Localization of Off-the-Shelf Mobile Devices Using Audible Sound: Architectures, Protocols and Performance Assessment *

Cristina V. Lopes Amir Haghighat Atri Mandal Tony Givargis Pierre Baldi
{lopes, ahaghigh, mandala, givargis, pfbaldi} @ics.uci.edu
California Institute for Telecommunications and Information Technology (CalIT2)
and
Bren School of Information and Computer Sciences, University of California, Irvine, CA, USA

Sound source localization will play a major role in the new location-aware applications envisioned in Ubiquitous Computing. We describe the design and performance of three architectures and corresponding protocols that use a variation of the Time-of-Flight method for localizing three different kinds of devices, namely 802.11-enabled PDAs, 3G cell phones, and PDAs without network connectivity. The quantitative assessment is based on the deployment made with 6 sensors in a 20x9m room, serving over 10,000 localization requests. Our experiments indicate that all architectures achieve localization within 70cm of the actual position 90% of the time. The accuracy is further improved to 40cm 90% of the time when geometric factors are taken into consideration. The effects of noise and obstructions are also analyzed. Within 1m localization error, realistic noise degrades the accuracy by 6 to 10%. The presence of obstacles, such as humans and cement columns, has no observable effect on the performance.

I. Introduction

Context-awareness is one of the major components of pervasive, or ubiquitous, computing. "Context" can be broadly defined, but it includes some notion of location. Location-aware applications interpret the location data obtained from localization systems and provide the users with specific information depending on their location. We have developed WebBeep, a cheap and easily deployable location-aware system for indoor use [4, 10]. The main goal behind WebBeep is to add location-dependent virtual browsing experience to users who are navigating in an indoor physical space with a roaming device. The users are provided with location-specific, dynamically generated web pages. A target space for such a system is, for example, a supermarket. In this deployment scenario, the generated location-dependent web page includes information such as promotional items close to the user, items they have bought last time they were at that location, etc. The users can also search for an item and see the location of that item, as well as their own location on a map generated by the web server.

In the process of developing WebBeep, we considered several indoor localization systems. Unfortunately, existing systems use specialized hardware which is neither easily available nor easily attachable to the variety of off-the-shelf mobile devices we are considering. Therefore, we developed an indoor localization system, called Beep, that is both cheap and universally applicable, as it senses audible sound – a medium available in virtually all roaming devices. The use of audible sound eliminates the need for additional infrastructure at the user end. Furthermore, localization techniques based in sensing audible sound have the potential to be applicable to a large range of applications involving the localization of human-made sounds.

This work is a comprehensive exploration of the opportunities and limitations on the use of audible sound as localization medium for off-the-shelf mobile devices. The reminder of the paper is organized as follows. Section III sets the mathematical foundations of the problem of positioning with audible sound, and describes our estimation technique. Section IV describes several architectures and protocols that we have experimented with. Section V presents the extensive experimental results, section II features the related work, and section VI concludes the paper.

*This work has been supported in part by the University of California MICRO grant no. MS-33822, in partnership with Microsoft Corporation, and by the National Science Foundation Award No. CCF-0347902. A fraction of this work has been published in the Proceedings of IEEE Consumer Communications and Networking Conference (CCNC'05), in January 2005. The current submission extends the previous work with new architectures and protocols, as well as with extensive experimental results.

II. Related Work

Previous work related to the development of indoor localization systems has been reviewed in a comprehensive survey by Hightower and Borriello [5]. In this section we discuss systems that use sound as the localization medium.

Active Bat [3] is the pioneering positioning system in using ultrasound as the medium for localization, and Beep takes inspiration on its use of the Time-of-Flight (TOF) lateration technique. In Active Bat users carry tags equipped with radio frequency (RF) transceivers and ultrasonic speakers, while base stations equipped with the same RF technology and ultrasonic microphones are mounted on the ceiling. For localization to take place, the tag synchronizes with the base stations via an RF signal and emits an ultrasonic pulse at the same time. Active Bat has been reported to achieve an accuracy of 9cm 95% of the time, but base stations have to be precisely mounted on the ceiling in large numbers to accommodate the required sensor density.

Cricket [12] also uses RF and ultrasound, but it has a different design goal than Active Bat and Beep. In this system the beacons advertise their location (physical or symbolic) to listeners carried by the users. The location advertisements and beacon-listener synchronization are performed via RF. The listeners then use ultrasound TOF measurement in order to determine their distance from each beacon in their range. Hence, a listener can either adopt the closest beacon's advertised location as its own, or perform lateration on three TOF measurements to infer absolute location. Cricket has been reported to perform 100% accurately on 1.2x1.2m regions.

Aside from infrastructure costs, the above systems impose their specialized hardware on end-users, i.e. tags and listeners. Beep differs from that work in that it takes advantage of any off-the-shelf hardware that has audible sound capabilities. Localization is performed using variations of the established TOF technique that can be used by devices with no ultrasonic capabilities and with heterogeneous sound production characteristics. The lack of special hardware on the user end makes Beep comparable to sound source localization systems. Two of these systems have recently been proposed.

Bian et al. [1] describe a sound source localization system designed to infer communication activity between people. The system consists of 4 quads (sets of 4 microphones in a rectangle pattern), carefully placed to cover most of the house where it is deployed. The system first computes time delay estimation for each pair of microphones and then considers peak weight and applies steepest gradient descent method in searching for the location of sound source. Their approach to the source localization is similar to ours, the main difference being the estimation method, explained in the next section. They report an accuracy of 68cm 95% of the time, which is similar to our result for the interior points. The effect of noise is not reported.

Scott and Dragovic [13] report a system for locating finger clicks and hand clapping. Localization is performed by constructing a non-linear system of equations consisting of times-of-arrival, microphone locations, measurement error, time-of-send, and location of the sound source. The known variables are then substituted, and the unknowns (time-of-send and sound source location) are calculated using the Levenberg-Marquardt minimization method. This approach is conceptually similar to Active Bat's and to ours. The main difference is in the estimation method employed, explained in the next section. Their 3D finger clicking experiment resulted in an accuracy of 27cm 90% of the time. Although this result seems better than ours (40cm 90% for interior points), it must be assessed in the context of their experiment: their 3D results pertain to a 1.8mx1.8mx1.2m volume enclosed by 6 sensors and using all 6 sensor measurements, while ours (in 2D) pertain to a 20x9m room covered by 6 sensors and using only the measurements reported by the 3 closest sensors. Our results show that it is possible to obtain good accuracy with a much sparser sensor infrastructure.

Beep is similar to these systems in that it also performs sound source localization. However, the sensed sounds are considerably different. Human-made sounds can be considerably harder to process than the simple and highly controlled sounds we use in Beep. Therefore, the results reported for Beep can also serve as a baseline to in-door localization of natural sounds: with sufficiently accurate identification of those sounds (i.e. signal processing), the accuracy of localization should be similar to Beep's, and any degradation will be due to dealing with the signal's complexity.

III. Positioning with Audible Sound

One of the most widely used localization techniques is trilateration based on the Time-of-Flight (TOF) of a beacon signal sent from the device and detected by the sensors. In this scenario, the roaming device informs

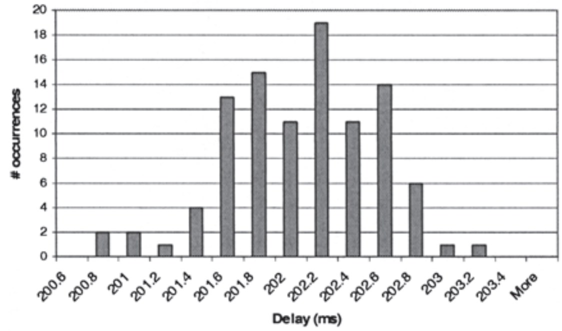

Figure 1: Distribution of the delay for the iPAQ H5550, with 100 measurements performed. Note that 100ms corresponds approximately to 34m.

the sensors to start a timer and immediately transmits a predefined acoustic signal, which is then detected by those sensors deployed at fixed points. Knowing the time-of-send, the sensors can infer the acoustic signal's time-of-flight after detecting the signal, which, multiplied by the speed of sound, corresponds to the roaming device's distance to each sensor.[1] Assuming ideal conditions where all n sensors report the correct distances D_i from the roaming device, the following spherical equations hold, and have a unique solution (x, y, z) in 3D:

$$(x - X_i)^2 + (y - Y_i)^2 + (z - Z_i)^2 = D_i^2 \tag{1}$$

where (X_i, Y_i, Z_i) are the coordinates of the ith sensor.

In general, the value of (x, y, z) can be derived uniquely from four sensor measurements, when the corresponding sensors have positions in 3D space. If the sensors are in a single plane (e.g. ceiling of a room) and the roaming devices are always located on one side of the plane, then (x, y, z) can be derived uniquely from three sensors in general position within the plane [11].

For devices with radio capabilities, then the "start timer" message can be sent by radio and, capitalizing on the speed difference of radio and sound, we can assume that such message is instantaneous and can accurately establish the time-of-send. For devices without radio capability, some other approach is necessary to establish the time-of-flight.

Unfortunately, the conditions we observed are far from ideal, and the direct application of this technique produces results that are severely flawed. Specifically, we observed that these devices have a significant delay between the time at which the microprocessor issues the command for the sound to be played and the time at which the sound is actually played. We also observed that the precise value of the delay varies considerably with the type of the device, and even with the device itself, and, for that reason, cannot be factored out easily. Figure 1 shows the distribution of measured delays for one of the devices we experimented with, an iPAQ H5550. The behavior of other types of devices is very similar, but the average delay has considerably different values: 250ms (approx. 86m) for the iPAQ H3670, 100ms (approx. 34m) for a Samsung i600 cell phone and 45ms (approx. 15.5m) for a Sony Vaio laptop. The delay has significant impact on the accuracy of trilateration: the perceived time-of-flight at the sensors is much larger than the real time-of-flight since it includes the unknown delay.

Under these conditions, and assuming that the sensors are placed on the same plane ($Z_i = 0$), the localization problem is the geometric problem of finding (x, y, z) satisfying:

$$(x - X_i)^2 + (y - Y_i)^2 + z^2 = (R_i - d)^2 \tag{2}$$

where d is the distance associated with the unknown delay, and R_i is the distance reported by the ith sensor.

The quadratic terms are identical in each equation and therefore can be eliminated by subtraction, leaving a linear system. The family of equations has four unknown variables and requires a minimum of four equations for a solution in the general case. It is sensible to use only the four equations associated with the four closest sensors since these are the most likely to be accurate and the most likely to correspond to straight propagation paths. Using more than four sensors could give better accuracy, at least in some situations, but would require a better model (e.g. probabilistic) to resolve possible inconsistencies. Thus, by renumbering the closest sensors from 0 to 3, the entire problem reduces to solving the linear system:

$$x(2X_i - 2X_0) + y(2Y_i - 2Y_0) + d(2R_0 - 2R_i) = R_0^2 - R_i^2 \tag{3}$$

for x, y, d with $i = 1, 2, 3$, and then deriving z as

$$z = \pm\sqrt{(R_0 - d)^2 - (x - X_0)^2 - (y - Y_0)^2} \tag{4}$$

and using the fact that all the sources are on one side of the plane containing the sensors to resolve the sign.

These equations are a variation of the TOF technique, the difference being that they include an extra

[1]For the purposes of this discussion, we assume that the speed of sound is constant, but, as noted in Section V that is not the case in practice.

delay/distance variable. In fact, these equations are well-known in GPS systems. In GPS terms, this delay/distance is accounted for in the distances reported by the GPS receivers, known as pseudoranges. The unknown delay/distance in the pseudoranges results from the difference between the time arbitrarily chosen by the receivers to start listening and the time at which the transmitter sends the signal. In our case the problem is further complicated by errors in the synchronization of the sensors and errors in the audio signal processing (more in Section IV.A). It is also worth noting that for many applications, the z coordinate is not important, only the x and y coordinates really matter, in which case three sensor measurements are sufficient.

Solving TOF in practice, however, is more difficult than what these equations suggest, because the physical system, which includes the sound production's and the sensing infrastructure's hardware and software, is not linear. Small deviations in what the theoretical R_i's should be can render the system of equations inconsistent. This problem occurs when implementing localization in practice, and estimation techniques, such as least squares, Kalman filter, or Levenberg-Marquardt (used in e.g. [13]) are usually employed.

Instead of using one of these well-known methods, we approximate the location (x, y, z) of the roaming device using a simple iterative algorithm that closely models the physical phenomenon at hand.[2] The key observation for our approach is that we know that the reported distances in each sensor are overestimated by an unknown, but relatively large, amount. Therefore, conceptually, we can formulate the location estimation problem in terms of overestimated spheres that gradually shrink by the same length until they stop intersecting. Figure 2 illustrates the idea in 2D. The estimation is done by repetitively shrinking by the same length the three circles (of radii $R_i, i = 1, 2, 3$) obtained from sensor measurements. The intersection area of the circles always contains the final solution (x, y). The iteration terminates when the intersection area becomes smaller than a desired threshold value. At this point the amount of shrinkage of the original circles gives the value of the delay d.

IV. Architectures and Protocols

Having established a simple location estimation method, we now focus on localization system implementations. We designed three architectures and corresponding protocols that use our location estimation method, but that have different properties. The following describes the commonalties among these architectures.

The positioning system consists of a set of acoustic sensors (S_i) that are connected to a central server through a wireless network. Each of these sensors has a microcontroller, a wireless network interface card and a microphone for detecting acoustic signals. When a user requests positioning, the user's roaming device transmits a pre-defined acoustic signal. The sensors detect this signal, and make an estimate of the time-of-flight. The time-of-flight is translated into distance by multiplying by the speed of sound. The distances are then reported to the localization server. The localization server, knowing the precise location (coordinates) of each of the sensors, performs trilateration to determine the coordinates of the user and reports the results to the roaming device.

Any implementation of this positioning system needs to include three operations: 1) sensing and detection of the audio signals; 2) identification of the roaming device requesting to be located; and 3) synchronization of the sensors.

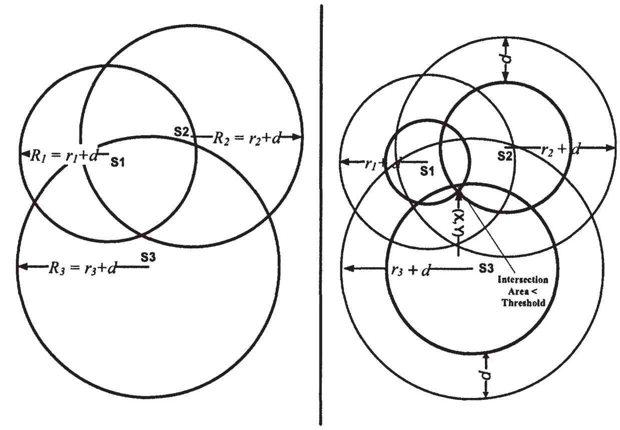

Figure 2: Distance circles around each sensor before shrinking (left) and after shrinking(right).

IV.A. WLAN/Sound (WLANBeep)

The first architecture, called WLANBeep, assumes that the roaming device is equipped with 802.11 wireless LAN (WLAN) technology. As such, the signaling between the device and the sensors includes both audio and 802.11 radio messages. Figure 3 shows this architecture. The components of this architecture include: 1) the wireless-enabled PDA as the user's roaming device; 2) the sensors; 3) the WLAN as the

[2] We are currently working on processing the experimental results using different estimation methods, in order to better understand which ones produce better results.

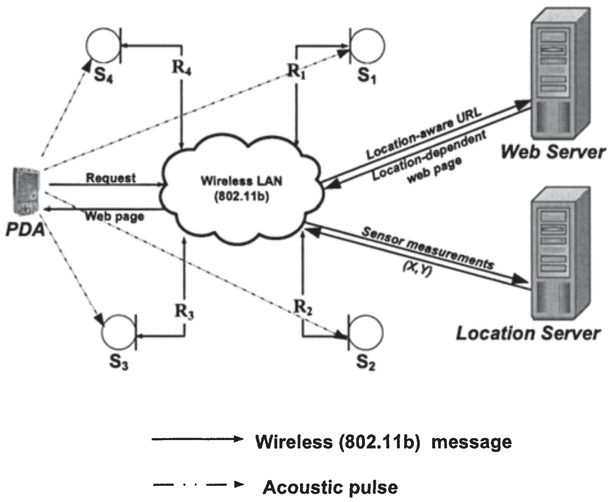

Figure 3: WLANBeep Architecture.

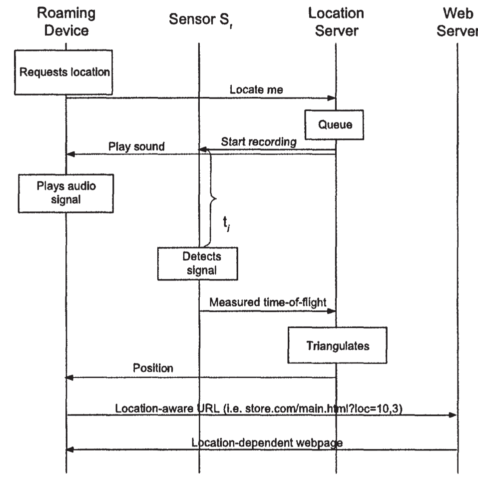

Figure 4: Sequence diagram showing the protocol used by WLANBeep and 3GBeep.

communication infrastructure among all components; 4) the localization server; and, eventually, 5) the web server, which can generate location-dependent web pages when it receives a location-aware URL from the roaming device.

We have designed one interaction protocol to be used with this architecture (see Figure 4). In this protocol, the roaming device first contacts the localization server to obtain the position, and then requests a web page from the web server, using a URL with the position embedded.

The device identification is done through its IP address, which the localization server receives on the first contact. Positioning requests from several roaming devices are queued by the localization server and served on a FIFO basis – i.e. no requests are lost.

In the absence of positioning requests, the sensors' audio recording is inactive, and only the radio listening is active. Once the localization server receives a positioning request, it broadcasts a message to all sensors instructing them to start recording audio. At the same time, the localization server transmits a message to the roaming device giving it permission to play the audio signal.

As mentioned in section III, for the trilateration algorithm to function properly, the sensors must be synchronized. In this protocol, synchronization among the sensors is done implicitly by the reception of the message to start recording. This approach is similar to that of other protocols that use receiver-to-receiver synchronization (e.g. [2]). In our case, this synchronization method incorporates an unavoidable error caused by the possibly variable time-of-travel of the radio message from the location server to the sensors. It also incorporates an additional source of errors: the measurement uncertainty of the audio signal processing at the sensors.[3]

The rest of the protocol includes the sensors detecting the audio signal and sending their time-of-flight measurements to the location server, which in turn performs the trilateration algorithm on the measurements and transmits the calculated position back to the roaming device. The obtained position is then embedded into a URL and sent to the web server, which generates a web page based on the user's profile and location. The protocol concludes when the web server sends the generated web page to the roaming device.

IV.B. Cellular Network/Sound (3GBeep)

The second architecture, called 3GBeep, assumes that the user's device is a third generation (3G) or GPRS (2.5G) mobile phone. For the sake of simplicity, we will generalize 3G to include GPRS during the rest of the paper. Figure 5 shows this second architecture. 3GBeep utilizes the 3G network, the Internet, and the set of components mentioned in WLANBeep's case. In this architecture the user's mobile phone must have access to the Internet through the wireless carrier's 3G network, hence making the localization server and the web server accessible to the mobile phone. The communication between the localization server and the set of sensors is carried out through the WLAN.

[3]We perform all signal processing in software, and on top of a non-real-time operating system (Windows).

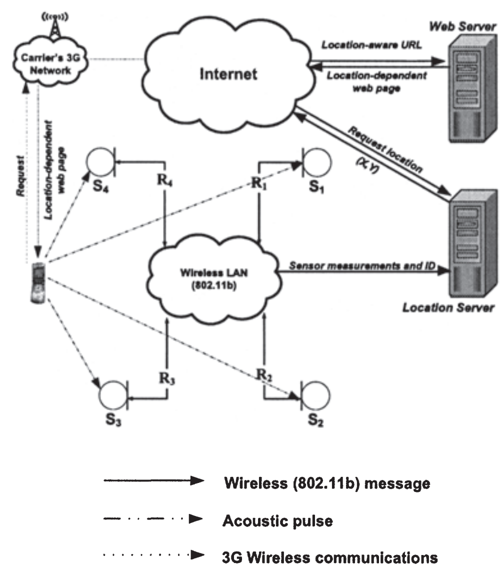

Figure 5: 3GBeep Architecture

The protocol among the different conponents is exactly the same as for WLANBeep (see Figure 4), the only difference being the network used by the roaming device.

IV.C. Sound Only (PureBeep)

The third architecture, called PureBeep, assumes no network connectivity on the part of the roaming device. As such, the signaling between the device and the sensors is done through audio only, and no data is passed back to the device. The goal here is to inform the system of device's location and not necessarily to send location-dependent information back to the device. Figure 6 shows this third architecture.

We have designed a second interaction protocol specifically for PureBeep. It is significantly different from the previous two protocols because of the absence of radio communication between the roaming device and the rest of the system. This constraint restrains us from being able to conduct device identification through radio communication. This problem can be resolved by encoding the device identifier in the audio signal itself. We used this technique in our prior work [8, 7], and others have explored it as well [9, 6].

Another issue affected by the roaming device's lack of radio communication capability is sensors' working periods and consequent power consumption. Unlike the previous protocol, here the system does not know when the device plays the audio signal. We addressed these issues using the protocol illustrated in

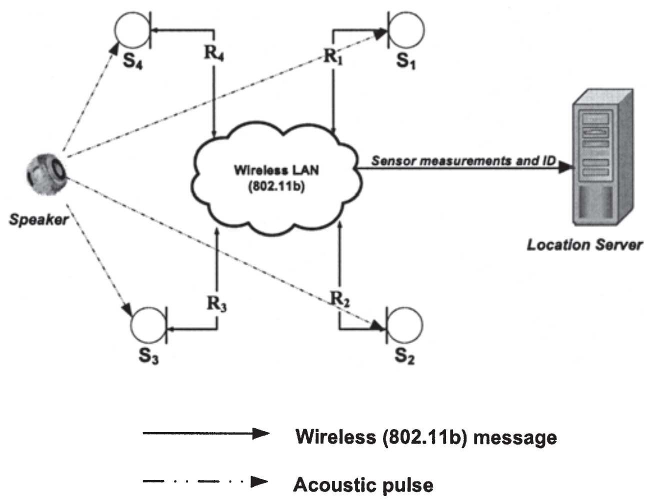

Figure 6: PureBeep Architecture.

Figure 7.

A subset of the sensors ("active sensors") continuously record audio, and trigger the localization process when they detect the acoustic signal. The signal played by the device consists of two segments: 1) the "sync" signal that triggers an active sensor, and 2) the encoded device identifier signal that is played with a precise delay after the "sync" signal in order to provide enough time for all sensors to start capturing audio. Once an active sensor detects the "sync" signal it notifies the localization server, which in turn broadcasts a message to all sensors in the region of that particular active sensor, including the active sensor itself.[4] The targeted sensors then start recording until they detect the second segment of the signal. The sensors subsequently notify the localization server of their measured time-of-flight and decoded device identifier.

This protocol may lead to the loss of positioning requests that are issued at positions nearby and at small time intervals from each other. Consider a scenario where a busy sensor detects a "sync" signal from a device, which results in all sensors in that region starting to record audio in search of the second acoustic signal from that device. Meanwhile if another device plays the "sync" signal, it goes unnoticed.

[4]We decided to make the active sensors contact the localization server instead of contacting the neighboring sensors directly, for a couple of reasons. First, we assume the sensors have no knowledge of the topology; second, in our implementation we wanted the active sensor to participate in the measurement. In order to do that accurately, that sensor needs to start recording at the same time as the others; that is more reliably achieved by the message from the localization server.

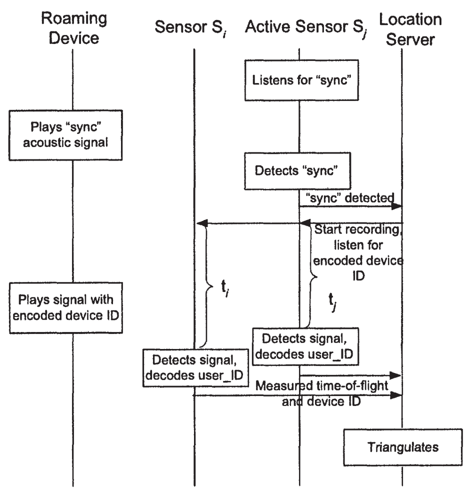

Figure 7: Sequence diagram showing the protocol for PureBeep.

V. Experimental Results

We have deployed the localization system, along with the three architectures described above, using a relatively large room, and assessed its performance under several conditions. This section describes the experimental setup and reports the results. Suplementary material (data, graphs, sounds and pictures) can be found on the Web site [14].

The deployment room was, roughly, a rectangle of 23m x 9m (see Figures 8 and 9). Six acoustic sensors were placed in pre-determined positions on the ceiling in a zig-zag shape.[5] Each sensor consisted of a Labtec Verse 333 PC microphone connected to a desktop. The localization server was a Pentium 4 PC, at 2.4GHz and with 256 MB of RAM. The trilateration was performed taking into account the 3 sensors that reported the closest distance to the sound source.

In this performance assessment of our approach to sound source localization, we are mainly interested in the accuracy of the (x, y) coordinates. Therefore the experiments reported here were conducted in 2D, using 3-sensor trilateration.[6] The test device was placed in the plane at about 1.2m from the floor, on top of a moving cart (so, the height z is fixed, not measured). We established 48 test points equally distributed throughout the room. The manual positioning of the roaming device introduced an unavoidable isotropic error of 6cm.

As for the roaming devices, we used an iPAQ HP 5550 for both WLANBeep and PureBeep, and an Audiovox SMT 5600 cell phone over Cingular's GSM/GPRS network for 3GBeep.[7] The signal was played at an intensity comparable to a cell phone ring.

The acoustic signal was a 4.01KHz tone with 0.2s duration; we used two of these for PureBeep, separated by 0.5s of silence. We chose this signal for two reasons. First, for testing purposes, and without loss of generality, we wanted the signal to be as simple as possible, so that we could employ computationally simple signal processing techniques – in this case, we used a Goertzel filter.[8]. Second, after performing a few tests with the devices, we concluded that their speakers work best within the 2-6 KHz range. The 0.2s of duration ensures a large audio data set for the signal processing code to work well without it becoming intrusive for human listeners.

Finally, each "test" at a given test point consisted of 100 localization requests at that test point.

V.A. Performance Under Quiet Conditions

WLANBeep (baseline experiment). Figure 8 shows the floorplan of the room overlayed with the up-to-scale bubble plot of the median error of this experiment at all test points (i.e. with 50% of the localization requests within those bubbles). The small dark dots on the plot show the directional bias of the errors. For these 4,800 localizaton requests, the system performed within 70cm of the theoretical points 90% of the time.

As the bubble plot shows, different test points have quite different results, the medians ranging from 5cm (at [10, 5.5]) to 74 cm (at [16, 7.5] and [22, 7.5]). The most severe errors occur at the edges of the room. This observation can be better quantified by comparing the histograms of the errors as shown in Figure 10. That figure shows the distribution of the error for a) all 48 test points in the room and b) the 20 interior points only.[9] Taking only the interior points into consideration, the accuracy of the system improves to localization within 40cm 90% of the time.

[5] There are reasons for using this configuration, but they fall out of the scope of this paper. The intuition is that we wanted a good coverage of the room with the smallest number of sensors.

[6] Introducing a third dimension is relatively straightforward, involving an additional sensor reading (4-sensor trilateration). For preliminary results of WLANBeep in 3D, we refer the reader to [10].

[7] In previous experiments in another test room, we used a Samsung SP-i600 on Sprint's CDMA network, which worked with comparable results.

[8] Note that we perform all signal processing in software, rather than in hardware.

[9] We call "interior points" to all the test points that are inside the area defined by the 2 lines of sensors.

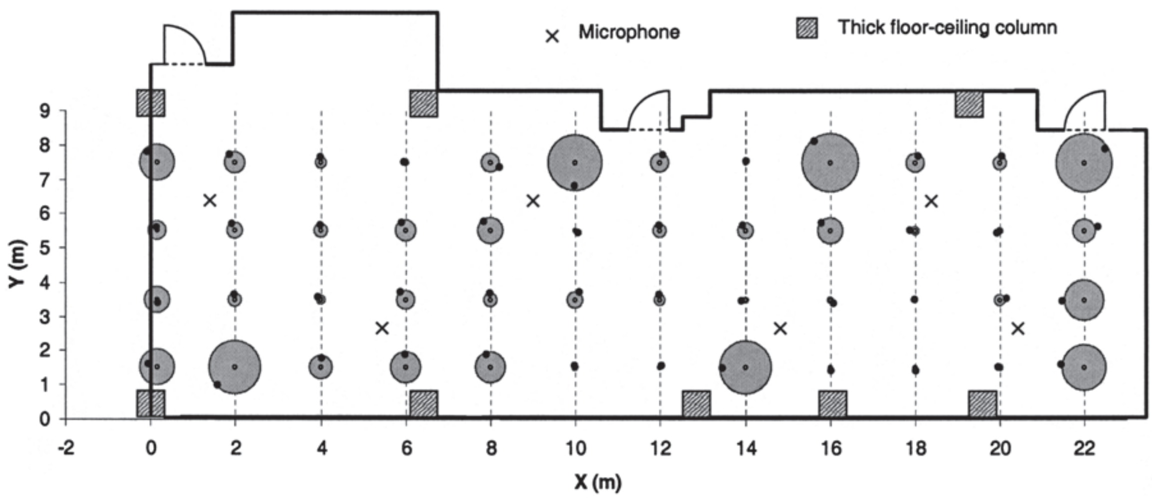

Figure 8: Floorplan of the test room overlayed with the bubble plot of the median error at each test point for WLANBeep under quiet conditions (baseline experiment). The circles, or bubbles, represent the median errors. The hollow dots at the center of the circles represent the ground truth, i.e. the test point; the darker dots represent the directional bias of the errors, i.e. the median of the measured x's and the median of the measured y's.

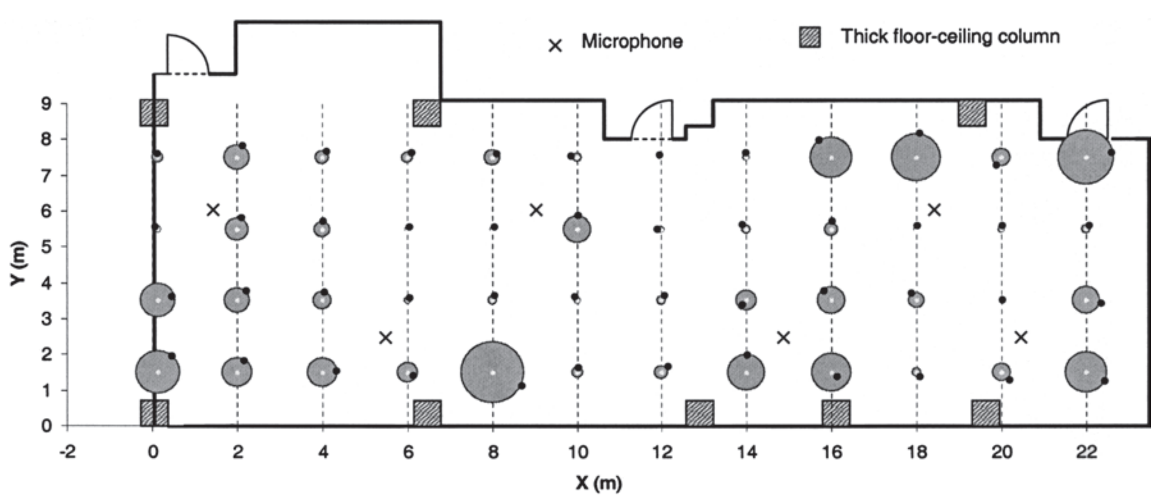

Figure 9: Floorplan of the test room overlayed with the bubble plot of the median error at each test point for PureBeep under quiet conditions. (See above for explanation of circles and dots).

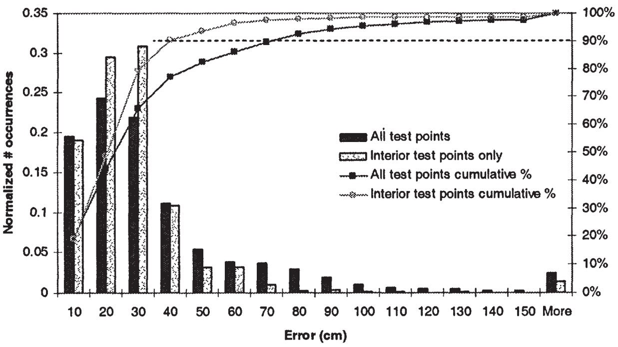

Figure 10: WLANBeep. Distribution of the errors for the combined 48 test points (dark bars and line) and for the 20 interior points only (light bars and line).

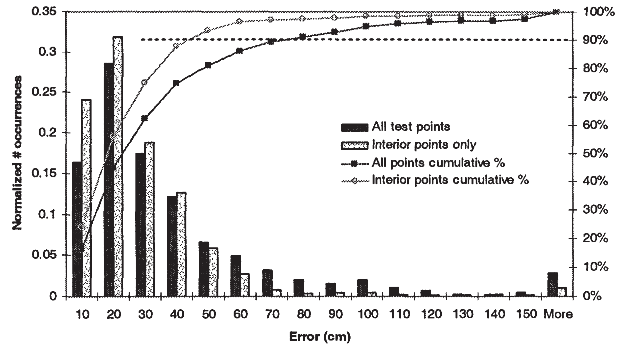

Figure 11: PureBeep. Distribution of the errors for the combined 48 test points (dark bars and line) and for the 20 interior points only (light bars and line).

The different results of the test points are due to factors of angular nature. The most dominant factor is the angle between the test point and the 3 pairwise combinations of the 3 closest sensors. For example, the test point [4, 5.5] establishes the following angles: α=140° (s1–point–s2); β=74° (s2–point–s3) and γ=146° (s3–point–s1). In the interior test points, these angles are relatively wide. However, for the points on the edges, one of these angles can be very small and even zero. For example, for the test point [0.15, 7.5] those angles are α=5°, β=35° and γ=30°. These angles play a crucial role in the accuracy of our trilateration method, as it is based on shrinking circles. A null angle implies handling two colinear circles that theoretically intersect at exactly one point (so, one circle is "inside" the other); in these circumstances, very small variations in each sensor measurement can result in wide variations in the measured distance. This angular sensitivity is the drawback of our simple TOF-based location estimation method, and it remains to be seen if other approaches will yield better results.

A potential second factor is the angular position of the microphones with respect to the test point. Other studies (e.g. [13]) have reported such a dependency. In our case, however, this does not occur. We monitored the reported distances from one of the sensors, while positioning it at several angles with respect to the test point, and there was no noticeable difference. Different microphones have different angular sensitivities; ours, placed on the ceiling, proved to be omnidirectional.

PureBeep. The method for detecting the acoustic signal in PureBeep is considerably different than the one used for WLANBeep and 3GBeep, since there is no radio signaling involved. Some preliminary tests showed PureBeep performed noticeably worse than WLANBeep. After improving the signal detection method, we extensively tested PureBeep in the 48 points of the test room. Figure 9 shows the bubble plot of the median errors in PureBeep and Figure 11 shows the histograms of those errors.

The results for PureBeep were similar to those obtained for WLANBeep. The medians are not exactly the same as those reported for WLANBeep at each test point. We attribute the differences to small setup errors that, especially at the boundaries, can have relatively large effects on the results. But on average, and similar to WLANBeep, the edge test points performed worse than the interior points. Considering all test points, localization is within 74cm 90% of the time (vs. 70cm 90% in WLANBeep); for the interior points only, localization is within 44cm 90% of the time (vs. 40cm 90% in WLANBeep). Again, these differences fall within the margin of error of the experimental setup.

3GBeep. A few preliminary tests indicated that the performance of 3GBeep was similar to the performance of WLANBeep, the major difference being the time that it took to serve each positioning request. While in WLANBeep the request/response was about 500ms, in 3GBeep that cycle took an average of 800ms, due to the utilization of the 3G network. In all other aspects, particularly with respect to the dependency on the angles, the behavior appeared identical. Based on those preliminary tests, the 3GBeep experiment included only 5 points (100 measurements each) that had proven to be relatively well-behaved for WLANBeep, so that we could study the effect of the 3GBeep architecture on the results.

Figure 12 shows the scatter plots of the results, along with the median errors at each point. These plots give a birds-eye view of the localization behav-

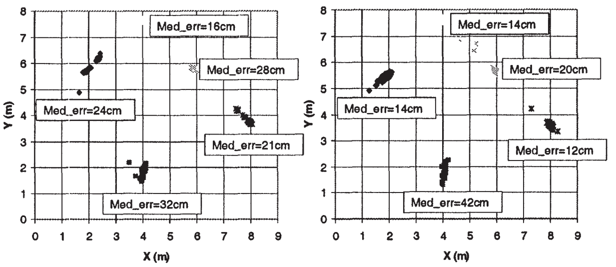

Figure 12: Scatter plot for the results on the 5 test points comparing WLANBeep (left) to 3GBeep (right). The test points were: [2, 5.5], [4, 1.5], [4, 7.5], [6, 5.5] and [8, 3.5].

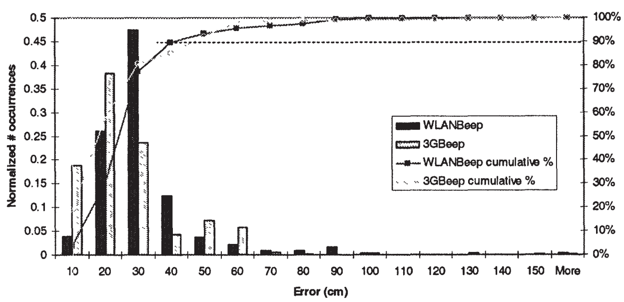

Figure 13: WLANBeep vs. 3GBeep. Distribution of the errors for the combination of the 5 test points shown in Figure 12.

ior for the two architectures. Figure 13 shows the histograms for the combined measurements. For these 5 points, 3GBeep tends to perform slightly better than WLANBeep in placing more results within 20cm of the theoretical points (57% in 3GBeep vs. 30% in WLANBeep), but the 90% mark is slightly worse (50cm in 3GBeep vs. 40cm in WLANBeep). In any case, the differences are within the margin of error of the experimental setup.

V.B. The Effect of Noise

The performance of any sound-based localization system is affected by the presence of other sounds – these other sounds are considered "noise." The effect depends on the kind of noise, on the kind of signal that is being detected and on the signal processing methods employed.

In a first experiment, we tested the effect of two different kinds of noise on the performance of WLANBeep, namely white noise and a 440Hz tone (central A). The following setup was used. We placed the roaming device at the test point [4, 5.5], which had shown a median error of 20cm in quiet conditions. Directly underneath the device, we placed a pair of desktop speakers connected to a computer. Those speakers produced the desired noise continuousely, at controlled intensities. As the noise was on, the device performed 100 localization requests. This setup established the same Signal-to-Noise Ratio (SNR) at the sensors, even though the sound intensities registered by the sensors were different. The results are shown in Figure 14. The intensity values (dB) shown in the figure are those reported by one of the sensors only (the one 3m away from the test point), and they serve to illustrate the ranges involved. Note that, at that sensor, -60 dB corresponded to our quiet conditions, and that the measured signal intensity was -40dB. The SNR, though, was the same at all the sensors.

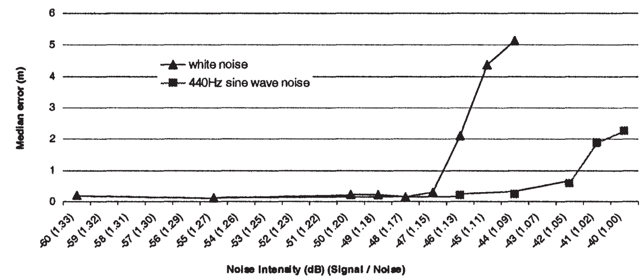

Figure 14: WLANBeep's performance in the presence of two kinds of noise. Intensity values (dB) reported by sensor 3m away from the test point.

As Figure 14 shows, the performance of the localization is unaffected by the increasing noise intensity, up to certain SNR thresholds, after which the presence of noise severely deteriorates the performance. As seen in the figure, the thresholds for the two kinds of noise are different. For white noise, the break point is around 13%, when the median error suddenly increases to over 2m. For the 440Hz tone, the degradation is more gentle, and the break point is around 2%, when the median increases to 2m. In other words, the figure shows that the performance of WLANBeep is reliable when the beep signal is at least 5% more intense than the single tone noise (at $SNR = 1.05$ the median error is still under 60cm); as for white noise, the performance of WLANBeep is reliable when the beep signal is at least 15% more intense than the white noise (at $SNR = 1.15$ the median error is under 30cm).

These differences are due to the choice of beep signal (a 4.01 KHz frequency), the signal processing method employed and the different interferences that the two noises have with the beep signal. Once a localization request is issued, each sensor proceeds to analyzing the sound stream to determine the exact be-

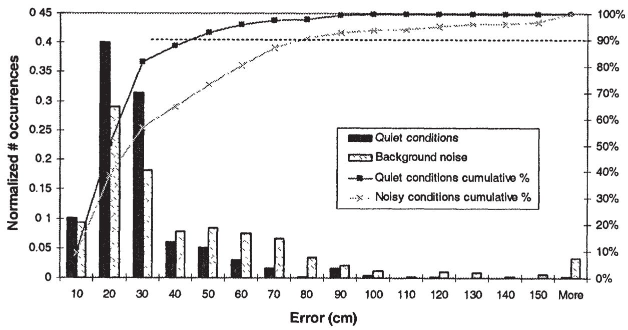

Figure 15: WLANBeep's performance in the presence of realistic noise.

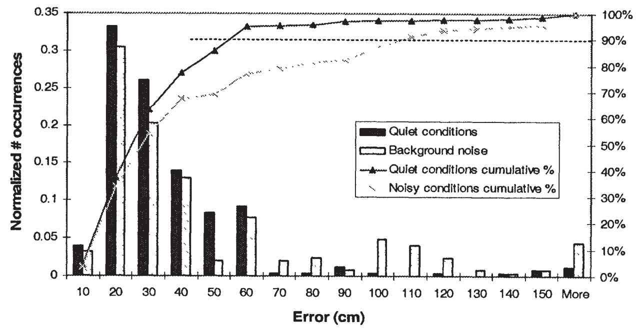

Figure 16: PureBeep's performance in the presence of realistic noise.

ginning of the signal. The presence of the beep frequency in the noise will interfere with that analysis. White noise contains all frequencies, including 4.01 KHz, therefore it has a relatively strong effect. But any sufficiently loud noise, even one that does not contain our frequency (such as 440Hz), will eventually disrupt the signal processing, because of scaling factors at the sound card and because of fixed thresholds used in the signal processing code.

Although these experiments give insights into the behavior of the localization system in the presence of noises with simple models, they fail to capture any realistic situation. Therefore we made a second experiment involving realistic noise. We tested the performance of WLANBeep as well as of PureBeep, since they involve considerably different signaling. In this experiment, the speakers were placed at the center of the triangle defined by three sensors, and localization was performed at 5 test points within that sensing area that had shown to be well-behaved.[10] The noise consisted of a superimposition of music and radio talk, and therefore its content was beyond our control. The intensity measured at the sensors was between -49dB and -43dB most of the time – we chose this range after having measured the intensity of ambient noise in a local supermarket. Figures 15 and 16 show the results for WLANBeep and for PureBeep, respectively.

As expected, the data shows a degradation of the results when noise is present. But, contrary to what the previous noise experiments might suggest, the degradation is graceful. For WLANBeep, and for these 5 test points, 93% of the time the error is within 50cm for quiet conditions and 80cm for noisy conditions; or, seen in another way, with noise, only 73% of the localization requests fall within 50cm of the theoretical test point (vs. 93% in quiet conditions). For PureBeep, and for these 5 test points, 86% of the time the error is within 50cm for quiet conditions and 100cm for noisy conditions. With noise, only 69% of the localization requests fall within 50cm of the theoretical test point (vs. 86% in quiet conditions). For the 1m range error, a reasonable value for many applications, WLANBeep degrades from 100% to 94% (i.e. 6% loss of accuracy) and PureBeep degrades from 97% to 87% (i.e. a 10% loss of accuracy).

As seen, the results for PureBeep in these 5 test points are slightly worse than WLANBeep's, but the values are close to the margin of error of the experimental setup. Nevertheless, the results suggest that the different signaling mechanism of PureBeep may play a role on its weaker tolerance to noise. In a PureBeep architecture, better signals and signal processing techniques are necessary to more accurately establish the acoustic synchronization message from the sound source to the sensors.

V.C. The Effect of Obstructions

In real deployments, we must consider situations when the source of the sound is obstructed. We tested our system for some of those situations.

One of the most likely obstructions is by the human him/herself: the person holding the beeping device may not be facing the sensors directly. In order to find out the effect of this configuration, we tested the system at different test points while placing the tester's body in the direct line between the device and one of the sensors. Table 1 shows the results for two of those tests; all other tests had similar results. This kind of obstruction seems to have no effect at all in the accuracy of the localization.

Finally, we tested the effect of a cement column as obstacle. For that, we placed the device in the corner of the room along x=6m, specifically in the two new

[10]Note that the SNR was different for the 3 sensors, and those differences varied from test point to test point. This variance was intentional, as it further simulated realistic conditions.

Table 1: The effect of obstructions made with the human body. (Error values in cm)

	[2, 5.5]		[4, 3.5]	
	Median	90%	Median	90%
Baseline	24	40	13	20
Obstruction	23	40	12	20

test points [6, 10] and [6, 9.5] – the latter being further inside the corner. Both of these had no direct line of sight to one of the sensors, and both of them are edge points.

Although there is no baseline data for these test points, because we couldn't remove the column, the results obtained are comparable to the results of edge points in general: the median errors were 52cm and 36cm, respectively; and the 90% mark was 70cm and 110cm, respectively. As baseline comparison, the test point [22, 7.5], with no obstructions, had a median error of 74cm and a 90% mark of more than 150cm.

In conclusion, these two types of obtructions don't seem to have any effect on the accuracy of the localization. Sound waves, unlike light, make rigid bodies vibrate, go around corners and fill the space as if there were no obstacles. In the presence of obstacles, especially when the material is such that it atenuates the sound considerably, errors may be introduced by the extra distance that the waves need to travel from the source to the sensor. In the cases we tested, the extra difference is negligible, maybe because the error of results (70cm 90%) hides the smaller errors introduced when the sound waves bend around these obstructions. Other cases may introduce more severe errors and should be carefully assessed.

V.D. Other Effects

During our experiments we observed that the temperature of the room affected the accuracy of the results. The speed of sound in air depends on the temperature in the following way:

$$v_{sound\_in\_air} \approx 331.4 + 0.6T$$

Differences of 10C, which are normal indoors, can have an observable impact on the results. We addressed this problem by making the speed of sound a parameter of our system, rather than a constant. In real deployments the room temperature should not be assumed as constant, but it could be automatically sent from a thermometer into the localization system.

VI. Conclusions

We presented a comprehensive exploration of the use of audible sound as localization medium for off-the-shelf mobile devices. The problem of positioning with audible sound was formulated given empirical observations related to the audio behavior of those devices. In particular, we have identified a source of uncertainty related to a sound production delay introduced by the device. We addressed this problem using a variation of the time-of-flight method that is similar to the technique used to factor out the pseudoranges' offset in GPS systems. We then proceeded to explore a number of architectures and corresponding protocols to localize and identify a source of audible sound. We have used as sources of sound Wi-Fi enabled PDAs, 3G mobile phones, and devices with no network connectivity.

In all three cases our experiments yielded results that were fairly accurate, namely within 70cm of the actual position 90% of time. Taking angular factors into account, the accuracy is 40cm 90% of the time. We studied the effect of different kinds of noise, namely white noise, a 440Hz tone and realistic background noise. As expected, noise degrades the accuracy. We have shown how the different kinds of noise have different effects on the accuracy. The presence of realistic noise results in an accuracy that is still be acceptable for many location-aware applications.

These results are promising, considering that the medium of localization is something as ubiquitously available as audible sound played by off-the-shelf mobile devices.

Even though the sound we used was played at a reasonable audio volume, comparable to that of cell phone rings, one concern with this approach for indoor positioning is the potential annoyance caused by audio transmission. One possible improvement is to make the sound more pleasing and less intrusive by disguising it as a cell phone ring. We have done some work along these lines [8] that can address this issue. The other possible improvement consists of making the system even more robust in noisy environments. This is yet another reason for moving away from single frequency sounds and adopt more interesting signals with specific patterns that survive collisions with other signals.

Acknowledgments. We thank Prof. Luke Bao and the staff at Calit2 for letting us use the test room during most of the summer of 2005. Thank you also to the anonymous and non-anonymous reviewers for helping us improve this work and this paper.

References

[1] X. Bian, G. Abowd and J. Rehg, "Using Sound Source Localization in a Home Environment," In Proceedings of PERVASIVE, Munich, Germany, May 2005, LNCS 3468.

[2] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," In Proceedings of The Fifth Symposium on Operating Systems Design and Implementation (OSDI), p. 147163, December 2002.

[3] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, (MOBICOM 99), pp. 59-68, 1999.

[4] A. Haghighat, C.V. Lopes, T. Givargis, and A. Mandal, "Location-Aware Web System," OOPSLA Workshop on Building Software for Pervasive Computing, 2004.

[5] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing " IEEE Computer, Vol.34, No.8, pp. 57-66, August 2001.

[6] S. Holm, O. Hovind, S. Rostad and R. Holm, "Indoors data communications using airborne ultrasound," In proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, 2005.

[7] R. Jurdak, C. V. Lopes and P. Baldi. "An Acoustic Identification Scheme for Location Systems," In Proceedings of IEEE/ACS International Conference on Pervasive Services, Beirut, Lebanon.

[8] C.V. Lopes and P. Aguiar, "Acoustic Modems for Ubiquitous Computing," In IEEE Pervasive Computing, Mobile and Ubiquitous Systems, Volume 2, No.3, pp 62-71. July-September 2003.

[9] A. Madhavapeddy, D. Scott, R. Sharp , "Context-Aware Computing with Sound," In Proceedings of The 5th International Conference on Ubiquitous Computing, October 2003.

[10] A. Mandal, C.V. Lopes, T. Givargis, A. Haghighat, R. Jurdak, P. Baldi, "Beep: 3D Indoor positioning using audible sound," In Proceedings of Consumer Communications and Networking Conference (CCNC), January 2005.

[11] D.E. Manolakis, "Efficient Solution and Performance Analysis of 3-D Position Estimation by Trilateration, " IEEE Trans. on Aerospace and Electronic Systems, Vol. 32, No. 4, October 1996, pp. 1239-1248.

[12] N.B. Priyantha, A. Chakraborty, H. Balakrishnan, "The Cricket Location-Support system," In Proceedings of the 6th ACM MOBICOM, August 2000.

[13] J. Scott and B. Dragovic, "Audio Location: Accurate Low-Cost Location Sensing," In Proceedings of PERVASIVE, Munich, Germany, May 2005, LNCS 3468.

[14] WebBeep Web pages. www.ics.uci.edu/ lopes/beep