Object Location In a Multiply Reflective Environment

(A Student Submission)

Brendan C. Walker, Hermann Chong, Tony Woo

Industrial Engineering
University of Washington
Box 352650
Seattle, WA 98195-2650
Phone: (206) 543-1541
Fax: (206) 685-3072
Email: twoo@u.washington.edu

Abstract

In this paper we present a computational method for determining a solution to the multi-path reflection problem, specifically in the audio domain. First we will explore the common need for such a solution. Then we will discuss prior attempts at solving the same problem to illustrate the novel approach of our method. From there, the complexities of the multi path and signal synchronization problems are explained and a potential solution is proposed. Specifically, if one assumes that the reflective geometry in the environment is known it is possible to determine the source of the signal. Using specifically designed receivers we can determine the incoming signal direction and ray trace back to the originating point. Then the aspects of receiver theory and the practical implementation details are further explored. Finally, some preliminary results from early receiver experiments are presented.

1. Introduction

Remote Object Location (ROL) is valuable in a wide range of commercial and research applications. Of particular interest however, are technologies like cell phone location in an urban environment or item location in a warehouse. The unifying stumbling point of ROL in all of these applications is the multi path reflection problem. If a signal (radio, sound, etc), which is used to track an object, is reflected during any point along its path to the receiver it will have a longer time of flight (TOF) than if it traveled directly from the source to the receiver. However, the receiver is unaware of how many reflections an incoming signal encountered and therefore will be tricked into thinking the object is farther away then it actually is.

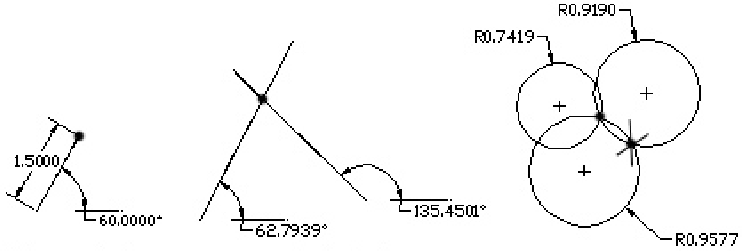
Each commercial domain has its own method for trying to solve this problem. Warehouse tracking technologies exists, such as Beaconing Bar Codes, for finding the stock locations. Specifically, when a given item's location is required, an electromagnetic radio pulse is sent out which wakes up tags placed on all inventoried items. When the inventoried item in question "hears" its specific signal it responds with its own radio wave response that is received and triangulated. Multi-path problems in this domain can be handled by using an algorithm that takes into account the reflective geometry of the environment [1]. These algorithms perform transforms the signal to eliminate areas that it could not have originated from based on the predefined geometry. However, while this is a powerful algorithm, it's also expensive to implement as it involves many mirroring and trimming operations on the geometry. It becomes much too expensive computationally when you attempt to implement it in 3D external environments.

Recently the U.S. cell phone industry has also been placed under pressure to develop similar technology to meet FCC regulations set forth in 1999. The new law states that a 911 operator has to be able to locate the position of the caller within a certain accuracy tolerance (around 100 meters) [2]. There have been several attempts to implement this technology through various methodologies. Many of these methods involved augmenting GPS into the cell phone network [3][4]. However, this is really just a temporary band-aid as GPS suffers from the same multi-path problems as any signaling method. The GPS simply gets a better line of site to the cell phone in an urban environment than the cell towers do. There are other technologies that involve mapping the unique radio signature from all the towers at every point in the city and then storing the result in a database, thus creating a spatial hash function. Of course, the prospect of having to map the radio signature to some reasonable degree for a whole city seems even worse than having to making a reflective model. Additionally, steps must be taken to eliminate ambiguities between any set of points that happen to have the same radio signature.

Creating models of what is reflective in a given environment is itself a non-trivial task, especially in the radio frequency domain. As such, for the purposes of this paper we constrict ourselves to the sound domain. The benefit to using sound is that all sound reflective objects can also be seen in the visible spectrum and thus visual methods can be used to build reflection maps. Several papers have been published on this topic making use of LIDAR and satellite imagery [5]. However, eventually a move to the RF spectrum would allow us to contribute something to cell phone location technology.

2. Problem Description

Traditionally, problems of this nature are solved using one of three basic types of triangulation (angle-angle, angle-distance, and distance-distance) with one or more receiving stations. Each type triangulation requires a different number of receivers (figure 1) due to ambiguities that arise from geometric assumptions implicit in the receiver design.

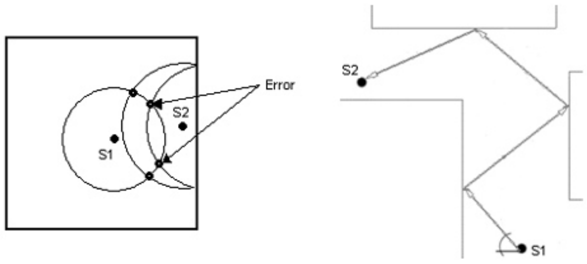


Distance-Angle Angle-Angle Distance-Distance

Figure 1 – Triangulation Types

Specifically, in an angle-angle system at least two receivers are needed because one direction is not enough to uniquely determine a point of origin. In distance-distance systems at least three receivers are needed. Using just two yields two possible points of origin, thus requiring a third to determine the true point of origin. It would initially seem that using an angle distance scheme is best on all counts because it requires less hardware i.e. only one receiver is needed instead of two or three. However, depending on the application, finding or designing a receiver that can determine the direction of the incoming signal may be difficult or expensive. Therefore it would be desirable to use fall back on another system that uses multiple cheap receivers.

As was mentioned previously, determining the distance to the point of origin can also be difficult due to multi-path complications when trying to use distance-distance style sensors. Namely, when the shortest path a signal takes from the transmitter to the receiver is longer then the Euclidian distance between the two due to reflection, the extra distance will push the possible location radius to a much larger value. However, if an angle-distance system is employed, and the geometry is known, ray tracing can be used to determine the location of the source. When a signal is detected we calculate the distance traveled based on the signals speed and TOF. Then we trace back this distance along the direction vector. Every time a wall is encountered we reflect the signal and follow the new direction.



Distance-Distance triangulation with reflection

Distance-Angle triangulation with reflection

Figure 2 – Distance-Distance vs. Distance-Angle

Round trip synchronization is usually used to determine how far away a transmitter is by "pinging." Specifically, the receiver sends a message to the transmitter, remembering at what time it sent it. The transmitter, upon receiving the "ping" immediately responds back. Once the receiver gets the response it again notes the time, and then can determine how long a signal took to reach the transmitter, assuming of course that we know how long the transmitter takes to respond and that the signal propagation speed remained constant throughout the course of its journey. This method however, fails if we wish to determine the distance of asynchronous sources, as we only know when the signal arrived, but not how long it traveled for.

3. Algorithm

Eliminating Synchronization Constraint

When we first approached this problem we were attempting to augment the BBC algorithm so that synchronization was not needed between the sender and the receiver [1]. We realized that synchronization is not possible when you have no control of the signal transmitter. Additionally, transmitter hardware is much simpler if it doesn't have to worry about responding to "pings" from a receiver.

When three small receivers are placed in a line with known distances between the receivers the distance to signal origin can be calculated by taking advantage of signal delays. Specifically, as a signal propagates it will encounter one of the receivers first. Upon detecting the signal the first receiver will start a timer. As the signal passes the other two receivers the intercept times are recorded. If the signal propagates at a constant speed and if there are no reflections in the immediate neighborhood of the receiver group then, by using simple trig, the time of flight can be computed.

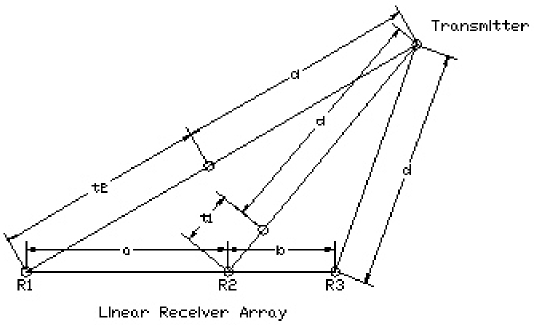
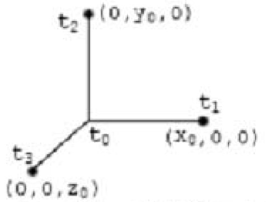


Figure 3 – Asynchronous Receiver

Eliminating Multi-path Issues

By using multiple receivers in a special configuration, it is also possible to convert the distance-distance configuration into a distance-angle one. As was mentioned before, a distance-angle configuration combined with reflective environment geometry can be used to solve the multi-path problem. In the simplest case, one could construct a sphere covered in receivers. Whichever receiver detected the signal first would point in the incoming signal's direction. However, this is hardly an efficient solution. If four receivers are arranged at known distances to each other, in a perpendicular fashion, in some known orientation in the world and if the same assumptions about propagation speed and reflection hold as the did for the linear receiver array, then both signal direction **and** distance to signal origin can be calculated. The equation derived to solve for distance is a quadratic expression yielding two solutions. Therefore two receiver "Orthants" are needed to mitigate the ambiguity.



$$x^2 + y^2 + z^2 = (r + t_0)^2$$
$$(x - x_0)^2 + y^2 + z^2 = (r + t_1)^2$$
$$x^2 + (y - y_0)^2 + z^2 = (r + t_2)^2$$
$$x^2 + y^2 + (z - z_0)^2 = (r + t_3)^2$$
$$\text{where } t_0 \text{ or } t_1 \text{ or } t_2 \text{ or } t_3 = 0$$

$$\begin{array}{lcl}
x = \left((t_0 - t_1)(2r + t_0 + t_1) + x_0^2\right)/2 & & x(r) = r(t_0 - t_1)/x_0 + \left(t_0^2 - t_1^2 + x_0^2\right)/2x_0 \\
y = \left((t_0 - t_2)(2r + t_0 + t_2) + y_0^2\right)/2y_0 & \longrightarrow & y(r) = r(t_0 - t_2)/y_0 + \left(t_0^2 - t_2^2 + y_0^2\right)/2y_0 \\
z = \left((t_0 - t_3)(2r + t_0 + t_3) + z_0^2\right)/2z_0 & & z(r) = r(t_0 - t_3)/z_0 + \left(t_0^2 - t_3^2 + z_0^2\right)/2z_0
\end{array}$$

Solving for Radius in 2 D :

$$Ar^2 + Br + C = 0$$

$$A = 4\left[y_0^2(t_0 - t_1)^2 + x_0^2(t_0 - t_2)^2 - x_0^2 y_0^2\right]$$
$$B = 4\left[y_0^2(t_0 - t_1)^2\left[t_0^2 - t_1^2 + x_0^2\right] + x_0^2(t_0 - t_2)^2\left[t_0^2 - t_2^2 + y_0^2\right] - 2t_0 x_0^2 y_0^2\right]$$
$$C = y_0^2\left(t_0^2 - t_1^2\right)\left[t_0^2 - t_1^2 + 2x_0^2\right] + x_0^2(t_0 - t_2)^2\left[t_0^2 - t_2^2 + 2y_0^2\right] + x_0^2 y_0^2\left[x_0^2 + y_0^2 - 4t_0^2\right]$$

Figure 4 – Derivation of signal direction from "Othant" receiver

Theoretical Implementation

With a theoretical framework in place, a 2D testing application was constructed in OpenGL that simulates a wave front as an expanding circle that can reflect off of a polygonal environment. A test application can also be made in 3D as the equations for signal direction remain very similar in form, however simulating the wave front and environment becomes much more expensive in 3D and there is no new insight gained in increasing the dimensionality of the problem. Eventually, a 3D version will be written so that the theory can be validated against experimental data.

The application supports arbitrary placement and orientation of receiver units as well as letting the user create a new wave front at arbitrary locations in the environment. Once a wave front passes over a receiver unit, a ray is cast from the receiver in the direction of the incoming signal, reflecting off any barriers that it encounters. Early testing of the program revealed that the scale of the receiver array is important. If the scale of the receiver unit is comparable in size to the reflective objects then it is possible for reflection to occur between the receivers. This reflection can obfuscate the time differences detected. Fortunately, this is a detectable phenomenon since any time delay longer than the time it takes for a signal to travel between the farthest two receiver nodes in the unit means a refection must have occurred.

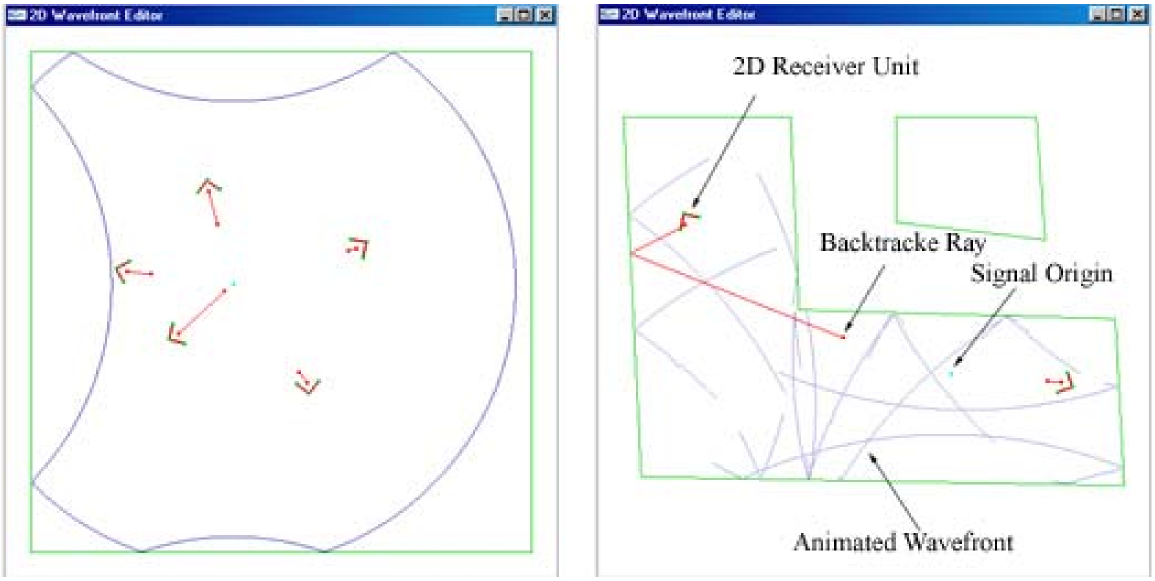


Figure 5 -2D wave front simulator screen shots

4. Practical Implementation

Our hope is to construct an actual receiver unit and then test it against an improved 3D simulator running a numerical sound field simulation algorithm [6]. If the simulation and the physical data correspond we will have further evidence to validate our theoretical model. Implementing a real world version of the algorithm is, of course, much more complicated. Primarily because there is no easy way to detect when the start of a specific waveform intersected a microphone. Additionally, ambient noise in the environment and signal degradation caused by surfaces that diffuse sound also add some degree of uncertainty when attempting to back trace the signal.

Currently, we are developing a way to detect a specific waveform and the time at which it first intersects an actively listening microphone (a receiver). Eventually, a computer with two sound cards will be reading audio input from four mono microphones. The input from these four microphones is then written into four separate circular buffers, parts of which are processed. A Fourier transform is run on the waveform data and then a spectrum mask is applied. This transform slides along the buffer using auto correlation to determine if some section of the audio in the buffer matches a specific waveform that we are looking for. If a match is found then the time at which the sound begins is marked for use by our source location algorithm. This is a common technique used in signal processing to search for a particular sound embedded in a complex waveform [7].

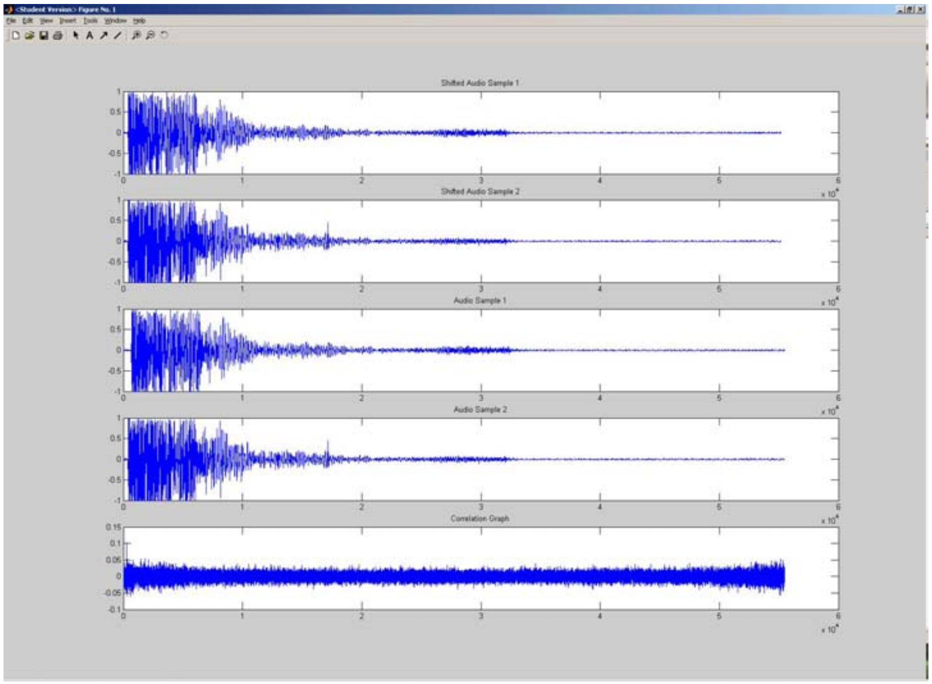


Figure 6 - Matlab Fourier Auto-correlation tests

While parts of the initial algorithm have been tested in Matlab, getting continuous real time results in a Java application will be a challenge due to the intense computational nature of the processing. At the moment we only have support for analyzing two waveforms at once and still need to make the Fourier Transform Autocorrelation code more efficient. However, once a prototype receiver is created we will be in a better position to determine if our model is accurate enough for practical applications.

Figure 7 - Java NoiseFinder application in development

5. Conclusion

A method for acoustical object location in reflective environments has been presented. It is based on using simple geometric arrangements of a small number of very simple receivers. By using orthogonal microphone geometries it is possible to calculate the direction and time of flight from an incoming signal. Using these two values coupled with an understanding of the surrounding reflective environment, it is possible to backtrack to origin of a signal. Once an optimized software signal processor is incorporated inter our algorithm it should be possible to test our geometric model in the real world and determine what refinements need to be made in our theory.

6. References

[1] Wang, Ming-En. "*Forward Ray Tracing and Remote Object Location In Two Dimensions*",1999.

[2] WTB / Policy. "*FACT SHEET - FCC Wireless 911 Requirements*".
http://www.fcc.gov/911/enhanced/releases/factsheet_requirements_012001.pdf
December, 2000.

[3] Diggelen & Abraham. "*Indoor GPS Technology*", Global Locate, Inc.
http://www.globallocate.com/files/IndoorGPSTechnology.pdf

[4] Moeglein & Krasner. "*An Introduction to Snaptrack Server-Aided GPS*", Proceedings of the Institute of Navagation conference, ION-GPS 1998.

[5] Iavarone, A. & Vagners, D. "*Sensor Fusion: Generating 3D by Combining Airborne and Tripod-Mounted LIDAR data*", International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 34-5/W10.
http://www.photogrammetry.ethz.ch/tarasp_workshop/papers/iavarone.pdf

[6] Rabenstein, R. & A. Zayati, A. "*Sound Field Simulation by Computational Acoustics. Part 1: Simulation Algorithm*", Int. Journal of Adaptive Control and Signal Processing, vol. 14, 2000, pp. 663-680.
http://www.Int.de/~rabe/ps/ACaSP.ps.gz

[7] Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; and Vetterling, W. T. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge, England: Cambridge University Press, 1989.