].

Haeberlen et al. [13] reported perhaps the most accurate Wi-Fi-localization results to date (along with Youssef et al. [32]). Using a Gaussian distribution to statistically model signal strength observations, they reported 95% accuracy over a set of 510 rooms. However, indoor localization is not a solved problem. Looking more carefully at the results, we note that their environment had a dense Wi-Fi infrastructure: at least 5 base stations were in range at all times. Also, their test and training data were drawn from the same three-minute survey interval. When considering training and test data from different times of day [13, Figure 8], 70% accuracy was achieved, with most observed errors falling within 5.5 meters. We believe that acoustic localization can close this remaining accuracy gap.

Although Wi-Fi-localization is useful, many areas of interest are not served by Wi-Fi and other areas are served by only a single long-range base station. In the latter case, trilateration is impossible and fingerprinting gives only a very coarse location. There are also some privacy concerns with radio-based localization. Wi-Fi and cellular radios include digital identifiers (such as the MAC address for Wi-Fi) which allow the infrastructure operator to track users' positions (unless users take special privacy precautions). Acoustic techniques require no such identity broadcasting.

Beyond radio.

Recently, several localization systems have been built without the use of radio. For example, Constandache et al. use a mobile phone's accelerometer and compass for localization [9]. Woodman and Harle [31] used a foot-mounted accelerometer to locate a person within a previously-mapped building. The main disadvantage of these dead-reckoning approaches is that they can never determine the absolute location; only changes to position are captured. Therefore, small errors add up over time to create significant location drift. Acoustic techniques could complement these approaches.

Scott et al. present perhaps the first acoustic localization work; they place microphones at key locations and users snap their fingers to alert the system of their presence [27].

---

[2] http://www.skyhookwireless.com

symbol	meaning	optimal value	Batphone
R_s	sampling rate	96 kHz	44.1 kHz
n_{spec}	spectral resolution	2048 bins	1024 bins
n_{fp}	ABS size	299 bins	325 bins
t_{spec}	frame size	0.1 s	0.1 s
t_{samp}	sampling time	30 s	10 s
	frequency band	0–7 kHz	0–7 kHz
	window function	Hamming	rectangular

Figure 1: System parameters and the values chosen for the Batphone implementation. Section 5 shows the effect of altering these parameters.

We previously used an ultrasonic sonar sensing system to silently detect the presence of users at computers [28].

Azizyan and Choudhury (SurroundSense) use a combination of sensors (the microphone, camera, and accelerometer on a mobile phone) to distinguish between neighboring stores in shopping malls [5]. Their problem is made easier by the fact that neighboring stores typically offer different services and therefore a different ambience (including sound, lighting, and decor). Their acoustic processing is straightforward: it considers only audio sample amplitude distribution as a fingerprint. Hence, they distinguish between stores with different loudness characteristics. They report that audio alone did not give good localization accuracy; however, it did work well as the first stage in their multi-sensor localization method.

Other work has used different features to classify ambient sounds. Lu et al. present a framework called SoundSense for classifying sound events recorded on an iPhone and give two examples of its use [19]. They have an audio feature extraction pipeline similar to that used in speech recognition. SoundSense is ill-suited to localization because it ignores quiet frames [19, Section 4.1.2] and its audio features are tailored to transient sounds rather than background sounds. However, if a location can be characterized by one or more frequent, repeating, transient sounds then we expect that SoundSense could be used to aid localization. Our work focuses on persistent sounds and we deal primarily with quiet environments.

A common goal of related work in the audio signal processing community is to assign one of perhaps a few dozen labels to a recording. Sound classes typically correspond to places such as street, nature, construction, car, restaurant, office, or kitchen. Standard audio features such as the mel-frequency cepstral coefficients [10], zero-crossing rate, and spectral centroid are used. The system of Eronen et al. performed only slightly worse than human listeners [12]. Chu et al. achieved similar performance using the matching pursuit (MP) signal decomposition [7]. Our work differs from the above because its goal is not to assign semantic descriptors to recordings as people would, but rather to match previously-learned, specific location labels to recordings.

3. ACOUSTIC BACKGROUND SPECTRUM

Our localization scheme is based on the Acoustic Background Spectrum (ABS), a new ambient sound fingerprint. The design of the ABS draws from common acoustic signal processing techniques. Figure 2 gives an overview of how an ABS is calculated. All the parameters associated with the technique are listed in Figure 1. Our scheme works by first

calculating the ABS of the room (Sections 3.1 and 3.2) and then classifying the room by comparing this ABS with the existing, labeled ABS values in the database (Section 3.3). We now describe each step in detail.

3.1 Spectrogram representation

The first step of our localization technique is recording an audio sample of length t_{samp}. In the next several steps, we transform this signal into a time-frequency representation called a power spectrogram. This involves

1. dividing the recording into frames of length t_{spec},

2. multiplying each frame by a window function vector, which reduces the signal magnitudes near the frame boundary, and

3. computing the power spectrum of each frame, which involves

   (a) applying a fast Fourier transform (FFT) of resolution $2 \times (n_{\text{spec}} - 1)$,

   (b) throwing away the redundant second half of the result, leaving n_{spec} elements, and

   (c) multiplying the result elements by their complex conjugates, giving the power.

These are the standard signal processing steps for computing the power spectrogram of a discrete, real-valued signal [22, 26]. Its computation is efficient, with most steps requiring just a scan of the $R_s t_{\text{samp}}$ samples and the majority of time spent computing the $t_{\text{samp}}/t_{\text{spec}}$ FFTs, giving a runtime complexity in the class

$$\Theta \left(\frac{t_{\text{samp}}}{t_{\text{spec}}} (R_s t_{\text{samp}} + n_{\text{spec}} \log n_{\text{spec}}) \right).$$

After the spectrogram is calculated, we filter out the frequency band of interest by simply isolating the appropriate rows. The n_{spec} bins of the original spectrogram span the range up to the Nyquist frequency of $R_s/2$. If we are filtering out the 0–7 kHz band, as indicated in Figure 1, the number of bins retained is

$$n_{\text{fp}} = \text{round} \left(\frac{7\,\text{kHz}}{R_s/2} n_{\text{spec}} \right).$$

The spectrogram quantifies how the frequency content of the recording varies over time. Sounds that are persistent contribute equally to all of the time columns of the spectrogram. On the other hand, relatively short, transient sounds contribute only to a few time columns. We will next describe how to use this difference.

3.2 Transient noise rejection

After the spectrogram is computed, we apply a new method for extracting a noise-robust spectrogram summary vector. Intuitively, we would like to filter out transient sounds so that the fingerprint is time-invariant. In other words, we would like to extract the *background* sound levels. As shown in Figure 2, we accomplish this by choosing one of the smallest values observed for each frequency during the sampling window. However, the absolute minimum value is sensitive to outliers caused by noise and signal processing artifacts. Instead we choose a value near the minimum, the 5th-percentile value (p05). We expect that transient noises

lasting for less than 95% of the sampling window will be absent from the p05 value since such noises will be manifested as additive features within the upper 95% of the sorted values. Choosing the p05 value involves either sorting or using a linear-time selection algorithm (such as quickselect) on each of the spectrogram rows. Thus, the transient noise rejection step is less computationally complex than the spectrogram calculation step; using linear-time selection, it is in $\Theta(n_{\text{fp}} t_{\text{samp}}/t_{\text{spec}})$. We sort the rows individually since it is more likely that each row will be at least 5% quiet than the full spectrum being quiet 5% of the time. The results in Section 5.3 show that the 5th-percentile value gives better performance than the mean, minimum, or other percentile values.

Scaling and normalization.

The final step in computing the ABS is to compute the logarithm of the spectrogram summary vector. This puts the fingerprint in decibel (dB) units, which is the standard practice in audio signal processing. We validated the benefit of this step via simulation.

Afterward, the fingerprint may optionally be normalized. We do this by dividing the ABS vector by its median value. Ideally, normalization would not be necessary. However, during our experiments, various hardware gain levels were adjusted to prevent clipping in loud rooms; this made normalization necessary. We expect the median value of the spectrum to remain constant among rooms with different levels of transient noise if and only if those transients are narrow-band. Empirically, this seemed to be the case.

3.3 Classification

After the ABS room fingerprint is calculated, it can be compared to previously-observed fingerprints to determine the location. We solve this classification problem via supervised learning. We assume a database of room fingerprints is available where each fingerprint is labeled with a room identifier. The problem at hand is to label the currently-observed room fingerprint. To do this, we choose a distance metric for comparing room fingerprints. In particular, we use the vector Euclidean distance (the city block or Manhattan distance was also evaluated, as shown in Figure 8(b)). We then use the simple nearest-neighbor method of classification. This means that we choose the label of the previously-observed room fingerprint with smallest Euclidean distance from the current observation. This can be thought of, equivalently, as choosing the sample with smallest root mean square deviation.

Formally, assuming a set of training pairs (a database)

$$(\vec{fp}_i, roomlabel_i) \in T$$

and a new testing fingerprint $\vec{fp}_{\text{test}}$, the classifier chooses

$$roomlabel_{\text{best}} = \underset{(\vec{fp}_i, \text{roomlabel}_i) \in T}{\text{argmin}} \sqrt{\sum_{j=1}^{n_{\text{fp}}} (fp_i[j] - fp_{\text{test}}[j])^2}.$$

Nearest-neighbor classification is required each time a location estimate is desired; the Batphone implementation does this operation every two seconds. Because the fingerprint space has high dimensionality ($n_{\text{fp}} \cong 300$), finding the nearest neighbor can become a bottleneck: we must scan through each fingerprint in the database and compute its distance to the observed fingerprint. This most straightforward
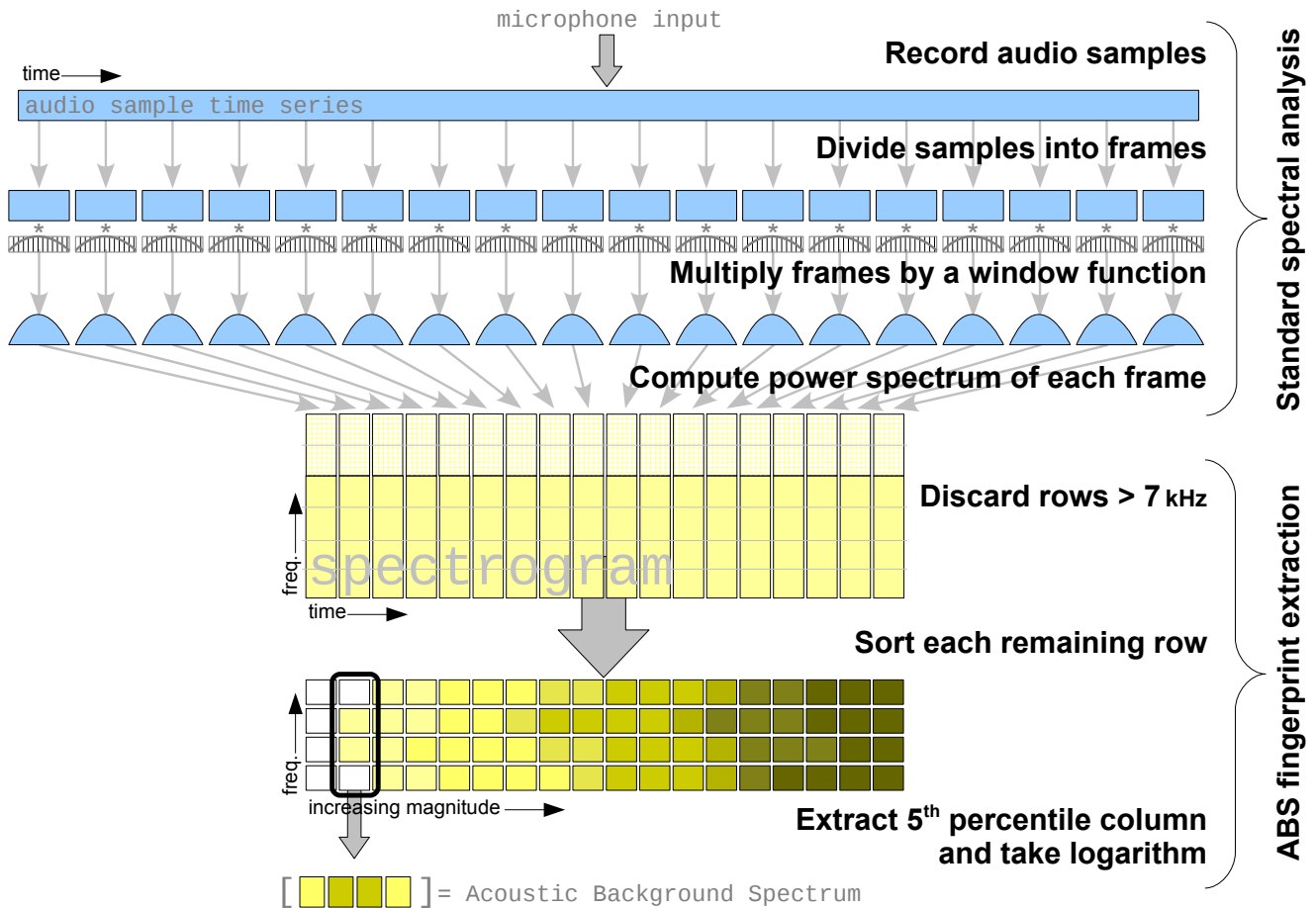
microphone input

time ➔

audio sample time series

Record audio samples

Divide samples into frames

* * * * * * * * * * * * * * * * *

Multiply frames by a window function

Compute power spectrum of each frame

Standard spectral analysis

freq.

spectrogram

time ➔

Discard rows > 7 kHz

Sort each remaining row

freq.

increasing magnitude ➔

Extract 5ᵗʰ percentile column and take logarithm

ABS fingerprint extraction

[▢▢▢▢] = Acoustic Background Spectrum

Figure 2: Acoustic Background Spectrum fingerprint extraction.

approach is clearly not scalable as the database grows to millions of fingerprints. However, nearest-neighbor query latency was imperceptible on Batphone when using a database of a few hundred fingerprints. In future work, we anticipate using a combination of geospatial database queries on a server backend and client-side caching and prefetching to deal with database scaling. We could also use coarse-grained information, such as nearby base station identifiers, to categorize signatures to reduce the search space.

It is important to point out that the problem of finding nearest neighbors is common to most fingerprint-based localization approaches. This problem is widely studied and ABS comparison can benefit from advances in the field. As the dimensionality of the fingerprint is reduced, the problem becomes easier. As we describe in Section 5.2, ABS fingerprints continue to provide good accuracy even when n_{fp} is reduced to 19.

4. TRACE COLLECTION AND SIMULATION

At the onset of this project we speculated that acoustics could be used to identify rooms, but we did not know which acoustic features would be most useful. We used an acoustic-trace-collection approach to gather the data necessary to evaluate acoustic localization. We visited 33 rooms in the Technological Institute and Ford Design Center buildings on

the campus of Northwestern University while carrying the recorder shown in Figure 3(a), mounted on a microphone stand. The experimenter moved the recorder to four positions in the rooms, each time capturing a 30 second WAV file (24-bit 96 kHz). Stereo recordings were captured, but only the left-channel data was used. Trace collection spanned several weeks, and each room was visited on two different days, giving a total of eight observations per room. We recorded various types of rooms, as shown in Figure 4. In our experiments, we did not observe any significant correlation between localization accuracy and room type or size. We included all the rooms we encountered in those two buildings that were unlocked and which we could access without disrupting class and work activities. Hence, the majority of our samples are from empty, quiet classrooms (Section 5.3 deals with noisy rooms). We did not exclude any samples after seeing the results, or for any other reason. Our traces are publicly available on the project web site [1].

Collecting these traces allowed us to simulate the performance of acoustic localization while varying several system parameters. In particular, capturing raw, unprocessed recordings gave us the flexibility to study the effects of applying various processing techniques to the same data. The basic simulation approach was to test system performance when observing each of the recordings in turn, assuming

(a) Zoom H4n   (b) Apple iPod Touch

Figure 3: Experimental platforms. (a) The Zoom H4n handheld recorder was used to collect traces used in the simulations. Its price is ∼$300. (b) The Apple iPod Touch (4th generation) running our Batphone localization software was used for the end-to-end experiment.
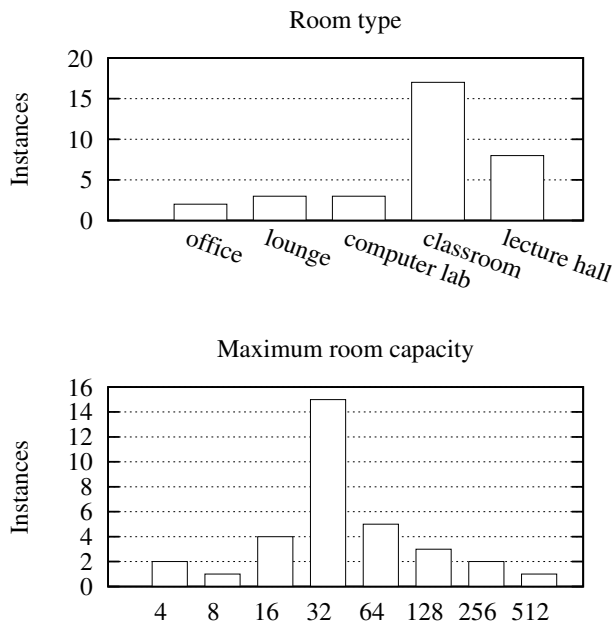

Figure 4: Histograms showing the distributions of room type and room size for the traces.

that all other recordings had been previously observed; this is called leave-one-out classification. Specifically, each of the 264 samples (33 rooms × 2 visits × 4 positions) was compared to 132 of the samples, which served as the training set. Half of the samples for each room were included in the training set, and the choice of test samples varied according to the room being tested. For the test sample's room, all four samples from the *other* day's visit were included in the training set. For the 32 other rooms, the training set included two samples from each of the two visits. Thus, the training set was unbiased toward any room and it excluded any samples from the same visit as the test sample. As explained in Section 5.4, including training data from the same visit as the test data would have made the resulting classification accuracy unrealistically high. Given the training data set described above, the location label for each test sample was assigned from the "closest" training sample; i.e., we used nearest-neighbor classification.

Using this trace collection and simulation scheme, we were able to choose the best parameters for Batphone and to directly compare its performance to that of prior acoustic approaches. These results follow.

5. SIMULATION RESULTS

We now describe an evaluation of the ABS technique using the *DECENT* criteria explained in the Introduction. The evaluation uses trace-based simulation and focuses not only on the overall performance of ABS, but also on its sensitivity to its parameters. Our selection of parameter values for the Batphone implementation is based on this study.

5.1 Distinctiveness

In order to support localization, a fingerprint should be unique to a single location. In other words, a one-to-one mapping should exist from fingerprints to locations. The role of infrastructure in localization systems is to provide such a mapping. For example, Wi-Fi localization systems rely on the uniqueness of base station MAC addresses. In the absence of uniquely-labeled infrastructure elements, fingerprints are constrained to a finite space and thus overlap of fingerprints is inevitable when the number of locations is sufficiently large. One of the main purposes of our experiments was to evaluate fingerprint distinctiveness, which we measure by the localization accuracy.

Figure 6 shows the localization accuracy as a function of the number of rooms being considered, which we call the problem size. For the full set of 33 rooms, localization accuracy is 69%. Pairs of rooms were distinguished with 92% accuracy. Both of these fractions are far higher than the 3% and 50% random chance probabilities. ABS fingerprints also compare favorably to our own implementation of Surround-Sense's acoustic fingerprint algorithm [5]. Note that these SurroundSense results include only their acoustic technique, not their full sensor fusion approach (i.e., we do not include the camera and radio).

Figure 5 shows the entire set of simulation data for the optimal parameter values; it plots all 264 ABS fingerprints, grouped by room and further grouped into two visits per room. This plot visually confirms the similarity of ABS fingerprints within rooms and distinctiveness between rooms. Figure 7 shows the classification confusion matrix for this simulation. We can see that prediction errors, indicated by off-diagonal elements, tend to be distributed across the

Figure 5: Acoustic Background Spectra (ABS) from 33 rooms, using the optimal parameter values. The vertical axis is the ABS value for each frequency bin; these values correspond to the log-scaled power spectrum. Fingerprints from two visits are vertically offset from each other to aid visual comparison. Each visit includes four overlaid fingerprints from different locations within the room.
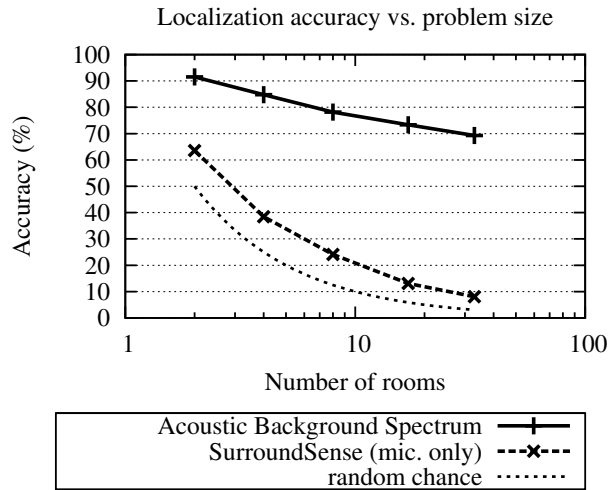
Localization accuracy vs. problem size

Figure 6: Accuracy as a function of the number of rooms being distinguished. 1,000 random subsets of rooms were used to calculate accuracy for problem sizes with fewer than 33 rooms.
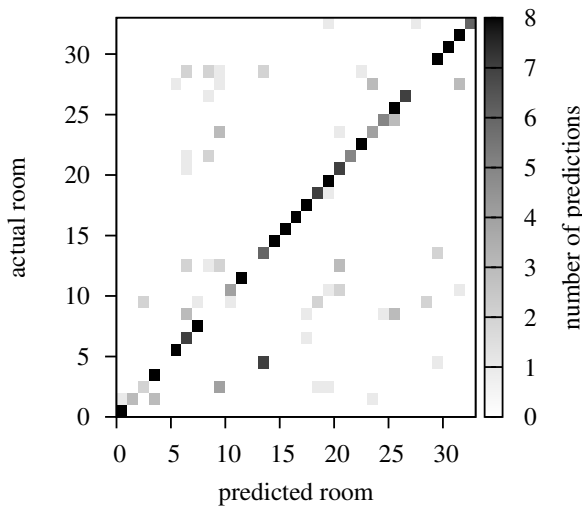
Figure 7: Confusion matrix for the 33 room simulation, using the optimal set of parameters. This shows all of the localization decisions made in the simulation. 69% of these location predictions were correct, and thus fall along the diagonal.

rooms rather clustered at any particularly-favored predictions.
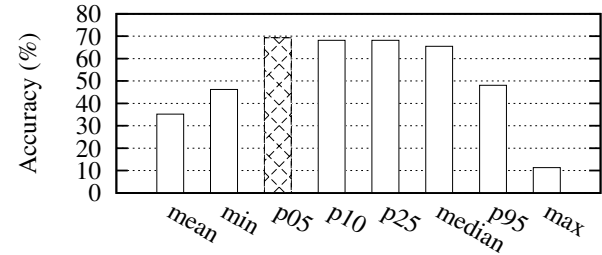
5.2 Parameter study

Figure 8 shows how localization accuracy is affected by changes in the ABS fingerprint parameters. Each of these figures represents the change in localization accuracy simulated as one parameter was changed while the other parameters were kept at their optimal values. We kept the problem size constant, using the full set of 264 samples from 33 rooms.

(a)

Frequency band	Accuracy
full (0–48 kHz)	59.8%
audible (0–20 kHz)	64.8%
low (0–7 kHz)*	69.3%
very low (0–1 kHz)	61.0%
(0–600 Hz)	51.5%
(0–400 Hz)	44.3%
(0–300 Hz)	40.9%
(0–200 Hz)	30.7%
(0–100 Hz)	15.5%
high (7–20 kHz)	28.4%
ultrasonic (20–48 kHz)	25.0%

(b)

Distance metric	Accuracy
Euclidean*	69.3%
city block	66.7%

(c)

Spectrogram window	Accuracy
rectangular	65.2%
Hamming*	69.3%
Hann	68.2%
Blackman	67.4%

(d) Fingerprint type

(e) Fingerprint size (n_{fp}), in floats
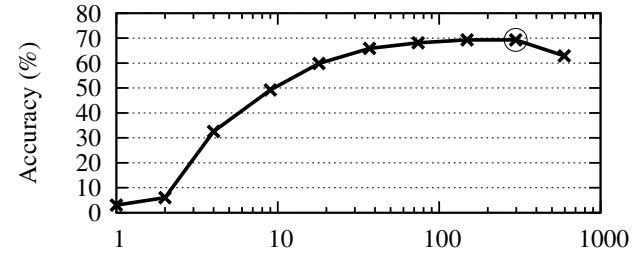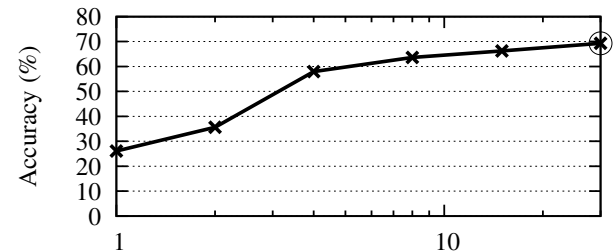
(f) Sample time (t_{samp}), in seconds

Figure 8: Parameter study showing the best localization accuracy achieved for 33 quiet rooms using a variety of signal processing parameters. Optimal values are highlighted.

Responsiveness.

The first parameter we consider is the sample time (t_{samp}), shown in Figure 8(f). Intuitively, we expect a longer recording to be more noise-robust and thus to provide better localization accuracy. However, a short sample time is desired for system responsiveness, especially if the user is mobile. Although the best results were obtained using the full 30 second sample time, reducing the sample time to 4 seconds reduced the localization accuracy by only 11 percentage points to 58%. An intermediate value of 8 seconds yielded 64% accuracy. Our sample time is shorter than the one minute sample time recommended in Wi-Fi localization work [13].

Compactness.

To reduce memory requirements, a fingerprint should be compact. The ABS fingerprint size can be adjusted by varying the frequency resolution (n_{fp}). However, reducing the fingerprint size generally reduces distinctiveness. Figure 8(e) shows the relationship between localization accuracy and the ABS size. Each frequency bin is represented by a floating point value. The optimal size of 299 bins requires 1,196 bytes of storage (assuming single-precision floats). However, if memory is more constrained, ABS fingerprints with only 19 bins (76 bytes) still provide 60% accuracy. Note that accuracy drops when moving to very high resolutions, presumably due to over-fitting.

Frequency band, metric, and window type.

Figure 8(a) shows the effect of limiting the ABS to various frequency ranges. We found that the 0–7 kHz frequency band gave the best results for quiet room samples (in Section 5.3 we consider the noisy case). By the Nyquist sampling theorem, this means that any audio sampling rate above 14 kHz would be sufficient to achieve the best results. These requirements are very modest; our iPod supports 44.1 kHz sampling. A sampling rate of at least 8 kHz is required to capture speech. At this low sampling rate, 61% localization accuracy can still be achieved. These results suggest that highly distinctive ABS fingerprints can be captured with even the most basic audio hardware.

The distance metric chosen for comparing fingerprints also affects the results, as shown in Figure 8(b). The city block (also called Manhattan) distance requires less than half the arithmetic operations of Euclidean distance, but gives results a few percentage points less accurate. Similarly, as shown in Figure 8(c), using a rectangular window in the spectrogram generation eliminates the vector multiplication cost in exchange for losing a few percentage points of accuracy.

5.3 Noise-robustness

Any ambience sensing method for fingerprinting should be robust to temporary changes in the environment, which we call noise. In the design of the ABS, the 5th-percentile (p05) fingerprint extraction step was intended to accomplish this, as described in Section 3.2. Figure 8(d) shows how the p05 transient noise rejection method affects localization accuracy. The most important comparison in that bar chart is to the mean. The mean method refers to collapsing the spectrogram into a single column by averaging all of the values across time (the rows) and it is one of the standard methods for computing the power spectrum (Welch's method). The results show that this basic spectral fingerprint produced by

the mean achieves 33 percentage points less accuracy than the p05 fingerprint, presumably because it produces fingerprints containing transient noise. Figure 8(d) also shows accuracy achieved using other percentile values, from the minimum (0th percentile), through the median (50th), to the maximum (100th). The 5th-percentile gave the best results, but other low percentile values have similar benefits compared to the mean.

To test the effect of occupancy noise on the ABS, we made recordings in a lecture hall before, during, and after lectures. This particular lecture hall (Tech LR5) has a maximum seating capacity of 88. Two lectures were recorded: "Economic History of the U.S. to 1865" with 44 attendees and "Real-Time 3D Game Engine Design" with 29 attendees. We defined three occupancy noise states that were observed in the recordings.

- *Quiet* times, prior to lecture, when a few students might be present in the room, but no conversation occurred,
- *Conversation* times, such as during the lecture, when a single voice can be heard, and
- *Chatter* times, directly before and after lecture, when many students were present, talking and moving noisily.

We divided the 162.5 minutes of recordings into 30 second windows and computed the ABS fingerprint of each window. We ran the classification algorithm described in Section 3.3 using the same 33 rooms of training data. Thus, the training set included only quiet samples while the test set included quiet, chatter, and conversation samples. This experiment tested the ability of the system to deal with the two types of occupancy noise. We did not expect the occupancy noise to be uniformly distributed across frequency. Therefore, we tested the localization accuracy using a variety of frequency bands in the ABS fingerprint, not just the 0–7 kHz band that gave the best results for the quiet parameter study, as shown in Figure 8(a). This allowed us to determine which frequency band is least sensitive to occupancy noise.

Figure 9(a) shows the accuracy achieved with the noisy test samples. Accuracy during the chatter noise state was very poor, never significantly passing the random change accuracy of 3.0%. However, the conversation state gave more varied results. Accuracy was poor when using the 0–7 kHz band in the conversation state; p05 transient noise rejection was insufficient. However, 63.4% accuracy was achieved when restricting the ABS to 0–300 Hz. We assume that the success at the 0–300 Hz band is due to its exclusion of the speech band, which is approximately 300–3000 Hz [21]. Unfortunately, as shown in Figure 8(a), overall quiet-state localization accuracy dropped from 69% to 41% when omitting the speech band, so there is a trade-off here between quiet-state accuracy and speech-noise rejection. These results suggest that automatically detecting the noise state and switching the ABS band accordingly has the potential to improve the overall success of our approach.

Figure 8(d) showed that p05 transient noise rejection improves over the mean method in the quiet state. The improvement is even more drastic in the conversation state. In that case, using the mean method with the 0–300 Hz band gave 0% accuracy compared to 63.4% for p05. In summary, a combination of transient noise rejection and band selectivity is needed to deal with speech noise.

| | Occupancy Noise State | | |
Frequency band	Quiet	Conversation	Chatter
(a)	Tech LR5 lecture hall		
full (0–48 kHz)	56.8%	0.0%	0.0%
audible (0–20 kHz)	78.4%	0.0%	0.0%
low (0–7 kHz)	**89.2%**	2.5%	0.0%
very low (0–1 kHz)	45.9%	0.0%	0.0%
(0–600 Hz)	37.8%	7.4%	0.0%
(0–400 Hz)	73.0%	11.4%	0.0%
(0–300 Hz)	75.7%	**63.4%**	0.0%
(0–200 Hz)	21.6%	27.2%	1.2%
(0–100 Hz)	21.6%	26.7%	4.7%
high (7–20 kHz)	21.6	0.0%	0.0%
ultrasonic (20–48 kHz)	32.4%	4.5%	3.5%
(b)	Ford 3.317 lounge		
low (0–7 kHz)	**98.2%**	47.2%	—
(0–300 Hz)	87.7%	**79.2%**	—

Figure 9: Localization accuracy in two rooms as a function of ABS frequency band and occupancy noise state.

We also tested noisy-state accuracy in a smaller room. In particular, we chose a kitchen lounge (Ford 3.317) which is open to a hallway. During an hour-long experiment, we invited graduate students to the lounge to eat free donuts and socialize. A few dozen people came and left. At any time, between 0 and 7 people were present in the room and conversing while recordings were made and the occupancy and noise level was monitored. Figure 9(b) shows that, again, switching to the 0–300 Hz band improved accuracy. Conversation-state accuracy was higher in this room than in the lecture hall; this could be due to the presence of longer pauses in speech while chewing donuts or due to the size of the room.

5.4 Time-invariance

In order for a room fingerprint to be useful for localization, it must remain relatively constant over time. Our traces spanned several weeks and our simulations excluded same-visit training data. Thus, our results already have factored in the effect that time has on acoustic fingerprints. The high accuracy numbers we report support the claim that ABS fingerprints are stable enough to be used for localization.

Nonetheless, ABS fingerprints did vary somewhat over time. Fingerprint pairs observed on different visits to the same room had signal distance, on average, 24% larger than pairs observed during the same visit. In other words, while fingerprints observed in the same room on different days are similar enough to support accurate localization, fingerprints observed in quick succession are even more similar. In fact, if our simulations had used same-visit training data they would have reported 95.8% localization accuracy. However, that would have been unrealistic since real localization systems must operate using only past training data. The stability characteristics of ABS fingerprints are similar to those of Wi-Fi fingerprints, as reported by Haeberlen et al. [13]. In that work, localization accuracy dropped from 95% to 70% when same-visit training data was excluded.

| HVAC state | | | |
Training	Test	Training type	Accuracy
on	on	different visit	71.7%
on/off	off/on	different visit	17.9%
on	on	same visit	98.9%
off	off	same visit	88.0%

Figure 10: Localization accuracy for 23 rooms under different combination of HVAC states.

HVAC effects.

Despite the empirical stability of ABS fingerprints, we did identify a scenario in which the ambient sound changed drastically with catastrophic results. Over one weekend, the climate control (HVAC) system in the Technological Institute was shut off for maintenance. We used this rare opportunity to measure the effect that these vents have on the ABS. The results are summarized in Figure 10. When using training data from the a different HVAC on/off state, localization accuracy dropped from 71.7% to 17.9% for the 23 rooms accessed over that maintenance weekend. So, changes in HVAC state have a large impact on ABS fingerprints. Yet, as we have already observed, ABS fingerprints remained stable during the main trace collection period. We conclude that HVAC did not significantly change state during trace collection. Of course, some buildings may experience HVAC state changes, and seasonal changes are also possible. However, since such combinations are not numerous, fingerprints representing each of the possible HVAC states can be collected for each room.

At first, we thought that we might identify HVAC sounds as the main source of distinctiveness in ABS fingerprints. However, the samples from the HVAC-off state still exhibited enough distinctiveness to classify with 88.0% accuracy compared to 98.9% accuracy for the HVAC-on state. Note that, because we had only one day to survey the HVAC-off state, we used same-visit training data in these results. These numbers would be lower if testing and training were done on different days, so they should not be directly compared with our other results. Still, they show that the HVAC sounds improve ABS-localization but are not essential.

6. BATPHONE IMPLEMENTATION

After completing the trace-based simulation we implemented ABS fingerprinting on Apple's iOS mobile platform. This application, named Batphone, is publicly available on Apple's app store and its GUI is shown in Figure 3(b). Batphone allows researchers to perform ABS localization in real-time as well as to capture and export fingerprints for off-line analysis. For details see the project website [1]. Our purpose was to evaluate the limitations imposed by a real device's audio hardware and to ensure that the system could operate in real-time.

Batphone is named after the small flying mammals that navigate and locate prey by emitting chirps and then listening to resulting echoes. Batphone, on the other hand, does not emit any sound (nor do our phones fly or eat insects); it follows the same fingerprint extraction steps described in Section 3, with some adaptations motivated by implementation considerations.

Hardware limitations.

Compared to the hardware used in trace collection, our deployed mobile platform has inferior recording capabilities. Its microphone is small, inexpensive, and crowded by other electronics. Also, the audio circuitry is designed for recording speech, and thus has a limited frequency ranges. In particular, audio sampling is 16-bit at 44.1 kHz. The system parameter values chosen were based on the results previously presented and are listed in Figure 1.

In a real system, location must be calculated in real-time and on a battery-powered device with limited computation power. We made a few adjustments to the ABS extraction algorithm to reduce its computational load while only slightly reducing its accuracy. In particular, we use a rectangular window function rather than a Hamming window and omit the normalization step. Section 8.3 shows our overhead measurements.

On-line training.

Another complication in the real system is online training and bootstrapping. Recent localization work has explored strategies for building fingerprint databases on-line rather than relying on explicit surveys [24]. Whenever a fingerprint is observed, the system either matches it with one already observed in the database or adds it to the database as an example of a new room. On-line systems also can automatically adapt to changes in the environment. We do not address the challenges of on-line database growth in this work. Instead, we use the same surveying and simulated training/testing approach that we used in the previous experiment.

Sliding-window 5th-percentile.

To improve responsiveness in the Batphone implementation, we implemented the ABS calculation steps of Figure 2 in a streaming fashion. The spectrogram rows are stored in sliding window buffers. After each recording frame arrives, its power spectrum is computed, each frequency element is added to the corresponding spectrogram sliding window, and the oldest element of each window is discarded. At this point the fingerprint is recomputed, based on the new 5th-percentile values from each sliding window. Thus, the fingerprint is updated every $t_{\text{spec}} = 0.1$ seconds rather than every $t_{\text{samp}} = 10$ seconds. Of course, if the user moves to a new location, it will be 10 seconds before all data from the old location are purged from the sliding window. However, this is much better than the 20 second worst-case purge time when not using a sliding window.

The most complex part of the streaming implementation is tracking the 5th-percentile value of the sliding windows. This is known as a rank-order filter [4]. In a naïve implementation, a selection or sorting algorithm would be run on the spectrogram rows after each new frame arrives. Instead, we use a much more efficient heap-based data structure to track the p05 value. Values below the current p05 value are stored in a max-heap while values above the current p05 value are stored in a min-heap. Additionally, a mapping between heap indices and a FIFO queue is maintained to allow the oldest data value to be quickly located in the appropriate heap. Details are omitted, but we are simply adapting standard algorithms for calculating the running median to use the 5th instead of 50th percentile [4, 14]. The runtime

complexity of a frame insert operation is dominated by n_{fp} heap inserts and is in $\Theta\left(n_{\text{fp}} \log\left(t_{\text{samp}}/t_{\text{spec}}\right)\right)$.

7. LINEAR COMBINATION DISTANCE

As seen previously in Figure 6, the performance of acoustic localization declines as the problem size (the number of rooms to choose from) increases. Our study considered scaling empirically up to 33 rooms.[3] Regardless of the asymptotic scaling of ABS, however, ABS fingerprints could be used to great effect on large-scale problems when combined with another fingerprint such as those from Wi-Fi or cellular radios.

One option is to use a multi-step localization scheme [5]. First, radio localization determines the approximate location, then acoustic localization can effectively determine the final location since it simply has to choose from among a small location neighborhood. We found this approach to be both intuitive and relatively effective. However, we propose a new approach that gives better accuracy by considering the two fingerprint spaces simultaneously; we propose a *linear combination* of each type of fingerprint distances.

The linear combination depends on some measurements of the environment characteristics. Each of n fingerprint types gives a distance d_i, which we expect to fall within some match range (min_i, max_i) when comparing two fingerprints observed in the same room. Furthermore, we assign each fingerprint type a weighting constant w_i indicating the relative importance or reliability of that fingerprint type. Combining these constants in the natural way yields the linear combination distance formula:

$$d_{\text{combined}}(d_i \ldots d_n) = \sum_{i=1}^{n} w_i \frac{d_i - min_i}{max_i - min_i}.$$

In our experiments, min_i and max_i are chosen simply as the minimum and maximum observed same-room distances. In a real system, some sort of outlier elimination would be needed (perhaps by using the 5th and 95th-percentile same-room distances). The weighting constants w_i were chosen experimentally; since only two fingerprint types were considered, this meant just setting $w_1 = 1$ while varying w_2 and choosing the value that resulted in highest localization accuracy.

8. BATPHONE RESULTS

Our second experiment was carried out with our Batphone application running on an Apple iPod Touch mobile device (Figure 3(b)). The procedure was identical to that described in Section 4, except that instead of capturing sound recordings the Batphone app was used to simultaneously capture two compact fingerprints: an ABS fingerprint and a Wi-Fi location coordinate, described below. It involved a set of 43 rooms mostly overlapping with the 33 rooms used in the previous experiments. Also, only two room positions were captured rather than four, meaning that the system had less training data. The two purposes of this experiment were to show the feasibility and performance of a real implementation on typical smartphone hardware and to compare Batphone's localization results to those of the device's

---

[3]One goal of making the Batphone implementation publicly available is to allow us to expand our scaling study through volunteers.
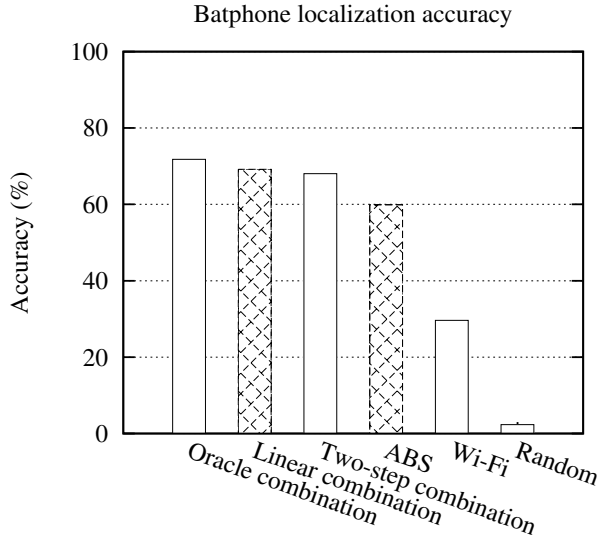
Figure 11: Localization accuracy for a set of 43 rooms using a variety of fingerprint types.

Figure 12: Comparison of localization performance for a variety of fingerprint types. Note that Batphone's ABS fingerprint is superior to Wi-Fi's at fine granularity but inferior at coarse granularity. The linear combination approach combines the best aspects of both localization methods.

Wi-Fi-localization service, which represents the commercial state of the art. We also evaluate the linear combination localization approach.

On our iOS 4 mobile device, Apple's Core Location service maps Wi-Fi RSSI scans to GPS coordinates. On older versions of iOS, Wi-Fi localization was provided by the Sky-hook service. If a cellular or GPS signal is also available, then that information is used as well (however, our test iPod had no such radios). We use the returned (latitude, longitude) coordinate as the Wi-Fi fingerprint, since we do not have access to Apple's internal Wi-Fi fingerprint representation. We compute distances between Wi-Fi fingerprints as simply the traveling distance between coordinates (known as the great circle distance). Core Location provides a dynamic accuracy estimate, which is typically in the tens of meters when using Wi-Fi. This accuracy is far inferior to that achieved by the latest research methods; it is clearly not intended for room-level indoor localization. However, Core Location serves as a useful comparison point because it is readily available and well-trained; its database has drawn data from millions of GPS-equipped iPhones. We treat it as a proxy for any of various research Wi-Fi-localization methods operating in sub-optimal conditions, e.g., due to sparse infrastructure, incomplete surveying, or radio interference. We expect such conditions to be quite common. The absolute performance we report for Wi-Fi is far less important than the fact that combining it with ABS yields a very accurate and scalable localization system.

8.1 Accuracy

Figure 11 shows our Batphone localization results. Wi-Fi performs poorly at room-level localization, as expected. Our Batphone ABS-localization implementation running on real mobile device hardware performs on par with the simulation results summarized in Figure 6. It appears that moving from expensive music-recording equipment to a mobile device's built-in microphone had no significant impact on per-
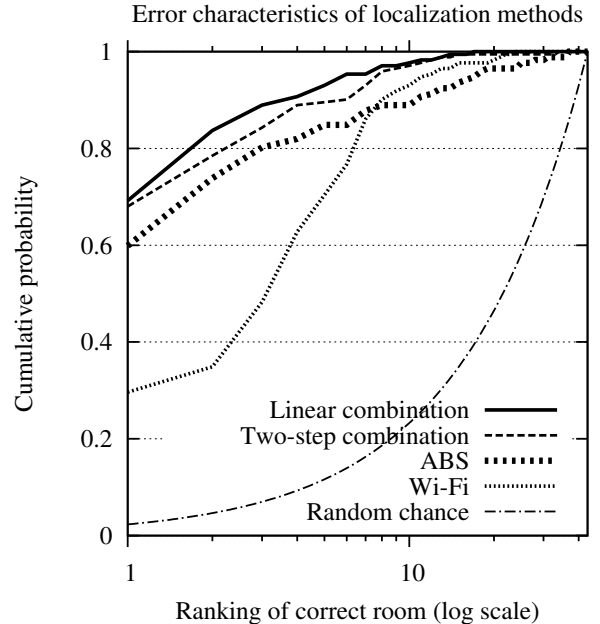
formance. Also, the linear combination distance performed much better than either ABS or Wi-Fi alone. We also compared to a two-step localization approach, which first eliminated all rooms with Wi-Fi distance greater than 35 meters (this value was optimized empirically) and then chose the room with closest ABS distance. This combination also performed well, but had accuracy 1.5 percentage points inferior to the linear combination. Note that empirically-optimized weighting constants used for the linear combination were $w_{\text{abs}} = 3$ and $w_{\text{wifi}} = 1$.

The oracle combination bar in Figure 11 gives a sense of the upper-bound performance of any fingerprint combination scheme. In this hypothetical approach, the algorithm first generates a location estimate using one fingerprint type. Next, an oracle tells whether that location choice was correct, and, if not, the second fingerprint type is used to make a new choice.

The error characteristics were also quite different for the localization methods. Figure 12 shows the distribution of rankings for the correct room. These rankings are determined by the sorting the distances to the closest fingerprint from each room. If the correct room is ranked number 1, then that room is selected as the current location and the localization method is said to be accurate; if the correct room is ranked number 2 then localization was "almost correct", etc. Figure 12 shows that acoustic and Wi-Fi localization methods have different strengths. Looking at the left-side of the plot, we see that ABS is much more often strictly-accurate (due, in part, to the relatively small problem size of 43 rooms). At the right-side of the plot, performance of Wi-Fi surpasses ABS; Wi-Fi makes fewer "really bad" errors.

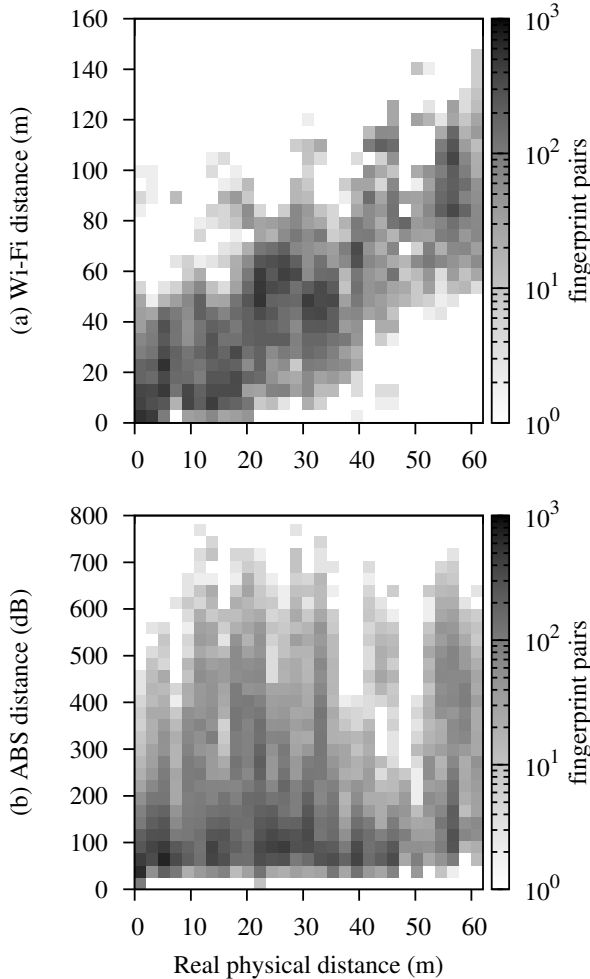2D histograms of physical and fingerprint distances

Figure 13: Relationship between fingerprint and physical distances.

The combined localization approaches work far better than either ABS or Wi-Fi, with the linear combination showing a slight advantage over the two-step combination.

8.2 Adjacent room discrimination

Unlike radio, sound waves do not penetrate walls well. Therefore, we expect that among adjacent rooms, ABS fingerprints will vary much more significantly than radio fingerprints. To test this hypothesis, we measured the physical distances between the rooms in our study using pixel locations on a floorplan image. Figure 13 shows the relationship between fingerprint distances and physical distance. These results are similar to those reported by Park et al. [24, Figure 2]. Figure 13(a) shows, as expected, that Wi-Fi distance tends to increase with physical distance. In particular, no distant room pairs had low Wi-Fi distance; i.e., Wi-Fi has high coarse-grained accuracy. In contrast, Figure 13(b) shows that ABS distances of rooms are generally uncorrelated with their physical distances. In particular, nearby rooms have a high degree of diversity (with pairwise distances relatively evenly spread across the ABS distance

range). We exploited these complementary features in the linear combination of these two fingerprints, described previously in Section 7 and illustrated in the results of Figures 11 and 12.

Assuming no correlation between ABS and physical distances (as suggested above), Figure 6 tells us that pairs of adjacent rooms should be distinguished by ABS fingerprints with around 92% accuracy.

8.3 Overhead

To test the overhead associated with ABS fingerprint extraction, we measured iPod battery lifetime while continuously running the Batphone app (with the screen on its dimmest setting). The device ran for 7.9 hours on a single charge, computing about 260,000 fingerprints and making 13,000 location estimates from a database of 190 stored fingerprints (while also rendering an animated GUI showing the results). Battery lifetime was 8.75 hours when running a stripped-down version of the app that simply captured ABS fingerprints without classifying them. For comparison, Wi-Fi localization gave the same 8.75 hour battery life and the manufacturer claims up to 7 hours of battery life while playing videos. While continuously determining location and rendering the results at a rate of once per two seconds, Batphone used 11.5% of the CPU and 16.7 MB of RAM (of 256 MB total). These energy and performance overheads certainly seem tolerable.

9. CONCLUSIONS

We have shown that accurate indoor localization can be achieved without reliance on a dense, static radio infrastructure. To accomplish this, we introduced a new acoustic ambience fingerprint, the Acoustic Background Spectrum, which is highly distinctive, robust, and easy to compute. We further introduced a method of combining location fingerprints that allows for highly accurate indoor localization by combining two (or more) less accurate fingerprints with different error characteristics.

Future work.

We are presently building a fingerprint sharing and matching infrastructure that can scale to many users. Also, we have not fully explored using the linear combination distance with other fingerprint types. It is likely that more accurate localization can be accomplished by adding accelerometer and camera data.

Portions of the noise rejection problem remain unsolved. The 5th-percentile filtering technique does reject some noise and we have a strategy for dealing with single-speaker noise (by switching to the 0–300 Hz band). However, we do not have a policy for controlling this frequency band switch. Furthermore, we were not able to cope with the noise introduced by tens of speaking occupants (the chatter state). We will explore whether adding training samples that include noisy states addresses the latter problem.

10. REFERENCES

[1] Batphone software project web page.
http://stevetarzia.com/batphone.

[2] Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. Cyberguide: A mobile context-aware tour guide. *Wireless Networks 8* (1997), 421–433.

[3] Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A., and Hopper, A. Implementing a sentient computing system. *IEEE Computer 34*, 8 (2001), 50–56.

[4] Astola, J., and Campbell, T. On computation of the running median. *IEEE Trans. on Acoustics, Speech and Signal Processing 37*, 4 (Apr. 1989), 572–574.

[5] Azizyan, M., and Choudhury, R. R. SurroundSense: mobile phone localization using ambient sound and light. *ACM SIGMOBILE Mobile Computing Communications Review 13*, 1 (2009), 69–72.

[6] Borriello, G., Liu, A., Offer, T., Palistrant, C., and Sharp, R. WALRUS: wireless acoustic location with room-level resolution using ultrasound. In *Proc. Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)* (2005), pp. 191–203.

[7] Chu, S., Narayanan, S., and Kuo, C.-C. J. Environmental sound recognition with time-frequency audio features. *IEEE Trans. on Audio, Speech and Language Processing 17*, 6 (2009), 1142–1158.

[8] Ciavarella, C., and Paternò, F. The design of a handheld, location-aware guide for indoor environments. *Personal and Ubiquitous Computing 8* (2004), 82–91.

[9] Constandache, I., Choudhury, R. R., and Rhee, I. Towards mobile phone localization without war-driving. In *Proc. Intl. Conf. on Computer Communications (INFOCOM)* (2010), pp. 2321–2329.

[10] Davis, S. B., and Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on Acoustics, Speech, and Signal Processing 28*, 4 (Aug. 1980), 357–366.

[11] Dey, A. K., and Abowd, G. D. CybreMinder: A context-aware system for supporting reminders. In *Proc. Intl. Symposium on Handheld and Ubiquitous Computing (HUC)* (2000), pp. 172–186.

[12] Eronen, A., Peltonen, V., Tuomi, J., Klapuri, A., Fagerlund, S., Sorsa, T., Lorho, G., and Huopaniemi, J. Audio-based context recognition. *IEEE Trans. on Audio, Speech and Language Processing 14*, 1 (Jan. 2006), 321–329.

[13] Haeberlen, A., Flannery, E., Ladd, A. M., Rudys, A., Wallach, D. S., and Kavraki, L. E. Practical robust localization over large-scale 802.11 wireless networks. In *Proc. Intl. Conf. on Mobile Computing and Networking (MobiCom)* (2004), pp. 70–84.

[14] Härdle, W., and Steiger, W. Optimal median smoothing. *Applied Statistics 44*, 2 (1995), 258–264.

[15] Hightower, J., Consolvo, S., LaMarca, A., Smith, I., and Hughes, J. Learning and recognizing the places we go. In *Proc. Intl. Conf. on Ubiquitous Computing (UbiComp)* (Aug. 2005), pp. 159–176.

[16] Kang, J. H., Welbourne, W., Stewart, B., and Borriello, G. Extracting places from traces of locations. In *Proc. Intl. Wkshp. on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH)* (2004), pp. 110–118.

[17] Kim, D. H., Hightower, J., Govindan, R., and Estrin, D. Discovering semantically meaningful places from pervasive RF-beacons. In *Proc. Intl. Conf. on Ubiquitous Computing (UbiComp)* (2009).

[18] Kuttruff, H. *Room Acoustics.* Halsted Press, 1973.

[19] Lu, H., Pan, W., Lane, N. D., Choudhury, T., and Campbell, A. T. SoundSense: scalable sound sensing for people-centric applications on mobile phones. In *Proc. Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)* (2009), pp. 165–178.

[20] Ludford, P. J., Frankowski, D., Reily, K., Wilms, K., and Terveen, L. Because I carry my cell phone anyway: functional location-based reminder applications. In *Proc. Intl. Conf. on Human Factors in Computing Systems (CHI)* (2006), pp. 889–898.

[21] Moore, B. C. J. *An Introduction to the Psychology of Hearing.* Emerald Group Publishing, 2003.

[22] Oppenheim, A., and Schafer, R. *Discrete-Time Signal Processing.* Prentice-Hall, 1989.

[23] Otsason, V., Varshavsky, A., LaMarca, A., and de Lara, E. Accurate GSM indoor localization. In *Proc. Intl. Conf. on Ubiquitous Computing (UbiComp)* (Sept. 2005), pp. 141–158.

[24] Park, J.-g., Charrow, B., Curtis, D., Battat, J., Minkov, E., Hicks, J., Teller, S., and Ledlie, J. Growing an organic indoor location system. In *Proc. Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)* (2010), pp. 271–284.

[25] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. The Cricket location-support system. In *Proc. Intl. Conf. on Mobile Computing and Networking (MobiCom)* (2000), pp. 32–43.

[26] Proakis, J. G., and Manolakis, D. G. *Digital Signal Processing.* Prentice-Hall, 1996.

[27] Scott, J., and Dragovic, B. Audio location: accurate low-cost location sensing. In *Proc. Intl. Conf. on Pervasive Computing* (2005), pp. 1–18.

[28] Tarzia, S. P., Dick, R. P., Dinda, P. A., and Memik, G. Sonar-based measurement of user presence and attention. In *Proc. Intl. Conf. on Ubiquitous Computing (UbiComp)* (Sept. 2009), pp. 89–92.

[29] Want, R., Hopper, A., Falcão, V., and Gibbons, J. The active badge location system. *ACM Trans. Information Systems 10*, 1 (1992), 91–102.

[30] Ward, A. *Sensor-driven Computing.* PhD thesis, Corpus Christi College, University of Cambridge, Cambridge, UK, Aug. 1998.

[31] Woodman, O., and Harle, R. Pedestrian localisation for indoor environments. In *Proc. Intl. Conf. on Ubiquitous Computing (UbiComp)* (2008), pp. 114–123.

[32] Youssef, M., and Agrawala, A. The Horus WLAN location determination system. In *Proc. Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)* (June 2005), pp. 205–218.