International Journal of Computer Architecture and Mobility (ISSN 2319-9229)
Volume 7 -Issue 1, January 2019

A Sonar Ranging Sensor for Smart Phones

Chinmay Rangnekar
Department of Computer Science and Engineering
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore, Madhya Pradesh, India
chinmayrangnekar27@gmail.com

Saloni Trivedi
Department of Computer Science and Engineering
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore, Madhya Pradesh, India
salonitrivedi1599@gmail.com

Aradhya Jain
Department of Computer Science and Engineering
Shri Vaishnav Vidyapeeth Vishwavidyalaya
Indore, Madhya Pradesh, India
aradhya299@gmail.com

***Abstract*—Visually impaired persons find themselves challenging to grow out independently. The main objective of our project is to help blind people to walk with ease and to be warned whenever their walking path is obstructed with other objects, people or other similar odds. As a warning signal, a buzzer or voice alert can be given to the person and frequency of beep changes according to the distance of object which is calculated by measuring the elapsed time between the initial pulse and its reflection. The technology proposed in our project does not require any external hardware component such as ultrasonic sensor, aurdino-microcontroller or any other device. We are implementing this by using the inbuilt speaker and head-mounted microphone in the mobile device which are working as a transmitter and receiver. By the use of our application the blind or visually impaired persons are somehow they will become self dependent. The app is using different technologies for improvising its accuracy, robustness and real time performance.**

I. INTRODUCTION

Sensors on mobile devices have allowed developers to create innovative mobile applications. For example, the use of GPS localization allows developers to create applications that tailor their content based on the user's location [1]. Since the release of Android 1.5, Google has added application program interface (API) support for eight new sensors [2]. These sensors include: ambient temperature sensors, ambient pressure sensors, humidity sensors, gravity sensors, linear acceleration sensors and gyroscopic sensors.

In this paper we explore the possibility of using sonar to provide depth sensing capabilities in both indoor and outdoor environments and address two unique research questions:

1) *How do we design a sonar sensor for smart phones using only the phone's existing hardware?*
2) *How do environmental factors such as noise, reverberation and temperature affect the sensor's accuracy?*

The proposed sonar sensor uses the smart phone's rear speaker and microphone, and implements the sonar capabilities on a software platform.

The software process consists of three major modules:

1) A signal generation,
2) A signal capture, and
3) A signal processing.

The sonar sensor can be evaluated using three metrics:

- accuracy,
- robustness,
- and real-time performance.

The accuracy of the sonar sensor can be evaluated by comparing the distances reported by the sensor with known distances. The sensor that is previously used can accurately measured distances within 12 centimeters. The robustness of the sensor can be evaluated by comparing the sensor's accuracy under different noise and reverberation conditions in different environments. Finally, the sensor's real time performance can be evaluated by measuring the time that it takes to process a signal and return a measurement when different optimizations are applied.

The main contributions of the paper are:

- Presents a design and implementation of a sonar sensor for smart phones that does not require specialized hardware.
- Uses the smart phone's temperature sensor to improve the accuracy of the readings.
- Evaluates the sonar sensor under different reverberation and temperature conditions.

II. BACKGROUND

A sonar system can be decomposed into three steps. Figure 1 shows a simulated example of these steps. During the first step, the system generates a pulse. This pulse travels through the air until it encounters an object. Once the pulse encounters an object, it is reflected by the object. These reflected waves then travel back to the system which records the reflected pulse. The time difference between the initial pulse and the reflected pulse is used to calculate the distance to the object. Since the speed of sound in air is known, the distance to an object can be calculated by multiplying the time difference between the initial pulse and the reflected pulse by the speed of sound, and dividing the result by two. We need to divide by two because the time difference between the reflected pulse and the initial pulse accounts for the time that it takes the wave to travel from the phone to the object and back.

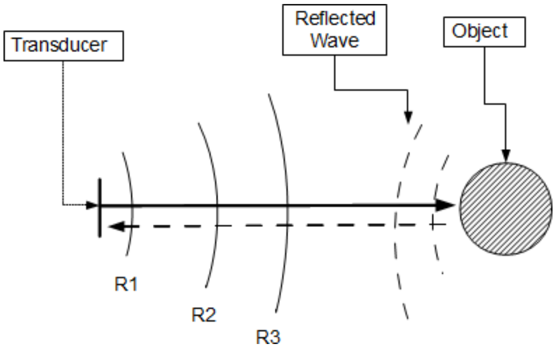

Fig. 1: This figure shows an overview of the process that the sonar system uses to calculate the distance from the system to an object. The reflected pulse will contain noise from the environment. This noise is reduced by filtering the signal.

As the wave travels further from the transmitter, its power density decreases. If an object is too far away, the energy density of the wave that encounters the object may not be enough to generate a reflected wave that can be picked up at the receiver. Distance is not the only factor in determining the amount of energy that is reflected. The amount of energy that is reflected is also determined by the composition and cross section of the object. Larger objects have larger cross sections and therefore reflect more energy, while smaller objects have smaller cross sections and therefore reflect less energy. Because objects with larger cross sections reflect more energy, they can be detected at larger distances. However, objects with smaller cross sections can only be detected at smaller distances because they reflect less energy.

III. RELATED WORK

In 1968 D. Dean wrote a paper entitled “Towards an air Sonar” in which he outlined some of the fundamental challenges of designing in-air sonar [3]. These challenges included acoustic mismatch and wind effects. Since Dean’s paper several in-air sonar systems, have been develop for a variety of applications. These systems include: ultrasonic imaging [4], ultrasonic ranging for robots [5] and SODAR (SOnic Detection And Ranging) systems that measure atmospheric conditions [6].However, all of these systems have been implemented using custom hardware. By using custom hardware these systems are able to address many of the challenges associated with in-air sonar systems. This is where our system is different. The sonar sensor that we proposed does not use any custom hardware and must compensate for the limitations of the commodity hardware in everyday smart phones.

The earliest occurrence of a smart phone based ranging sensor in the literature occurred in 2007 when Peng et al. proposed an acoustic ranging system for smart phones[7]. This ranging sensor allowed two smart phones to determine the distance between them by sending a collection of beeps. The sensor was accurate to within centimeters. The sensor is a software sensor and only uses the front speaker and microphone on the phone. Our sensor is different from the sensor in [7] because it allows smart phones to determine the distance from the phone to any arbitrary object in the environment.

In 2012, researchers at Carnegie Mellon University proposed a location sensor that allowed users to identify their specific location within a building [8]. The system proposed by the researchers used a collection of ultrasonic chirps that were emitted from a collection of speakers in a room. A smart phone would then listen for these chirps and use this information to locate a person in a room. The phone was able to do this by using the chirps from the speakers to triangulate itself. For example, if the smart phone is closer to one speaker than another it will receive that speaker’s chirp before it receives the chirp from another speaker. Since the locations of the speakers are known and the interval of the chirps are also known, the phone is able to use the chirps to triangulate its location. Our system is different from this one, since it attempts to determine the location of the smart phone relative to another -object.

Other researchers have also implemented in-air sonar systems on other unconventional systems. For example, researchers at Northwestern University have implemented a sonar system on a laptop [9]. Other researchers have also uploaded code to Matlab central that implements a sonar system on a laptop by using Matlab’s data acquisition framework [10]. The closest sensor to the proposed sensor is an Iphone application called sonar ruler [11]. The application measures distances using a series of clicks. The application does not filter the signal and requires the user to visually distinguish the pulse from the noise. Our sensor is different from the sonar ruler application because our sensor filters the signal and does not require the user to manually inspect the raw signal. Removing the need for user input allows the proposed sensor to be abstracted using an API. Being able to abstract the sensor using an API is important because it allows the sensor to be easily used by other applications.

IV. DESIGN

The system is comprised of three major components:

1) A signal generation component,

2) A signal capture component and

3) A signal processing component.

Figure 3 shows an overview of these components. The signal generation component is responsible for generating the encoded pulse. The signal capture component records the signal that is reflected from the object. The third component is the signal processing component. The signal processing component filters the signal and calculates the time between the initial pulse and its reflection. This component is comprised of two sub-components. The first component is the filtering component and the second sub- component is the peak detection component. We discuss each component in detail in the following sections.

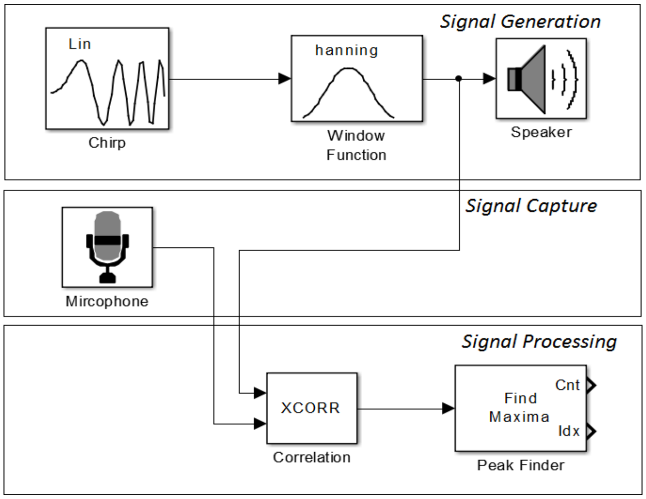

Fig. 3: The figure shows an overview of the sonar system's architecture.

A. Generating the Signal

The signal generation process is comprised of two sub processes. The first sub process generates an encoded pulse, while the second sub process shapes the encoded pulse. We discuss each part of the process in a separate subsection. We begin by discussing the pulse encoding process which is also called pulse compression.

B. Capturing the Signal

Once the system has transmitted the pulse, the next step is capturing the pulse's reflection. The signal capture component is responsible for capturing the signal's reflection. However, accurately capturing the signal possess two unique challenges. The first challenge is working with the constraints of the phone's sampling rate and the second challenge is concurrently managing the hardware's microphone and speaker buffers.

1) Sampling Constraints and Hardware Requirements:

The range of frequencies that can be recovered by the phone is limited by the maximum sampling rate and frequency response of the hardware. This is because in order to recover a wave we must sample at more than twice the wave's frequency. This means that the frequencies that can be contained in the linear chirp are limited by the sampling rate of the microphone and speaker.

The microphone and speaker on the nexus 4 has a maximum sampling rate of 44.1kHz. This means that without the use of compressive sensing techniques it is only possible to generate and record a maximum frequency of 22, 050Hz, since Nyquist sampling theorem states that we must sample at twice the frequency of signal that we are attempting to recover.

To ensure that we remain within the sample range of most phones, we use a linear chirp that ranges from 4kHz to 8kHz. Limiting the frequency range of the linear chirp allows us to address the sampling constraints of the hardware. In addition to the sampling constraints of the hardware, the phone's speaker and microphone have frequency response constraints. This means that they are only able to generate and receive a limited range of frequencies.

This frequency response depends heavily on the make and model of the microphone and speaker, which can vary drastically among devices. To mitigate this we select a frequency range for the chirp that is slightly above the range of the human voice. So most smart phone microphones and speakers should be able to transmit and receive the pulse.

2) The Concurrency Problem:

State of the art sonar systems have the ability to concurrently manage the microphone and speaker buffers. Synchronizing the buffers is important for sonar systems because ideally the system would start recording immediately after it has finished sending the pulse. Starting the recording immediately after the pulse is transmitted provides a baseline for calculating the time difference between when the initial pulse was sent and when the reflected pulse was received.

If the buffers are not well managed, the recording may contain the initial pulse, and the time index of the reflected pulse will not accurately represent the time between the initial pulse and the reflected pulse. The android operating system does not allow for real-time concurrent management of the microphone and speaker buffers so synchronizing them is challenging. This means that we must find a way to accurately calculate the time between the initial pulse and reflected pulse without managing the buffers in real-time.

We solve the concurrency problem by starting the recording before the pulse is transmitted. Starting the recording before transmitting the pulse allows us to record the initial pulse as well as the reflected pulse. We can then calculate the distance by calculating the time between the first pulse and the second pulse, since we have recorded both. This solution works because the proposed sonar system is mono static which means that both the microphone and the speaker are located on the same device.

C. *Processing the Signal*

Now that we have explained how the sonar system generates a pulse and captures its reflection, we can discuss how the captured signal is processed. The process of analyzing the signal is comprised of two sub processes. The first process is the filtering process. The signal is filtered by calculating the cross correlation between the known signal and the noisy signal. The result of the filtering process is passed to the peak detection process, which detects the peaks in the output and calculates the distance between each peak. The distance between peaks is used to calculate the distance between an object and the sonar system.

D. *Temperature's Impact on the Speed of Sound*

Environmental factors such as temperature, pressure and humidity affect the speed of sound in air. Because these factors affect the speed of sound in air, they also affect the accuracy of a sonar system. The factor that has the most significant impact is temperature [12] [13]. In this section, we show how the ambient temperature can significantly affect the accuracy of a sonar system. We also propose a method for increasing the system's accuracy by using the phone's temperature sensor to estimate the air's temperature.

Equation 6 from [14] describes the relationship between the speed of sound and the air's temperature. Where Tc represents 8 the air temperature and v(Tc) represents the speed of sound as a function of air temperature.

$$v(Tc) \approx 331.4 + 0.6 * Tc$$

From equation 6 we can see that underestimating the temperature will result in a speed of sound that is slower than its actual speed. Underestimating the speed of sound will cause objects to appear further away than they actually are, while overestimating the temperature will overestimate the speed of sound thus causing objects to appear closer than they actually are.

Overestimating or underestimating the temperature by a single degree results in an error of 0.6 meters for every second of time that has elapsed. Therefore we can improve the accuracy of the sonar sensor by using the phone's temperature sensor. Several phones such as the Samsung S4 now have ambient temperature sensors [15].

V. CONCLUSION AND FUTURE WORK

The proposed sonar sensor is comprised of three components: a signal generation component, a signal capture component and a signal processing component. Designing a sonar system for smart phones presented two unique challenges:

1) concurrently managing the buffers and
2) Achieving real time performance.

We addressed the concurrency problem by starting the recording before transmitting the pulse. This allowed us to capture the pulse along with its reflected pulses. Doing this allowed us to determine the index of the pulse and reflections by filtering the signal. We addressed the real-time performance problem by reducing the algorithmic complexity of the filtering process from $O(n^2)$ to a $O(n\log(n))$ by performing the cross-correlation calculation in the frequency domain. Finally, we evaluated our sonar sensor using three metrics: accuracy, robustness, and efficiency. We found that the system was able to accurately measure distances within 12 centimeters.

We evaluated the robustness of the sensor by using it to measure distances in environments with different levels of reverberation. We concluded that the system works well in environments that have low reverberation such as outdoor environments and large rooms but does not work well in areas that have high reverberation such as small rooms. In the future, we plan to investigate strategies for improving the sonar sensor's accuracy in environments with high reverberation. We also evaluated the system's real-time performance.

We found that by performing three optimizations we were able to reduce the computation from 27 seconds to under 2 seconds. In the future we will be releasing the sonar application on the android market. This will allow us to test the sensor's performance on different hardware platforms.

VI. REFERENCES

[1] J. Nord, K. Synnes, and P. Parnes, "An architecture for location aware applications," in System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on. IEEE, 2002, pp. 3805– 3810.

[2] "Sensors Overview," http://developer.android.com/guide/topics/sensors/ sensors overview.html, 2013, [Online; accessed 22-November-2013].

[3] D. Dean, "Towards an air sonar," Ultrasonics, vol. 6, no. 1, pp. 29–32, 1968.

[4] M. P. Hayes, "Ultrasonic imaging in air with a broadband inverse synthetic aperture sonar," 1997.

[5] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun, "Sonar-based mapping with mobile robots using em," in machine learninginternational workshop then conference-. morgan kaufmann publishers, inc., 1999, pp. 67–76.

[6] S. Bradley, "Use of coded waveforms for sodar systems," Meteorology and Atmospheric Physics, vol. 71, no. 1-2, pp. 15–23, 1999.

[7] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: a high accuracy acoustic ranging system using cots mobile devices," in Proceedings of the 5th international conference on Embedded networked sensor systems, ser. SenSys '07. New York, NY, USA: ACM, 2007, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1322263.1322265

[8] P. Lazik and A. Rowe, "Indoor pseudo-ranging of mobile devices using ultrasonic chirps," in Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems. ACM, 2012, pp. 99–112.

[9] S. P. Tarzia, R. P. Dick, P. A. Dinda, and G. Memik, "Sonar-based measurement of user presence and attention," in Proceedings of the 11th international conference on Ubiquitous computing. ACM, 2009, pp. 89–92.

[10] R. Bemis, "Sonar demo," http://www.mathworks.com/matlabcentral/ fileexchange/1731-sonar-demo, 2014.

[11] L. C. Corp, "Sonar ruler," https://itunes.apple.com/us/app/sonar-ruler/ id324621243?mt=8, 2014.

[12] G. S. Wong and T. F. Embleton, "Variation of the speed of sound in air with humidity and temperature," The Journal of the Acoustical Society of America, vol. 77, p. 1710, 1985.

[13] J. Minkoff, "Signals, noise, and active sensors-radar, sonar, laser radar," NASA STI/Recon Technical Report A, vol. 93, p. 17900, 1992.

[14] "Speed of sound in air," http://hyperphysics.phy-astr.gsu.edu/hbase/ sound/souspe.html, note = Accessed: 2014-12-05.

[15] "Samsung GALAXY S4 SPECS," http://www.samsung.com/ latin en/consumer/mobile-phones/mobile-phones/smartphone/ GT-I9500ZKLTPA-spec, 2013, [Online; accessed 27-January-2014].