

An Acoustic Identification Scheme for Location Systems

Raja Jurdak
UCI/School of Information
and Computer Science
Center for Pervasive Comm.
and Computing
rjurdak@ics.uci.edu

Cristina Videira Lopes
UCI/School of Information
and Computer Science
Center for Pervasive Comm.
and Computing
lopes@ics.uci.edu

Pierre Baldi
UCI/School of Information
and Computer Science
California Institute for
Telecomm. and Information
Technology Cal-(IT)²
pfbaldi@ics.uci.edu

Abstract

Many pervasive computing applications require location awareness in order to successfully integrate technology into our daily lives. A location system consists of a group of sensors that determine the position of a mobile user and provide the user with useful context-specific information. Locating the user requires signal exchanges between the user's mobile device and the sensors. This paper considers an acoustic-based location system for pervasive computing applications. The system is comprised of a set of microphones connected to a central server. Mobile users produce acoustic signals through standard speakers, which are already available in most mobile devices, to perform location operations through the system. We focus on the design of robust acoustic signals using multi-frequency symbols that serve both in locating and uniquely identifying the user. We conduct experiments at distances between 1 and 17 ft to explore the ability of the server to recognize and decode signals originated by different users in the same general area.

1. Introduction

Most pervasive computing applications require location awareness in order to successfully integrate technology into our daily lives. A location system consists of a set of sensors that determine the position of a mobile user and provide the user with useful context-specific information. Locating the user requires signal exchanges between the user's mobile device and the sensors.

Here, we present work that is part of a project to design an acoustic location system that combines the benefits of a computer network with the intrinsic worth of location-awareness for roaming users in the physical world. In this

context, a user equipped with a roaming device will be able to obtain on-demand user and location-specific information. Applications for this system include cheap and easily deployable location systems, extensions to web protocols, and location-dependent multi-user games. Areas of deployment include places that have been to a large extent abstracted on the Internet, such as retail stores, shopping malls, museums, amusement parks, and other places where people seek related information, and eventually pay for products and services. The main contribution of this paper is to explore the design of robust acoustic signals for locating and identifying multiple users within the coverage area of the location system.

The rest of the paper is organized as follows. Section 2 reviews the previous related work. Section 3 provides an overview of the operation of the acoustic location system. Section 4 discusses the frequency encoding and transmission of symbols which are the building blocks of user ID's in our system. Section 5 discusses mainly the mechanisms involved in detecting the frequency content and in decoding the symbols contained in the received acoustic signal. Section 6 presents the experiments we performed to validate this encoding scheme for acoustic location and identification. Section 7 discusses the results of the experiments and concludes the paper.

2. Related Work

Several location systems [1] have been proposed to provide location awareness in a ubiquitous computing environment. In Active Badge [2], which is one of the earliest proposed location systems, users wear badges that emit diffuse infrared signals. Pre-installed sensors detect the infrared signals and report them to a central server to determine the user's location. Infrared waves have several undesirable features for location systems, including interference from

fluorescent lighting and sunlight.

Other location systems, such as Active Bat [3] and Cricket [4], rely on ultrasound signals. Active Bat's architecture is similar to Active Badge in that it requires mobile users to wear ultrasound tags, and ceiling-mounted ultrasound receivers capture the tag's signal and report it to the central server. Active Bat uses an ultrasound time-of-flight lateration technique, in which the user sends both an ultrasound and radio signal, and the system computes the difference in arrival times between the two signals to determine the user's position. Cricket enhances Active Bat by using the radio signal arrival time to narrow the time window in which arriving signals are considered. Dolphin [5] is another ultrasound positioning system that has a distributed flavor. In Dolphin, the location of only a few nodes is known, and the remaining nodes can infer their own location based on the location of the reference nodes. Nodes in Dolphin also send messages periodically to advertise their position and to maintain synchronization.

Because of their reliance on technologies such as infrared and ultrasound, most existing location systems often require the user to carry additional hardware such as badges or tags. Requiring additional specialized hardware on the user side introduces cost and feasibility issues, which in turn limit the large-scale proliferation of existing systems.

Other location systems proposed the use of hardware that is already found in mobile devices. RADAR [6] uses the signal to noise ratio and signal strength of a mobile user's IEEE 802.11 [7] transmissions to locate the user in a 2 dimensional environment. One drawback of the RADAR system is its assumption that the mobile device is equipped with IEEE 802.11, which does not apply to all mobile devices. Security and privacy also arise as important issues in radio frequency location systems: the system can track the user without the user knowing it.

Unlike WLAN technologies that are not available in smaller mobile devices and that are protocol dependent, the acoustic interface is available in virtually all mobile devices and is universally compatible. Acoustic technology has been recently considered for ubiquitous computing and communications applications. Lopes and Aguiar [8] have explored the use of musical sounds or other familiar sounds for low bit rate communications using hardware that is readily available in desktop computers, palm devices, memo recorders, televisions and other electronic devices. Similarly, our aim is to use sounds that are easily reproducible by most mobile devices for indoor location.

The work in [9] considers an outdoor location system based on a network of acoustic sensors to provide high location accuracy at considerable monetary cost for military and scientific applications. The system in [9] assumes a fully distributed self-organizing architecture where the sensors discover the topology and integrate into the network,

which adds complexity and cost to the sensors themselves. In contrast, the design goal of our system is the development of an indoor acoustic positioning system with reasonable accuracy for cheap and easy deployment. Our system adopts a centralized topology, where microphones are only input devices through which the acoustic signals are relayed to the centralized server. Furthermore, the system in [9] employs complex algorithms for sensor synchronization, as well as beamforming techniques to determine the direction from which the signal arrives at the microphone. On the other hand, our system uses simple UDP sockets for temporary synchronization between the server and sensors, which eliminate the overhead for continuous synchronization between the sensors. Our system also replaces complex beamforming techniques with basic triangulation¹ at the server to reduce the latency of a location operation.

The work in [10] also proposes the use of acoustic waves for indoor location systems, as well as for low bit rate communications. In [10], Madhavapeddy et al. consider several audio modulation techniques, including Dual Tone Multi-Frequency (DTMF), melodic sounds, and inaudible signals at the border of the acoustic range. In Madhavapeddy's acoustic location architecture, one of several listeners detect the acoustic signal and report the signal characteristics to a central server. Our system uses a similar architecture to determine the user's location but with finer granularity. While the location system in [10] aims to identify the room in which the user is located, our system employs acoustic signals to locate a user's approximate position within the room.

3. System Overview

3.1. Signal Description

Mobile users in Madhavapeddy's acoustic location scheme emit a single tone signal at 800 Hz. From a technical viewpoint, we believe that the use of a single acoustic tone to locate and identify users is insufficient because frequency-specific noise impulses (or their harmonics) in some indoor environments may prevent the listeners from capturing the correct signal from the user. Furthermore, using a single tone signal does not support multiple simultaneous users. Madhavapeddy's main motivation in using a single frequency for the location operation is that the base capability of mobile phones is to produce monophonic tones. However, phones capable of producing polyphonic tones are becoming increasingly popular. Other mobile devices

¹Triangulation at the server uses multiple distance measurements from different microphones [1]. Around each microphone, the server considers a sphere with the reported distance of that particular microphone. The location of the user is then computed as the intersection of the spheres around the microphones that report the shortest distances.

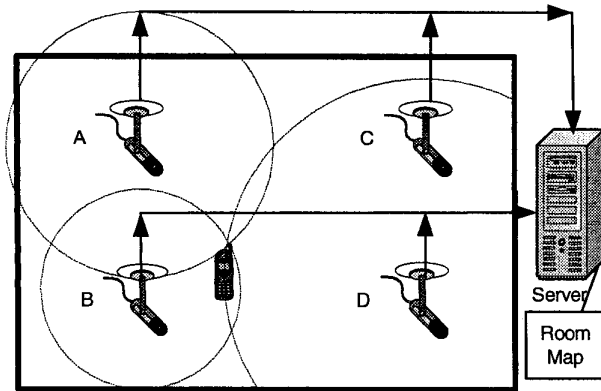


Figure 1. System Architecture and Location Operation.

such as handheld devices and pocket PC's are already capable of producing polyphonic tones. Thus, our acoustic location and identification scheme combines two variants of Madhavapeddy's proposed modulation techniques:

1. We use 3 frequencies to represent each symbol in our alphabet, while DTMF uses 2 frequencies to represent each symbol in touch/tone phones. Using 3 frequencies provides redundancy that increases the signal's robustness to frequency-selective interference and fading.
2. We choose the frequencies that belong to the same major musical scale, as suggested in [8], to represent each symbol in our alphabet.

3.2. Architecture

The system architecture includes a set of listening devices (microphones) that are connected to a central server, as shown in Figure 1. Figure 1 shows a simple scenario where four microphones, which are mounted in a grid formation on the ceiling of a room, are also connected to a server that stores a map of the room. The distance between microphones is assumed to be sufficiently small to ensure that at least three of the microphones detect the acoustic signals from any area within the room.

3.3. Location Operation

We illustrate the sequence of events involved in a location operation through the following example. Consider a customer that is seeking directions to the appliances section in a large department store, which already has a deployed

wireless data network such as IEEE 802.11 [7] or Bluetooth [11]. The customer's mobile device is also assumed to have a wireless data communications capability. Consequently, the store data network can assign the mobile device an IP address through DHCP so that the location system can identify the device on the data network.

The customer can issue a location query through his mobile device to request directions, and the server subsequently provides the requested directions. However, the server must first determine the current location (context) of the customer within the store before providing the customer with detailed directions to the appliances section. To enable the server to locate the user, the mobile device sends its ID (see Section 4) in the form of acoustic signals. At least three of the ceiling-mounted microphones detect these acoustic signals. In the topology of Figure 1, microphones A, B and D detect the user's acoustic ID. In general, each microphone that detects the signal reports it to the central server. The server should then estimate the distance of the user from each microphone. Using absolute signal strength to estimate distance is not reliable in this application because the amplitude of the acoustic signal varies among mobile devices. Instead, the server uses time measurements to estimate the distance of the user from each microphone. The steps involved in distance estimation are the following:

1. While the server is constantly listening for client requests, the client on the mobile device notifies the server through the wireless data network that it is about to send its acoustic ID to initiate a location query.
2. The client then immediately sends the acoustic signal encoding of its ID. The server can compute the latency between the end of the notification signal received on the data network and the beginning of the acoustic signal. Based on this latency and the propagation characteristics of both signals, the server estimates the distance of the user from a single microphone.

Our preliminary experiments have shown that the server can determine the distance of the sound source located at a distance of up to 22 ft with an error of 6 inches for a single frequency signal. After estimating the distance of the user from each microphone, the server uses triangulation to determine the current location of the user, as shown in Figure 1. Once triangulation yields an estimate of the user's location, the server uses its locally available store map to determine the proper directions from the user's current location to the appliances section. The server can then send directions to the user through the wireless data network.

In other settings where the mobile device does not have a separate communications capability, the system could use relative received signal strengths from various microphones to estimate the user's location. Subsequently, the system can use a modulated acoustic signal, as described

in [8] and [10], to provide directions to mobile devices that are equipped with microphones.

4. Signal Encoding and Transmission

4.1. Acoustic ID's

We now focus on the structure of the acoustic signal. The signal encoding serves both in locating the user and in uniquely identifying the user among all current users within the room. The purpose of embedding a unique ID into the location signal is twofold. First, in the absence of a separate data network with its own addressing scheme, acoustic ID's provide a way to uniquely identify users in the coverage area and to route information to users. Secondly, if two or more users simultaneously issue location queries, ID's enable the server to determine the source of each query.

Assigning unique ID's requires mobile devices to register with the server upon entering the coverage area of the location system. During the registration process, the user requests an acoustic ID along with an IP address to identify him/her uniquely among other users within the coverage area. Once the system assigns the user an ID and an IP address, the user can roam within the coverage area and use the assigned ID to initiate location operations.

4.2. ID encoding

We use a coding scheme similar to the DTMF scheme to encode each symbol. DTMF encodes each digit using 2 separate frequencies in the range 697 to 1633 Hz. Frequencies in DTMF are classified as high and low frequencies, and DTMF encodes each digit using one low and one high frequency. As a result, some pairs of digits in DTMF have one frequency in common. Consequently, if a receiver detects only one of the two frequencies of a DTMF symbol, it is impossible to decode that symbol. This becomes more of an issue for wireless acoustic communication, which has higher signal losses than the intended application area of DTMF in wired phone lines. Even if no frequencies are lost, the common frequencies among digits in DTMF may prevent receiver from deterministically identifying 2 digits received simultaneously over the air. Thus, we propose enhancements to DTMF in our coding scheme to provide a more robust signal that meets the needs of wireless acoustics.

4.2.1 Symbols and Frequencies

Our coding scheme uses an alphabet of 3 symbols, which is easily extendible to 6 or more symbols. These symbols form the building block of user ID's, since each combination of symbols represents a unique user ID. Each symbol

Digit	Freq 1	Freq 2	Freq 3	Scale
1	2794	3520	4186	F Major
2	3136	3951	4699	G Major
3	2489	2960	3729	D# Major

Table 1. Frequency Encoding of Symbols

is encoded using 3 frequencies, and unlike DTMF, there are no common frequencies among any two symbols. The choice of using 3 frequencies to encode each symbol stems from the fact that the microphones may not detect all the transmitted frequencies. This redundancy, together with the property that each frequency correlates only to one symbol, ensures that symbols are correctly decoded by the server even if some frequency components are lost. Our scheme requires the receiver to detect two out of the three frequencies² that represent a symbol to decode that symbol.

We use frequencies in the range of 2200 to 4700 Hz, since this band is less susceptible to indoor background noise, which we observed to be at frequencies below 2 Khz. Another motivation for using frequencies in the range 2.2 Khz-4.7 Khz is that the speakers of many mobile devices operate well in that range.

Unlike other communication technologies, acoustic signalling and communications can be perceived by humans. As a result, any wireless acoustic system should ensure that the emitted signals are pleasant, or at least tolerable, to the human auditory system. To address this issue, each set of three frequencies that represent a symbol in our system belong to the same major musical scale, so that sending an audio signal for a single symbol does not annoy users.

As mentioned earlier, the coding scheme encodes 3 symbols using 9 frequencies, where each symbol is the sum of three sinusoidal signals at frequencies which lie on the same major scale. Table 1 indicates the frequencies that encode each symbol, and the musical scale on which each set of three frequencies fall. Note that all frequencies in Table 1 are separated by at least 150 Hz, so that any frequency shifts due to hardware variations do not result in false ID's.

In addition to the musical scales in Table 1, we are exploring several other musical scales, such as the blues major scale, the blues minor scale and the flamenco scale, for producing acoustically pleasant ID's.

²The requirement of receiving two out of three frequencies for each digit may be overly redundant for some environments with predictable noise patterns. This requirement could be relaxed to one out of three frequencies, which would increase the range of detection of a user, but it would also increase the probability of error in location and identification. The modified scheme could also triple the number of users by using one frequency to identify each user, at the cost of decreased reliability.

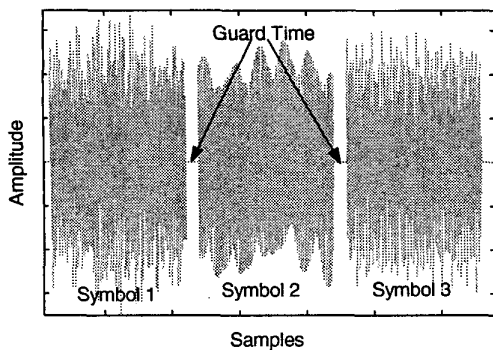


Figure 2. Signal shape of the transmitted ID [1,2,3].

4.2.2 Reverberation

Ideally, a mobile device would send each symbol immediately after the preceding symbol. In reality however, acoustic waves in a room undergo reverberation due to the reflection of sound within the room [12]. Reverberant sound in a room dies away as the sound energy is absorbed by multiple interactions with the surfaces of the room. Thus, the ID encoding scheme requires a guard time between each two symbols to reduce the effects of reverberation from one symbol to the next. Figure 2 plots the signal of ID [1,2,3] with the symbols separated by 9.7 millisecond guard times. Figure 3 shows the same ID as it is captured by the receiver. The symbol boundaries in Figure 3 are still visible at the guard times, where only reverberant sound with decaying amplitude is present. In addition to guard times, our algorithm compares the amplitude of the frequencies in arriving signals. If during a time slot the ratio of amplitudes of two symbols is above a certain threshold, then the symbol with the lower amplitude is regarded as reverberation from neighboring time slots.

4.3. Synchronization

Our scheme requires mobile devices to issue a hello signal at the beginning of an acoustic transmission to allow the server to synchronize to the transmission. We define the hello signal in our system as a sinusoidal signal at a known frequency. We reserve the frequency of 2200 Hz as the hello frequency. This value for the hello frequency is appropriate because: (1) it is within the optimal range of operation of speakers and microphones; (2) it maintains a guard band of 200 Hz from the noisy indoor spectrum below 2000 Hz; (3) it maintains sufficient separation from the nearest symbol frequency, which is at 2489 Hz; (4) its harmonic at 4400 Hz maintains ample separation from nearby symbol frequencies.

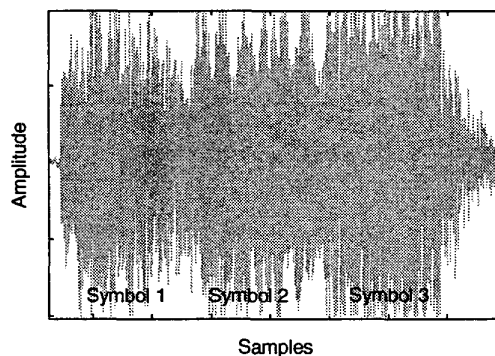


Figure 3. Signal shape of the received ID[1,2,3]: The amplitude values are normalized.

A related design choice for the hello signal is whether to send it just before the start of an ID, or to embed it within the first symbol of an ID. Initial testing of these two cases has revealed that the server synchronizes more accurately to the signal when the hello signal is embedded in the first symbol of the ID. Furthermore, synchronizing to the hello signal requires that it is at least a few milliseconds long, which implies that sending the hello signal prior to the ID incurs more delay in the decoding process. Consequently, our scheme includes the hello signal with the first symbol of each ID.

5. Signal Reception and Decoding

5.1. ID Decoding

We begin by describing the decoding procedure for one symbol, and we later extend that process to decode sequences of symbols that represent ID's. Once the acoustic signal is received by the microphone, the signal must be decoded in several steps with increasing granularity. First, the receiver discovers the beginning of the useful signal by synchronizing to the hello frequency. Once the server synchronizes to the mobile device's signal, it partitions the acoustic signal into time slots with one symbol duration per time slot. The server then proceeds to discover the frequency content in each time slot. We use the Fast Fourier Transform (FFT) to transform the time domain acoustic signal in each time slot to the frequency domain. Let z be the received time domain acoustic signal after sampling, and Z be the frequency domain representation of z . To derive the frequency content of Z , we use the equation:

$$P_{zz} = Z \times \text{conj}(Z) / N_Z \quad (1)$$

where P_{zz} is the power spectrum of the signal, and N_Z is the number of points (samples) in the signal.

Next, we implicitly filter the signal by examining the samples of P_{zz} corresponding to the frequencies in Table 1. In fact, for each frequency f in Table 1, we examine the values of components of P_{zz} in a small range of frequencies centered around f in order to account for hardware imperfections that lead to frequency distortion. As we mentioned above, the frequencies in Table 1 are spaced far enough from each other so that a shift in one frequency does not crossover to the window of the neighboring frequency. If the value of P_{zz} exceeds a threshold value at or close to a frequency f , then the server determines that f is present in the signal. Section 6.1 provides a more detailed discussion of setting the threshold. Once two or more frequencies that correspond to the same symbol are detected, the server determines that the symbol is present in the signal. By performing this quick check in subsequent time slots, the server can decode a sequence of digits to get the received ID.

5.2. Synchronization

Discovering the frequency content using the FFT method is applicable to individual time slots once the boundaries of each time slot are defined. Defining the boundaries of time slots requires discovery of the beginning of the first time slot, which is achieved by synchronizing to the hello frequency.

Once the server captures the signal, it uses a sliding window technique to discover the start of the signal. First, the server examines chunks of the signal that are equal in length to the embedded hello signal. Once it identifies the section of the received signal with the highest correlation to the hello signal, the server then narrows the window of search for the hello signal to that section. Within that section, the server examines the signal samples in smaller chunks. By progressively shrinking the window size and the chunk size, the server eventually identifies the sample that corresponds to the beginning of the hello signal, which is also the beginning of the first time slot in the ID (see section 4.3).

Synchronization to the beginning of ID's is even more valuable when several mobile users simultaneously perform location operations. Figure 4 illustrates the case of a mobile device sending its ID while the server is still in the process of receiving another user's ID. Suppose the server receives the acoustic ID of mobile device A at time t , and the duration of a symbol is T . The receiver synchronizes to A's transmission by detecting the start of the hello signal, and begins decoding the ID of A. Suppose also that at time $t+a$, where a is shorter than the duration of an ID, the receiver hears another transmission from user B. In Figure 4, the acoustic ID of B arrives at the server during the second time slot of A's transmission. While decoding the second sym-

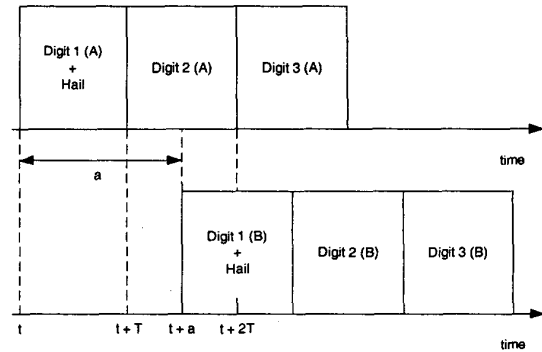


Figure 4. Synchronization

bol in A's ID, the server detects that the hello frequency is present in the signal. As a result, the server synchronizes to the new transmission by performing a correlation check to synchronize to the hello signal of the new transmission, and initiates a separate thread to derive the frequency content of this transmission. Between time $t+a$ and the end of A's ID, the frequency content of the received signal is the combination of symbols sent by A and B. The FFT method detects up to six frequencies in each time slot, but the method cannot determine the source of each frequency. Thus, identifying the received ID's requires the following additional steps.

User A's second symbol had started arriving at time $t+T$, so the server can identify the frequencies of A's second symbol by checking the frequency content between $t+T$ and $t+a$. The server can also determine B's first digit by omitting A's frequency content from the total frequency content received between time $t+a$ and $t+2T$. To decode subsequent symbols of each ID, the server determines the frequency content of each time slot in the same way.

6. Experiments and Results

In order to validate our ability to detect multiple frequencies and symbols simultaneously through acoustic signals, we conduct experiments in an office of dimensions $18 \times 9 \times 9$ ft, using a typical PC microphone and speakers. In the experiments, the PC speakers send the frequencies corresponding to the encoded symbols through the air. The microphone then captures this audio signal, and once the sound card samples the data, we use Matlab [13] to perform an FFT analysis to determine the frequency content of the captured signal. Table 2 contains the parameter values in our experiments.

Each experiment was performed on five different occasions, and the results reflect the average of the five trials. The variance of results among separate trials of each exper-

Table 2. Parameter Values

Parameter	Value
Sampling Frequency (hz)	11025
bits/sample	8
Symbol duration (samples)	1000
Symbol duration (ms)	90.7
Guard time (samples)	100
Guard time (ms)	9.07
Room Dimensions (ft)	18 x 9 x 9

iment was small, which suggests deterministic behavior of the system.

The first set of experiments explores the effects of distance, frequency selection, and number of simultaneously emitted frequencies on our ability to detect transmitted frequencies. The second experiment set investigates the decoding capability of the server to synchronize to and decode two asynchronous acoustic ID's that overlap in time.

6.1. Frequency Detection

We first explore the effect that simultaneous symbol transmissions has on the server's ability to detect individual symbols. A central feature of frequency detection ability in our scheme is the threshold value in the FFT method (see Section 5.1). Setting the threshold value involves a tradeoff: a high threshold value insures that noise is not mistaken for a useful signal, but it also results in failure to detect some frequencies. On the other hand, setting the threshold too low allows detection of more frequencies, but it also increases the chance of false positives from noise. Thus, for each number of simultaneous symbols, we observe the minimum threshold value within the FFT method that results in detection of all frequencies. We also use random combinations of the symbols (1,2,3) to avoid any biases introduced by frequency-specific behavior of the hardware used in our experiments.

Figure 5 shows the relative threshold value for the server to detect all the transmitted frequencies. For distances ranging from 1 ft to 17 ft, the server could detect all the frequencies in the cases of 3, 6 and 9 simultaneously transmitted frequencies. However, the required threshold value for detecting each frequency using the FFT method (see section 5.1) becomes lower as the number of simultaneous frequencies increase.

The detection threshold for 6 simultaneous frequencies (2 simultaneous symbols) is about 44% of the detection threshold of one symbol. Also, the threshold for 9 frequencies (3 symbols) is 36% of the 1 symbol threshold. The minor decrease in the detection threshold as the number of frequencies increases from 6 to 9 suggests that the received signal level degrades more slowly as more frequencies are

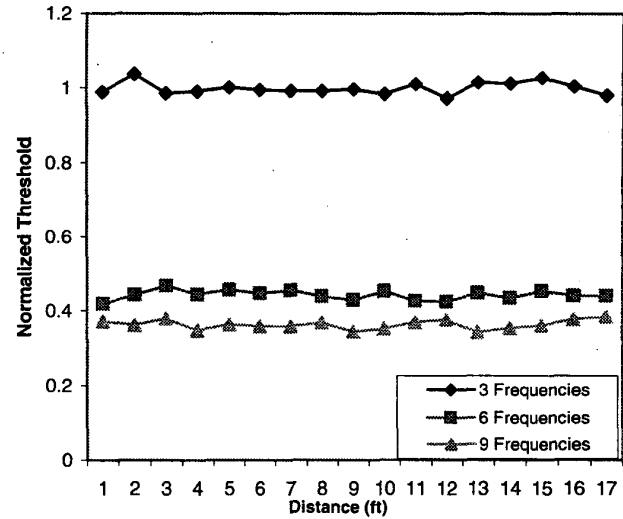


Figure 5. The normalized threshold value in FFT method for full frequency detection as a function of distance.

added. This feature is desirable for the scalability of the scheme.

The threshold and distance experiments revealed distinct channel responses for each combination of frequencies. To understand the channel frequency response for our hardware and to quantify these distinctions, we studied the required threshold for different symbol combinations. Because acoustic location systems do not necessarily use the same hardware or frequency set, the results in this study apply only to our particular setting. These results also serve as a basis for the development of a general process that calibrates thresholds for any hardware or frequency set.

Figure 6 provides a threshold comparison of all possible symbol combinations. Symbol (3) has the highest threshold among all symbol combinations. Symbol (2) and symbol (1) have respective relative thresholds of 0.7 and 0.19. Two symbol combinations follow similar trend to the single symbol case, where the pair of symbols with the highest threshold is (2,3), and the pair with the lowest threshold is (1,2). The combination of the three symbols (1,2,3) has a threshold that is higher than both combinations (1) and (1,2). The results have 2 clear implications: (a) Symbol (1) contains frequencies with unfavorable channel response, and thus all combinations containing this symbol have low thresholds. (b) Symbol (3) contains frequencies with favorable channel response, and the presence of symbol (3) in a time slots improves channel response for other symbols (1 and 2).

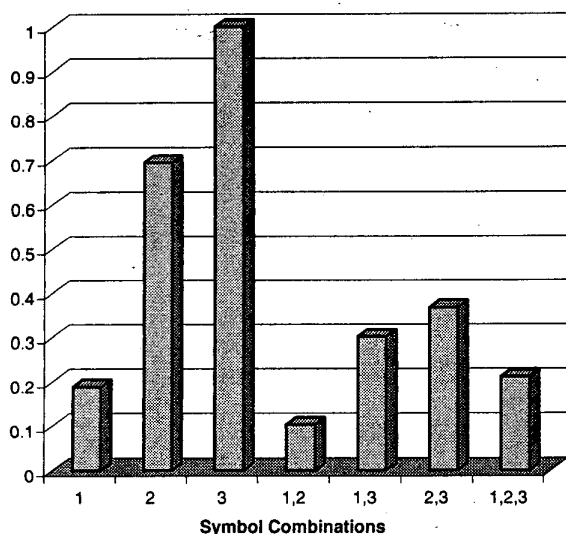


Figure 6. A comparison of the normalized thresholds of frequency combinations.

6.2. Symbol Decoding

The next set of experiments investigates the server's ability to recognize two distinct ID's arriving asynchronously. These experiments reveal that the main issue of correct decoding is synchronization to the hello frequency through the correlation algorithm (see Section 5.2). As the distance increases, the hello signal strength at the server is lower, and the correlation check at the server is more likely to yield an incorrect beginning of the signal. This degradation with distance applies mainly to synchronization to the hello signal, and does not affect the detection of the frequency content as the experiments in Section 6.1 reveal. Thus, we explore the effect of both the hello frequency amplitude and the distance on the server's capability to decode 2 asynchronous symbols.

Figure 7 plots the percentage of symbols that are correctly decoded by the server as the distance between the users and the sensors varies from 1 ft to 17 ft. Each of the three plots in Figure 7 is characterized by the ratio of the amplitude of the hello frequency to the amplitude of the symbol frequencies. For example, the plot for "Amplitude=2" indicates that the hello frequency amplitude is double the amplitude of the symbol frequencies.

When the hello frequency amplitude is the same as the amplitude of all other frequencies, the server can synchronize to and decode only one of two acoustic ID's arriving asynchronously. The decoding ability does not vary with distances of 1 to 13 ft, as the percentage of decoded ID's remains around 50%. Occasional peaks in the decoding

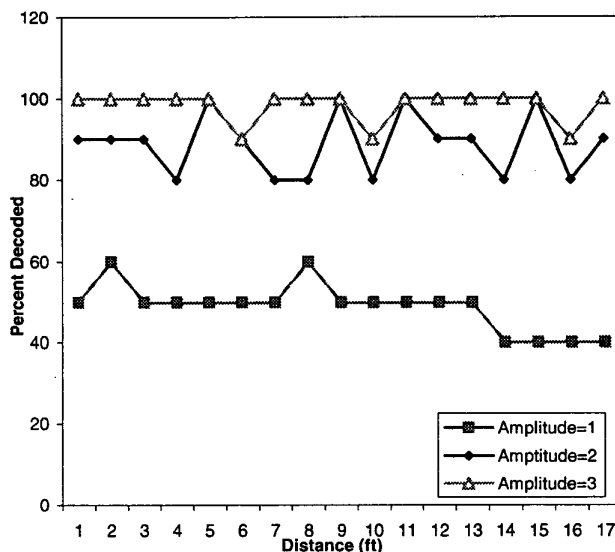


Figure 7. Percentage of decoded symbols versus distance.

ability at distances of 2 ft and 8 ft are due to both hardware variations and multi-path reflection effects within the experiment location. For distances of 14 ft or more, the decoding ability drops to 40% because the hello frequency attenuation at these distances makes it more difficult to synchronize properly to the acoustic ID. On average, the server could decode 49% of the transmitted ID's for all distances within 17 ft.

If the hello frequency amplitude is doubled, the decoding ability at the server improves significantly. First, doubling the hello frequency amplitude eliminates the effect of distance on the decoding ability. As Figure 7 shows, the trend in decoding ability is constant for all distances within 17 ft. A second observation is that the decoding ability varies between 80% and 100% in a seemingly random fashion. Again, this is attributed to multi-path effects and to variations in the delay and response of the sound card according to the interrupt behavior and processing load at the host device. Overall, the server could decode 89% of acoustic ID's.

Finally, the plot for "Amplitude=3" eliminates much of the variation in decoding ability that was observed for the previous case. Because the hello frequency amplitude is now triple that of symbol frequencies, the server can synchronize better to both acoustic ID's at all distances within 17 ft. The decoding ability of the server is perfect for most distances, and on average the server could decode 98% of the transmitted acoustic ID's.

7. Discussion and Conclusion

7.1. Background Noise

Some of the results were derived with music and other clutter noise sources in the background. Although exposure to noise sources was not systematic in all experiments, the ability of the server to decode symbols in the presence of clutter noise boosts our confidence in the coding scheme.

7.2. Thresholds

Detection of symbol frequencies is not as dependent on the amplitude as synchronization to the hello signal, primarily because the server can detect symbol frequencies as the mobile device moves further from the receivers by lowering the detection threshold in the FFT method. Further research is required on algorithms to set these thresholds adaptively, especially when there are simultaneous transmissions from nodes that are both near and far from the microphones. Because the server aggregates the received signals from several microphones, the effect of simultaneous transmissions from near and far nodes should be minimal.

7.3. Range

The receiver could detect all frequencies within a range of 17 ft, and the hello frequency amplification enables the decoding of 2 asynchronous symbols at up to 17 ft. Future work will explore an extended range at distances up to the single frequency detection range (currently 22 ft) in larger deployment areas. The experiments will reveal the applicability of multi-frequency ID's over larger ranges and in more diverse multi-path environments.

The importance of range in this system is offset by the fact that microphones are cheap. Thus, deploying a dense grid of ceiling-mounted microphones, each with a limited signal detection range, is cost-feasible. A dense grid of microphones would enable the location system to capture any acoustic signal within the coverage area.

7.4. Calibration

Because the aerial acoustic channel includes the speakers, air, and microphone, channel behavior may be hardware-specific. Thus, calibration may be needed at some point prior to sending the acoustic ID's. The results in Figure 6 provide valuable insight into the hardware response to the current choice of frequencies. For example, the channel has unfavorable response for the frequencies of symbol 1. Also, the channel has nonlinear behavior for various frequency combinations. Observing the specific responses of

several speaker and microphone sets provides understanding on the trends in channel response, which can subsequently be developed into a calibration process.

7.5. Security and Privacy

This acoustic location and identification system provides a user with valuable context-specific information, but it also raises security and privacy issues. For example, department stores could track users' movements and behavior within the store, and use data mining techniques to market specific products to users. Therefore, the location system should include security measures to ensure the privacy of users.

In sum, we have presented a model of an acoustic location and identification system that uses unique user ID's based on multiple-frequency symbols. We have investigated the range of detection of 1, 2 and 3 simultaneously transmitted symbols using 3, 6 and 9 frequencies respectively. The results have yielded a range of at least 17 ft for up to 9 simultaneous frequencies. Our other experiment has revealed that decoding asynchronous symbols at the server is highly dependent on synchronizing to the hello signal. To overcome this issue, we have proposed an increase in the amplitude of the hello signal by a factor of 2 or 3. Once the server synchronizes to the signal, symbol decoding becomes independent of distance. As a result, the acoustic identification scheme scales well for indoor location systems deployed in large rooms.

8. Acknowledgment

This work is part of a larger project that is funded by the UC MICRO Program. Raja Jurdak is in part supported by a grant from the Center for Pervasive Communications and Computing at UCI. The work of Pierre Baldi is in part supported by a Laurel Wilkening Faculty Innovation award, a Sun Microsystems award, and grants from NSF and NIH.

The authors would like to thank Atri Mandal and Amir Haghghat for their valuable insight and discussions on acoustic positioning and triangulation. The authors also thank the anonymous reviewers.

References

- [1] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer*, Volume 34, No. 8, pp 57-66, August 2001.
- [2] R. Want et al. "The Active Badge Location System," *ACM Trans. Information Systems*, 1992, pp. 91-102.
- [3] A. Harter et al. "The Anatomy of a Context-Aware Application," *InProc. 5th Ann. Intl Conf. Mobile Com-*

- puting and Networking (*Mobicom 99*), ACM Press, 1999, pp. 59-68.
- [4] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. "The Cricket Location-Support System," *Proc. 6th Ann.Intl Conf. Mobile Computing and Networking (Mobicom 00)*, ACM Press, 2000, pp. 32-43.
 - [5] Y. Fukuju, M. Minami, H. Morikawa, and T. Aoyama. "DOLPHIN: An Autonomous Indoor Positioning System in Ubiquitous Computing Environment," *In Proceedings of IEEE Workshop on Software Technologies for Future Embedded Systems(WSTFES2003)*, 2003, pp.53-56.
 - [6] P. Bahl and V. Padmanabhan. "RADAR: An In-Building RF-Based User Location and Tracking System," *In Proc. IEEE Infocom 2000*, IEEE CS Press 2000, pp. 775-784.
 - [7] IEEE 802.11 WG. "Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999. Standard.
 - [8] C. V. Lopes and P. Aguiar. "Acoustic Modems for Ubiquitous Computing," *IEEE Pervasive Computing, Mobile and Ubiquitous Systems*, Vol. 2, No. 3, pp 62-71, July-September 2003.
 - [9] J.C. Chen, L. Yip, J. Elson, H. Wang, D. Maniezzo, R.E. Hudson, K. Yao, and D. Estrin. *In Proceedings of the IEEE*, Vol. 91, No. 8, August 2003.
 - [10] A. Madhavapeddy, D. Scott and R. Sharp. "Context-Aware Computing with Sound," *In The Fifth International Conference on Ubiquitous Computing*, October 2003.
 - [11] "The Bluetooth Special Interest Group," 1999.
 - [12] F. A. Everest. *The Master Handbook of Acoustics*, New York : McGraw-Hill, c2001.
 - [13] Matlab, <http://www.mathworks.com/>.