Non-audible Acoustic Communication and its Application in Indoor Location-based Services

Kashif Ali
EECS, University of California Berkeley, USA
kashif@cs.berkeley.edu

Tayyab Javed, Hossam S. Hassanein and Sharief M.A. Oteafy
School of Computing, Queen's University, Canada
{tayyab, hossam, oteafy}@cs.queensu.ca

Abstract—**Location-Based services are gaining momentum as an important advancement in context aware services. That is, empowering users to identify potential services in their current space, and the prospect for services that are able to target local users, are pushing interest in research and industry alike. This paper explores the use of non-audible sound as a communication medium to tag and access location based services and gain access to their pertinent information. We propose and demonstrate the indoor implementation of a prototype of a location-based service-enabling system for hand-held devices. The system allows users to use their hand-held devices to search and interact with available services in their surroundings. A beacon placed in the service location broadcasts a service code mappable to the services particular to that location, and encoded via an ultrasound signal. The hand-held device can then identify that signal and prompt the user with the available services. We detail the novel system design and the ensuing architecture, and demonstrate the viability of the system which is tested over a variety of environments and scenarios. We conclude with an overview of the wide range of applications of this system, and note how it can enhance the way clients access location based services.**

I. INTRODUCTION

Location is an essential aspect of how people relate, organize and communicate with the physical environment. Wireless technologies, the Internet, navigational systems and geographical information systems assist us in meeting our location needs. Location is, in general, linked and related with other information systems to create and provide value added services, a paradigm known as Location-based Services (LBS) [1]. LBS comprise of user devices, position technologies, wireless network and application servers. All these components communicate together to enable services and enhance mobile user experience. Examples of LBS include outdoor applications, such as congestion-free routing between destinations, and indoor applications, such as collecting socially-ranked items on a menu at your favourite restaurant.

Positional technology is an important and enabling component of any location-based services. Numerous technologies have been developed and utilized to fulfil the positioning needs for both outdoor and indoor LBS. Examples for outdoor positional technologies are Global Positioning System (GPS) [2], assisted GPS, enhanced observed timed distance [1], cell global identity in cellular system [3], and time of arrival for the radio signal. Examples of indoor positional technologies are WiFi [4], Zigbee, Bluetooth [5], Radio Frequency Identification (RFID) and Ultrasound [6].

Various LBS applications requirements demand use of different positional technology. The requirements are defined using a wide range of parameters such as positional accuracy, power consumption, indoor/outdoor, latency, bandwidth, interference and operational scale. The factors play a major role in the adoption (or lack thereof) of LBS', especially in indoor environments where the abundance of devices coupled with lack of GPS signals cause poor localization. Although much research has been directed in the realm of LBS, most of the current approaches are hindered by factors pertaining to accuracy, battery consumption, cost of such systems, supported data rates, lack of encryption, frequency overlapping, indoor vs outdoor performance discrepancy, access latency, possible interference, range and scalability, in addition to poor traction and adoption by current LBS applications.

In this paper we propose a new paradigm for indoor location services, which capitalizes on non-audible acoustic communication. The reason acoustic waves are chosen as the communication channel is that to detect and identify sound waves we do not need any extra hardware on the client side, as hand-held devices already have built-in microphones. Also, sound waves do not penetrate through solid walls and are contained in a closed area making them a good candidate to be used as a location identifier. Our goal for a nominal LBS enabler is to adopt a paradigm that scales with the abundance of resources already with today's users.

In the remainder of this paper we argue for the viability of the non-audible based LBS system that we developed. In Section II we detail the design goals of our LBS system, and its components, along with the Encoding algorithm. Section III details the implementation of (Non-Audible System) NAS-LBS, and the architectural design challenges that entailed its design. Section IV highlights the thorough performance evaluation carried out, and our conclusions are presented in Section V.

II. NAS-LBS: NON-AUDIBLE SOUND-LOCATION BASED SYSTEM

The focus of our work is to propose and evaluate an efficient and ubiquitous location-based system for mobile devices. Efficiency is defined in terms of time to discovery with minimal system resources. Our objective is to enable smart spaces and associated services for all existing, and potentially

978-1-4673-9814-5/16/$31.00 ©2016 IEEE

futuristic mobile devices with minimal functional or operational interference. Our novel system, namely NAS-LBS, is an acoustic based location system which utilizes the non-audible acoustic channel to create a new channel of communication between the service providers and the clients. It has four major components; Audio Beacon Generator (ABG), mobile device application, back-end server and the controller. The remainder of this section details our motivation and design of NAS-LBS, and highlights how the system works on any mobile device operating system.

A. *Motivation and Design Objectives*

Smart spaces are geographical locations that have the ability to sense and react to people and devices present in it. A variety of sensing devices work together to collect data and provide various services to its users. To enable clients to discover a service, service providers must have some means to communicate with clients. To make sure every client can easily discover the service, it is advantageous for service providers to use the communication technologies that are highly accessible to the masses. Most mobile phones have built-in microphones. If such microphones can be used to receive certain information sent for the clients by the service provider, we have our highly accessible communication channel.

The communication should be inaudible to humans and only mobile phones shall be able to identify it. To achieve this, a device can be used to produce sound waves inaudible to humans but can be detected by the mobile phone's microphones. This is how our system enables its clients to detect services in their surroundings. A broadcasting device broadcasts a service code encoded in inaudible sound, particular to a location based service, which a mobile phone receives via its microphone, decodes the inaudible sound through signal processing and maps it to a service using a service mapping algorithm. Once the interface on the mobile phone prompts the user with the available services, a user can either use or ignore the services. For this system, for every area where a service provider wants to provide its services to its possible clients, a unique service code is generated. The service area is then recognized by the client application by that service code.

1) *Design Goals:*

- The system should realize the concepts of ubiquitous computing and can be installed easily in indoor locations. System should interface with mobile phones and integrate the information of the surrounding environment with the mobile interface in a seamless fashion.
- System should incorporate service discovery system that provides the user with the available services in real-time should integrate with already setup user interfaces so that service providers do not have to follow new protocols.
- System should use minimum resources of the client device to operate.
- System should be able to efficiently differentiate amongst different spaces.

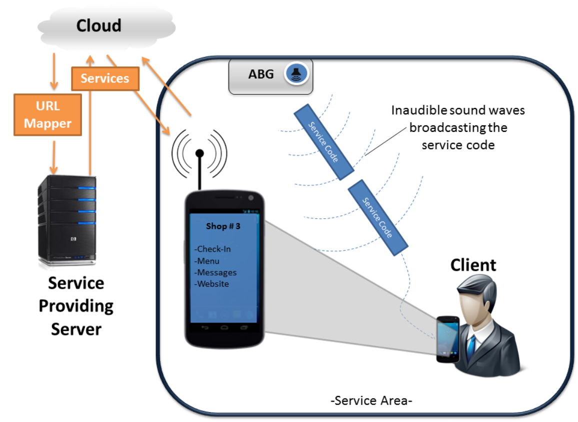

Fig. 1. System Architecture and Communication

B. *System Architecture*

There are four main components of our system. The remainder of this section details their functionality, and Figure 1 shows their interaction.

Audio Beacon Generator: The ABG acts as an entity that enables a process of discovery for a service in an area. It broadcasts the service code of that service encoded into an audio signal via a speaker that has frequency response up to 21 kHz. A user can configure and operate the device with the software provided with the system. The device can also be manually configured, customized and operated. We call this device ABG.

Controller: Controller is a background service that operates inside the ABG as a software and it controls the behaviour of the ABG. Its main responsibilities include listening for messages over network for system setup and operation, initiating the broadcasting of the ABG, sending messages to other ABGs notifying them of their broadcasting turn (in case of multiple-ABGs in one location).

Mobile device: Clients searching for new services around them interface with the system via a device that enables them to discover and interact with the services. A client device listens to the broadcasted audio signals in its surroundings via its integrated microphone, decodes the audio signal into the service ID/URL and allows the user to access the services via the Internet. It can be a mobile phone, laptop or another device. We have implemented our client side on Android operating system for now.

Service providers: Service provider in our system is basically a web-server that provides some location based services to its users. The URL of this website is encoded inside the acoustic signal broadcasted by the ABG. This acoustic signal is then broadcasted by the broadcasting device so that any clients in the surrounding area can decode it and access the encoded web-service as shown in Figure 1.

Back-end server: As different types of services have different interfaces, it is important to let the service providers be in the control of how the user interacts with the services. In our system the service provider enables the user to interface with the services via HTML documents running on a web server. Since all the existing mobile phones are able to browse HTML documents, users can directly interface with the service provider via mobile phones when the URL of that website is communicated to the mobile. The service provider can design the interface in any way they want with an option to design mobile based web interface using tools like flash to provide easy interfaces. The URL of this website is shortened using TinyURL which converts it into a short string of approximately seven characters. This string is encoded to an audio signal by the ABG as configured by the ABG Configuration Console. The client application on decoding the URL launches the service website.

Our backend server includes features like Service Address, Location Message and Facebook Check-in. Location message is a message for the place where the ABG is placed. Users can also post messages for each other on that particular location via the webiste. The messages are stored in a SQL database. A simple php interface was developed to blend the service interface with the mobile operating system.

C. NAC Algorithm: Non-audible Acoustic Communication

Sound waves exhibits behaviours like reflection, absorption, refraction, diffusion and diffraction. In a closed space, these collectively cause a phenomenon commonly known as reverberation. In reverberation, the sound waves remain persistent in an indoor environment for some time. This depends on the environmental composition and time it takes for sound waves energy to become negligible. Our encoding and decoding algorithms are designed to overcome such reverberation.

1) Encoding Algorithm: Encoding of the service code in the ultra-sound audio signal is achieved by custom version of Four Frequency Shift Keying (4-FSK) after applying a Base64 compression to the string. The whole encoding process has two steps. Base64 is an encoding scheme in which each character is represented by 6 bits instead of 8-bits. As in our case the string to be encoded is always textual data, Base64 compression is viable. This allows us to compress the original string by 25 percent. The Base64 compression algorithm also converts the compressed string into a stream of bits so that they can be passed to the 4-FSK encoding algorithm for conversion into an audio signal.

We use a custom version of four frequency shift keying encoding scheme with two synchronizing frequencies. In this encoding scheme four unique frequencies carry the actual data while two frequencies are used to indicate the start and/or end of the data. The algorithm reads a pack of two bits, turn by turn, from the bit stream provided by the Base64 compressor and generates an audio file with packets of sinusoidal waves individually comprising of different frequencies depending on the data encoded. The audio file generated has bit depth of 16 bits and has a sampling rate of 48 kHz.

The generated audio files encodes the data in form of audio packets, each carrying information of two bits of the original data. A data packet carries three different frequencies; synchronization frequency, silence frequency, and data frequency and is divided into four sections. The synchronization frequency in the audio packet signifies that a new packet has begun and that upcoming acoustic signals will be a part of it until another synchronization frequency is received. The silent signal is introduced in between synchronization frequencies and data frequencies to make it possible to distinguish them separately. This causes the client device to detect the synchronization frequencies and the data frequencies at the same time with modified amplitudes resulting in wrong detection of the broadcasted signals. There are four data frequencies in 4-FSK encoding scheme, ranging from 20.00 kHz to 20.60 kHz. After the data frequency a silence is generated so that in reverberating environments two audio packets do not bleed into each other. The data packet is arranged in a sequence of synchronization frequency, silence frequency, data frequency, and finally silence frequency. To optimize length of the audio signal and resolve errors in the communication, lengths of each of the parts of the packets can be modified as they greatly affect the operation of the algorithm.

2) Decoding Algorithm: The decoding algorithm receives a stream of audio buffers equivalent to a single audio packet received by the microphone. It decodes each buffer, by processing the values of amplitudes of the received coding frequencies using a Fourier transform. Signals are then converted into audio data packets and audio data packets to a stream of bits. The algorithm continues to pass audio buffers to the decoding function until synchronization frequency in any of the buffers is identified which indicates start of a new packet. The algorithm then waits for the time equivalent to length of Silence 1 and starts accumulating the amplitudes of each data frequencies as received in each buffer. The algorithm then waits for the time equivalent to Length of the data signal and starts looking for the other synchronization frequency. As it receives the buffer with the traces of the other synchronization frequency, it completes the current audio packet. The data frequency with the biggest sum of the amplitudes is considered to be the data frequency for the current packet and its corresponding bits set is added to the decoded stream of bits. The pseudocode of the decoding algorithm, divided into different sections based on the task, is shown in Listings 1. After every service code a silence of 200 ms is introduced. This helps the client to differentiate between two different broadcasts. When enough number of bits are decoded, they are sent through the Base64 extractor to get the original service code.

D. System Execution Process

Setup: In the first phase ABGs are placed in the desired locations depending on the requirements of the service provider. If a room contains one service only, then one ABG is required. If the room is divided into section, one ABG is placed in each section of the room. When ABGs are powered up they start listening for messages over TCP connections. An operator can configure them via interfacing with the ABG Configuration Console. A user can connect to the network of the ABGs through his local machine. Then via ABG

Algorithm 1 Pseudo code of Audio Streamer

INPUT: main audio buffer
OUTPUT: buffer
while size of processed buffer $<$ size of buffer for 10 seconds **do**
  Buffer = read 441 samples from main audio buffer
  Decoder (buffer)
  size of processed buffer = size of processed buffer + processed buffer
end while

Configuration Console a user can create a 3D model of the room and place the ABGs in that model as in the actual room.

Broadcasting: When all the ABGs in the room are in the listening phase, an operator can connect to the network of the ABGs through his local machine and send the start message to any of the ABGs via the ABG Configuration Console application. This will command the ABG to broadcast its service code once and send a start message to the next ABG as setup by the operator. If there is only one ABG in the area it will just broadcast the service code. This process will continue itself until a user sends a stop signal to the ABGs. **Reconfiguration:** If the service provider wants to update the service code, the ABG Configuration Console can be launched to load the map file of the room. Then the user can click on the ABG he wants to reconfigure and update the operating parameters. Then the user can send the update settings message to the ABG so that the ABG can update its settings. It is not required for a ABG to be first reset and then configured. This can be done while the ABGs is in the process of broadcasting.

III. NAS-LBS Implementation Prototype

A. Audio Beacon Generator (ABG)

ABG is implemented with a low-power open source hardware single-board computer BealgeBoard [7] connected to a Speaker and a Wi-Fi Dongle. In this prototype of the system Beagleboard-xM C3 was used which carries 1 GHz ARM Cortex-A8 core processor and 512 MB RAM. The Bealge-Board runs a custom Ubuntu 11.4 operating system designed for the BealgeBoard. The system applications running in Ubuntu are programmed in C. X-Mini 2 Portable speakers with a frequency response up to 20 kHz were used for the audio playback. Belkin's WLAN dongle is used in the ABGs as the WLAN interface. Our server application that controls the behaviour of the ABG listens for TCP connections via wireless ad-Hoc network for configuration and operational messages on port 9931. Each ABG's IP address is manually configured so that they operate in the same subnet. This allows multiple ABGs to communicate with each other and the user interfacing with the ABG Configuration Console on a computer.

B. Mobile Client Application

We developed an Android application for the system prototype. The application required minimum of Android SDK API level 8 for the operation. The application provides the user with a interface that allows him to scan for location based services and on success shows him a list of scanned services. The application scans the services by listening to acoustic signals from the mobile microphone for 10 seconds and decoding the identifiable signals via real-time signal processing. On selecting one of the scanned services, the application launches a web browser that opens up the web site of the location based service. On demand, it records and processes the audio data for 10 seconds.

C. Server

The service providing server is implemented as a website to interface with users and offers a variety of services. The web content was designed in HTML and PHP.

IV. Performance Evaluation

To evaluate the performance of the system a variety of test cases were designed and considered. Our evaluation cases cover all the important operational aspects (time of discovery and acoustic communication errors in small rooms, big rooms and hallways) of the system. The test were carried out in our Queen's Telecommunication Research Lab. Different areas of the room were chosen in the room to scan the services and different matrices were noted with debugging functions integrated with the client application.

A. Configuration and setup

1) Single ABG: As there are two modes of operation for the ABGs, we evaluate both of them separately. For the stand alone operation mode of the ABG, we use three different room sizes. The system is evaluated in different environments is because the surroundings of mobile client affects the acoustic communication.

Case 1: Single ABG in One Room To evaluate the system in a normal sized office room single ABG is placed in one end of a small office and the mobile client scanned the services and measured the metrics after every 5 feet till mobile reaches the other end of the room. The maximum distance between two opposite wall was 10 feet so we measure the metrics for 3 spots in the room.

Case 2: Single ABG in a Large Room To evaluate the system in a big office a single ABG is placed at one side and the service scans are run at increment of 5 feet away from the ABG to the other end of the room. The debugging functions in the client application measured the matrices overtime the scan was run.

2) Multiple ABGs: In these tests multiple ABGs are placed in a variety of environments and configurations to evaluate the system performance in environments with more than one service.

Case 1: Three ABGs in Three Different Rooms This scenario evaluates the system in an environment where there are multiple isolated rooms. We consider three connected rooms, each with its own dedicated ABG configured to broadcast the service code of the service available in its room. The ABGs are configured to operate in individual mode. Values of different parameters are recorded on selected locations in the floor.

Case 2: Three ABGs in One Large Room We can easily isolate rooms and their respective services by installing one ABG in each room, but it is important to evaluate if the system can still work in a large room with different cubicles and section separated by small wooden walls or open spaces. As the sound waves reflect and diffract with the environment they can reach almost everywhere if there is a small gap to let the sound waves traverse through. For the system to work properly, it is important that the amplitude of the received signal on the client side is at least higher than a threshold value in the operating zone.

To carry out the test, three ABGs were placed in our lab which has three sections separated by two half high plastic and metal walls. Fourteen spots were chosen in the room to evaluate the matrices, in group mode.

B. Performance Metrics

Following are the performance metrics used to evaluate the performance of the system.

Time of discovery: For a location based service discovery system, the amount of time it takes to discover the service is an important metric. We measure this metric by measuring the time the mobile client application takes from the moment of request for scan from the user to the moment where at least one service is successfully identified and shown to the user. This includes all the overheads of the process like network communication delays.

Error Percentage: As our system is based on an acoustic medium for digital data communication, it can be error prone. If the service code broadcasted by the ABG is not identified in the first round, the application will process the next broadcast, taking more time in the discovery process. This makes error percentage an important matric for our system. We measure this by counting the number of correctly identifiable broadcast messages received on the client side as a function of distance.

Algorithmic Configurations: One audio data packet is divided into four sections (synchronization part, silent part 1, data part and silent part 2) and it carries information of two bits. The length of these individual signals directly affects the performance of the algorithm. We can represent one audio data packet as a-b-c-d where a is the length of synchronization signal, b is the length of silent signal 1, c in the length of data signal and represents the length of the silent signal 2. Considering this, we select the following audio data packet configurations to be evaluated. *1-1-1-1:* For this configuration length of each signal is equal to one signal which is 0.01 seconds for our algorithm. Considering the service code has a length of 7 6-bit characters, with this configuration the encoding algorithm generates the encoded audio sample of this service code of length of 0.84 seconds. This configuration supports faster time of discovery as it takes less time to record and decode a short acoustic signal but at the same time it is error prone in reverberating environments. *1-1-2-4:* In this configuration the data part is comprised of two signals which is equal to 0.02 seconds. The silent part 2 is comprised of four signals which is equal to 0.04 seconds. With this configuration the generated audio sample has length of 1.68 seconds. The extended data and silent 2 parts enhance the chances of the audio packet being detected correctly by the client application. *2-2-2-2:* In this configuration all the sections of the audio packet are two signals which is equal to 0.02 seconds each. With configuration the generated audio sample has length of 1.68 seconds. *3-3-3-3:* In this configuration all the sections of the audio packet are three signals which is equal to 0.03 seconds each. With configuration the generated audio sample has length of 2.52 seconds.

V. System Evaluation Results

To evaluate the system, different scenarios and environments are considered so that the important parameters of the system can be evaluated. The tests are carried out using Samsung Galaxy S II running Android 3.4.4 as the client mobile device. We use 1-1-2-4 as our default data packet configuration. Test case 2 and 3 were repeated for each of the audio communication configurations though.

A. Single ABG

1) Case 1: Single ABG in One Room: The results of communication error metric show that the client application received all the broadcasted messages correctly. The reason there was no error in the communication is that small rooms do not have reverberation. With no reverb the client application can correctly identify short pulses of particular frequencies.

In this case the average time of discovery metric show that the overall average time of discovery noted was 3.26 seconds ranging from a minimum of 2.141 to a maximum of 4.406 seconds. This variation is due to communication from the ABG to the client device is a one way broadcast as the message is an inaudible audio signal played by the ABG repeatedly. If the user requests a service scan during a broadcast, that information is neglected and the client application waits for and decodes the next broadcast.

2) Case 2: Single ABG in a Large Room: The client device received almost all the broadcasted messages correctly on all the spots. Considering a small room an ideal environment for the system, the average time of discovery recorded was 3.3 seconds ranging from a minimum of 2.136 seconds to a maximum of 5.002 seconds.

B. Multiple ABGs

1) Case 1: Three ABGs in Three Different Rooms: The results of communication errors matric show that for most of the cases there was no error. The client device discovered and identified the correct ABG every time as the rooms were isolated from each other by walls. This implies that the ultrasound signals can be used to isolate and identify different enclosed areas. The measured average time of discovery has a minimum of 2.14 seconds to a maximum of 4.94 seconds.

2) Case 2: Three ABGs in One Large Room: The results of communication errors matric shows that all the messages are received correctly by the client device from all the spots. The reason is the time division multiplexing nature of the broadcasting. Average time of discovery varies between 2 seconds to 9 seconds with an average of 5.3 seconds. The reason for this wide range is that each ABG broadcasts its service code one after the other and client has to wait till the ABG mapped to his location gets its turn to broadcast the service code. In our tests, each broadcast takes 1.68 seconds and the time delay after every broadcast is 200 milliseconds, then three different broadcasts take an average of (1.68 + 200) * 3 = 5.64 seconds (ignoring the TCP communication delay).

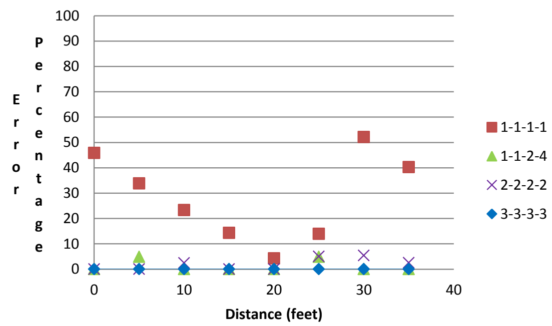

Fig. 2. Communication Error for the room scenario for algorithm test cases

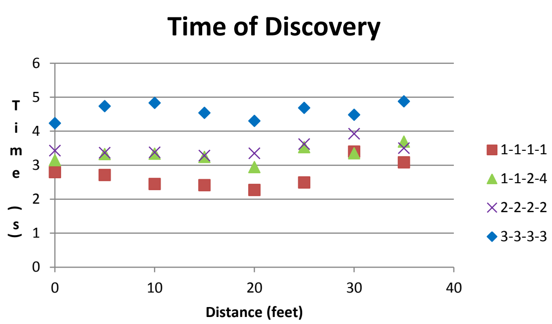

Fig. 3. Average Time of Discovery for the room scenario for algorithm test cases

Hallway: Figure 2 and Figure 3 show that in hallways and large rooms with lots of reverberation, 1-1-2-4 is the best audio data packet configuration as it has them shortest times of discovery with average of 3.136 seconds. 1-1-1-1 was not able to work properly as most of the messages were not received properly. 1-1-2-4 also carries least number of corrupted messages supporting lesser values of time of discovery.

VI. Conclusion

This work explores the use of ultrasound to create an easily accessible communication channel that can be used to help

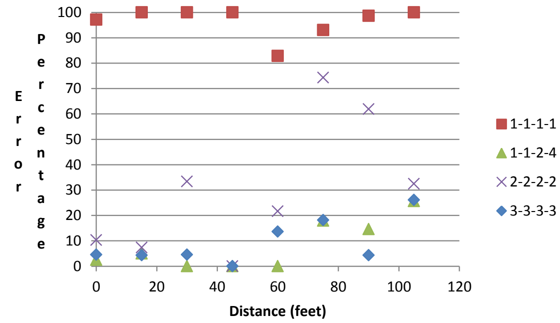

Fig. 4. Communication Error for the hallway scenario for algorithm test cases

the user in discovering and searching location based services in their environments. A design of a working prototype of such a system was demonstrated and implemented. Different operational modes of the system were considered to support multiple services in multiple subareas in a single service area. A hardware module for the system named ABG is designed to encode and broadcast digital data as non-audible sound signals. These modules were designed to be configurable via network. A client application was designed for Android operating system as was tested on multiple devices to assess the system compatibility. It was found that all of the modern hand-held devices are compatible with the system. The client application has ability receive digital data via ultrasound received from it's microphone. A 3D map builder and configuration application was developed for the system. Multiple test scenarios are discussed and test results show that the system can be used publicly for service discovery and accessing location based services more efficiently.

References

[1] Y. Gu, A. Lo, and I. G. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys and Tutorials*, pp. –1–1, 2009.
[2] A. Schmidt, B. Pfleging, F. Alt, A. Shirazi, and G. Fitzpatrick, "Interacting with 21st-century computers," *IEEE Pervasive Computing*, pp. 22–31, 2012.
[3] J. Machaj, "Impact of wi-fi access points on performance of rbf localization algorithm," in *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*. ELEKTRO, 2012, pp. 70–74.
[4] S. Ishida, K. Tomishige, A. Izumi, S. Tagashira, Y. Arakawa, and A. Fukuda, "Implementation of on-demand indoor location-based service using ad hoc wireless positioning network," in *Proceedings of the IEEE 14th Intl Conf on Scalable Computing and Communications (UIC-ATC-ScalCom)*. IEEE Computer Society, 2014, pp. 34–41.
[5] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, ser. Mobicom '12. New York, NY, USA: ACM, 2012, pp. 269–280. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348578
[6] T. V. Haute, E. D. Poorter, F. Lemic, V. Handziski, T. Voigt, A. Wolisz, I. Moerman *et al.*, "Platform for benchmarking of rf-based indoor localization solutions," *Communications Magazine, IEEE*, vol. 53, no. 9, pp. 126–133, 2015.
[7] "Beagleboard-xm rev c," http://beagleboard.org/static/BB-xM$_{R}EV_{C} - 2011 - 05 - 23.zip$, 2011.