On the Feasibility of Real-Time Phone-to-Phone 3D Localization

Jian Qiu[1†], David Chu[2], Xiangying Meng[3†], Thomas Moscibroda[4]
[1]China Mobile Research Institute, [2]Microsoft Research, [3]Peking University, [4]Microsoft Research Asia
qiujian@chinamobile.com, {davidchu, moscitho}@microsoft.com, xiangying.meng@pku.edu.cn

Abstract

High-speed, locational, phone-to-phone (HLPP) games and apps constitute a provocative class of mobile apps that are currently unsupported on commodity mobile devices. This work looks at a key problem for enabling HLPP: a specific variant of the localization problem in which two phones estimate each other's relative positions in 3D space without infrastructure support. Moreover, position estimates should reflect changes due to the phones' possible mobility.

We present a solution for achieving high speed 3D continuous localization for phone-to-phone scenarios. Our basic approach uses acoustic cues based on time-of-arrival and power level. It assumes at least two microphones and one speaker per phone, which is common on new smartphones. Accelerometers and digital compasses assist in resolving ambiguous acoustic-only localization. Continuous localization is achieved with the aid of a loose time synchronization protocol and an extended Kalman filter. Experimental results across a range of motion paths show localization resolution to within 13.9 centimeters for 90% of estimates, and to within 4.9 centimeters for 50% of estimates when the phones are several meters apart.

Categories and Subject Descriptors

C.5.3 [**Computer System Implementation**]: Microcomputers—*Portable devices*

General Terms

Algorithms, Design, Performance

Keywords

acoustic localization, mobile phones, mobile gaming

[†]Work performed while at Microsoft Research

SenSys'11, November 1–4, 2011, Seattle, WA, USA.

1 Introduction

The proliferation of smartphones and the growth of a rich accompanying application ecosystem has given rise to effective solutions for countless applications scenarios. But even with the most modern and sophisticated of smartphones, certain classes of applications remain elusive. Consider for example a real-time, face-to-face multiplayer pong game, in which two players Alice and Bob are located in the same room, and use their phones as if they were paddles. Alice and Bob wave their phones; the orientation and location of one phone relative to the other affects the ping pong ball's flight and the game's outcome. Similarly, consider a high-speed, real-time sword fight in which each participating player uses her phone to simulate a sword. Or perhaps Alice and Bob wish to combine their mobile device screen surfaces into one extended gaming surface. The devices could be in any orientation and position relative to one another.

Such *high-speed, locational, phone-to-phone (HLPP) games* do not exist on commodity phones and enabling them seems particularly challenging. Existing solutions that provide real-time, locational functionality in commercial game consoles such as the Kinect or the Wii rely heavily on the existence of fixed, external infrastructure such as microphone arrays or cameras. In contrast, enabling HLPP games such as pong or sword-fight (as well as similar non-gaming HLPP applications) would require implementing a similar functionality entirely on the two participating phones, without the help of any such external infrastructure.

The underlying technical challenge that has to be solved in order to enable these novel HLPP-type applications on smartphones is a new real-time 3D mobile device localization problem. While numerous variants of localization and positioning problems have been studied in the context of mobile phones, the one required for our purpose is particularly challenging, because the two devices need to establish a relative coordinate system with neither the use of additional infrastructure nor hardware modifications. Moreover, the coordinate system must be maintained in real-time, i.e., as the relative positions of the phones change, position estimates should be continuously revised with delay that is ideally imperceptible to human observers.

In this work, we show that in principle, such fine-grained, real-time 3D localization is feasible on commodity smart phones. Our basic insight is to leverage the different sensors already commonly found on mobile devices. The core

algorithm uses each device's multiple microphones to perform *acoustic 3D triangulation* and derive position estimates. Some smartphones such as Apple's iPhone 4 and Google's Nexus One already ship with two mics for video conferencing purposes, and more devices are continuing this trend. To accurately perform triangulation, we develop a new method that combines time-of-arrival (TOA) and signal power cues. In addition, each device uses its standard accelerometer and digital compass to assist triangulation by resolving ambiguous positions, and identifying *alignment regions* when two phones might be well-positioned for triangulation. Outside alignment regions, inertial displacement is used to estimate position. In order to enable each phone to track the other phone's movement, position estimates are continuously collected, and our algorithm employs a Kalman filter to smooth point samples and decrease measurement variance. The filter accounts for the various forms of measurement errors inherent in the phone-based triangulation process.

The idea of using acoustic signals for deriving information about two phone's relative position was first used in BeepBeep [12], which showed that a pair of audio tone exchanges between two phones can be used to estimate distances between two phones. Our solution builds on these ideas by showing that with the benefit of additional sensors on commodity phones, one-dimensional position can be extended to real-time three-dimensional position.

We have implemented our new high-speed, phone-to-phone 3D localization algorithms on commodity Android devices. Our preliminary investigation shows that using the combination of sensors and acoustic signaling, two phones can localize each other's relative 3D position with under 13.9cm error over 90% of the time, and under 4.9cm error over 50% of the time at a distance of up to several meters, even without additional infrastructure support. Moreover, by employing Kalman filtering as well as other means, we can maintain continuous real-time localization under simulated motion paths with both linear and angular position change. Our findings are preliminary in that we have not evaluated robustness to noise and live motion paths.

The paper is organized as follows. §2 discusses background on acoustic ranging, and particularly the BeepBeep [12] system. §3 discusses the challenges, and how they impact our system design. A detailed description of our techniques to enable static and real-time/mobile localization are then given in §4 and §5, respectively. §6 discusses implementation details, and §7 presents the results of our evaluation. Finally, related work and other interesting aspects are discussed in §8 and §9.

2 Background

Acoustic signaling on mobile phones has been used for estimating distances between two phones. Our real-time 3D positioning algorithm uses one such system, BeepBeep [12], as a building block. We therefore review the BeepBeep mechanism in this section.

BeepBeep is a pairwise acoustic ranging (1D localization) system for COTS devices. Its main advantages are that it does not rely on time synchronization, and is robust to soft-

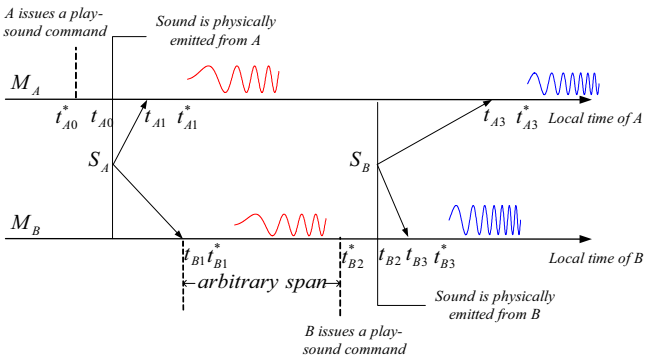


Figure 1. BeepBeep mechanism for distance ranging

ware delays. Each device is assumed to have one mic and one speaker, and can communicate through WiFi or another radio protocol. The BeepBeep ranging algorithm (see Figure 1[1]) is based on active acoustic time-difference-of-arrival. It operates as follows. Device A sends an audio Tone A at a time of its choosing, t_{A0}. Device A and Device B both record the arrival of Tone A at their respective microphones and timestamp the arrival with the local clock time, t_{A1*} and t_{B1*} respectively. At some arbitrary point in the future t_{B0}, Device B emits an audio Tone B, which is similarly recorded and timestamped by both Device A and B as t_{A3*} and t_{B3*} respectively. Each device calculates the interval (in local time) between the arrival of Tone A and Tone B. In Figure 1, this is $t_{A1*} - t_{A3*}$ for Device A and $t_{B1*} - t_{B3*}$ for Device B. Lastly, the devices exchange the interval information (e.g. via WiFi) and subtract the intervals from each other. The final difference is the time of flight of the two tones, or the roundtrip time between Device A and B. Since the speed of sound can be assumed constant, the ranging distance can be estimated as half the roundtrip distance.

Note that only the sum of the distance from Device A's mic to Device B's speaker and from Device B's mic to Device A's speaker is available with this method. The component distances are not computed directly.

3 Challenges and Design Overview

Enabling HLPP-type games and applications requires the solving of a novel 3D localization problem that needs to be accurate, real-time, and run on the two participating phones only. These requirements result in a number of unique and interesting challenges. In this section, we discuss some of these challenges and show how they impact the design of our suggested solution.

3.1 Requirements

Centimeter-resolution accuracy: The class of HLPP (gaming) applications we envision demand very high resolution accuracy. This suggests that omnipresent commercial geolocation infrastructure such as GPS satellites, WiFi access points, or 3G cell towers are unable to provide the desired accuracy. Whereas these techniques can deliver resolution at the granularity of one or more meters, we seek substantially better accuracy.

[1] Included with permission from the authors of [12].

Ad hoc usage without infrastructure support: Densely deployed localization infrastructure such as in Active Bats [8] and Cricket [13] can achieve high accuracy, but they obviously cannot be used on mobile smartphones. Our goal is to achieve highly accurate, real-time positioning information *in the absence* of external infrastructure, relying entirely on phones.

Mobility tolerant: HLPP applications such as the pong game or sword fight rely on people's mobility to play the game. Therefore, localization must be able to cope with constantly changing motion up to the speed of reasonable human movement.

Works with existing phone hardware: Finally, we seek to understand the possibilities and limitations of real-time, phone-to-phone 3D localization on *commodity phones*, rather than on some sophisticated custom hardware. This voluntary constraint limits the resources and inputs that can be used by our algorithms to the sensors currently found on smartphones.

Furthermore, we are most intrigued by scenarios in which the distance separation between the phones is in the range of normal human social interaction. For example, in casual gathering places like a bus stop, family room and coffee shop, people are typically interacting with each other within a distance of several meters. This also suggests that we can reasonably assume line-of-sight between the two parties, as it is more typical for people to engage in prolonged interactions while within sight of the other party.

3.2 Design Overview

The requirements discussed in the previous section guide our system design. For example, WiFi-based localization solutions are insufficient since they would not meet our accuracy requirements, which suggests the use of acoustic signaling. However, a simple, BeepBeep-like acoustic system fails to solve the problem since it only provides 1D distance information at any given time step.

In fact, it is easy to see that at least two mics or two speakers are necessary in order to provide 2D position information. By the principal of acoustic reciprocity, a transmission from speaker to mic is equivalent to a transmission with the speaker and mic reversed [11]. Therefore, exchanging each phone's local 2D measurement doubles the number of 2D measurements per phone. At this stage, as long as the 2D measurements are not coplanar, possible 3D positions have been reduced to two. In principle, three mics (or three speakers) on one phone is sufficient for completely passive 3D localization, and eliminates the need for any communication. However, no commodity phones are yet equipped with such capabilities. Therefore, it is mandatory that we engage pairwise data exchange and extra sensory information in order to resolve the remaining ambiguity. Information from these extra sensors, accelerometer and digital compass ubiquitous to new phones, are non-trivially combined to resolve to one unique 3D position.

Accelerometers and compasses have another use as well. Acoustic localization performs well when the two phones are favorably positioned within one another's *alignment region*. Intuitively, the alignment region corresponds to cases in which the two phones' speakers and mics are facing each other. Outside of the alignment region, we engage the assistance of accelerometer and compass for displacement tracking. However, displacement tracking is susceptible to significant error accumulation over time. Therefore, our basic algorithm consists of *primary mode* supplemented with *fallback mode*. Primary mode consists of acoustic localization using the two mics available on the phone. Fallback mode consists of IMU-based displacement tracking. Figure 2 shows a commercially-available phone with two mics, and its canonical axes. The main focus of this work is primary mode. We start by providing an overview of the primary mode protocol.

Our protocol in primary mode starts off by executing the *Initialization Stage*. The protocol then switches to continuously iterating through *Tone Exchange Stage*, *Distance and Angle Measurement Stage*, and *Position Estimation Stage*.

1. *Initialization Stage:* Two phones X and Y establish loose time synchronization by calculating WiFi round trip time and exchanging local clock values.
2. *Tone Exchange Stage:* One of the phones, say X, turns on its two microphones and sends a "start" message to phone Y. Upon receiving "start", Y replies with a "confirm" message, turns on its two microphones for recording, and sends an audio tone as soon as its microphones are turned on. After receiving the "confirm" message, X schedules its time to send its audio tone based on the time sync information. Microphones on both phones are turned off after X finishes sending its audio tone. During this stage, both phones also sample their accelerometer and geomagnetic sensors to obtain a rotation matrix.
3. *Distance and Angle Measurement Stage:* The recorded tones are first correlated with reference tones to compute raw TOA cues. Then the two phones exchange their TOA cues to calculate distance values and local angles. The local angle measurement and the rotation matrix obtained in the previous stage are transmitted together to the other phone.
4. *Position Estimation Stage:* The distance measurement, two angle values and rotation matrices are fed to a Kalman filter to estimate the phone's relative 3D location coordinate.

The next two sections establish the algorithmic details by which we accomplish this protocol.

4 Static Localization

In this section, we derive the basic positioning mechanism, assuming that the two mobile phones are static, i.e., they do not move. We first consider two localization cues for the simpler 2D problem, which will then serve as building blocks for our solution to the full-fledged 3D case.

4.1 Localization Cues

In primary mode, triangulation is based on two pieces of information: TOA and power difference. Each of TOA and Power contributes to its own individual angle estimate. These angle estimates are later combined in §5.3.

4.1.1 Time of Arrival Cues

Assume that each phone has two mics and one speaker, labeled as in Figure 3. Using BeepBeep as a subprocedure,

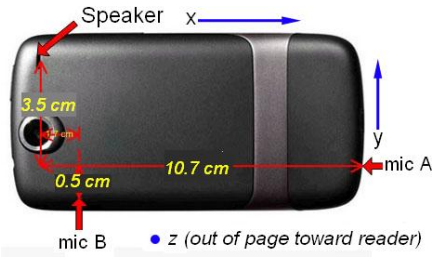


Figure 2. Typical phone with 2 mics and 1 speaker

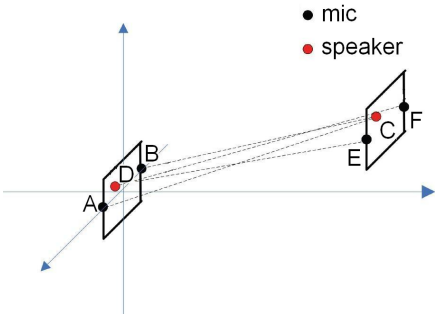


Figure 3. Two phones each with 2 mics and 1 speaker

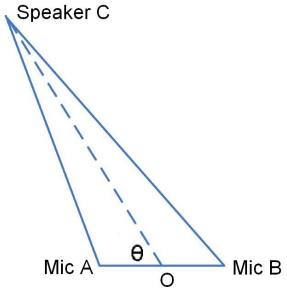


Figure 4. Deriving θ by the law of cosines

we obtain four distances for each pair of mic-speaker combinations.

$$d_1 = AC + DE \qquad d_2 = AC + DF$$
$$d_3 = BC + DE \qquad d_4 = BC + DF$$

While we cannot solve these equations for any individual mic–speaker distance, we can solve for the difference of two mic–speaker distances for mics hosted on the same phone:

$$DE - DF = \frac{d_1 - d_2 + d_3 - d_4}{2} \quad (1)$$

$$AC - BC = \frac{d_2 - d_3 + d_1 - d_4}{2} \quad (2)$$

Consider the triangle shown in Figure 4 formed by mic A and mic B on the local phone and speaker C on the remote phone. Let O be the point that bisects AB, and let θ be the angle $\angle COA$. The distance between the two mics AB is fixed by the geometry of the phone.

We derive θ using the distance measurements above. From the law of cosines, it follows that

$$\cos(\theta) = \frac{AO^2 + OC^2 - AC^2}{2AO \cdot OC}$$
$$\cos(\pi - \theta) = -\cos(\theta) = \frac{B0^2 + OC^2 - BC^2}{2BO \cdot OC}.$$

Combining these equations, and solving for θ yields

$$\theta = \arccos \frac{(BC - AC)(BC + AC)}{AB \cdot (2 \cdot OC)} \approx \arccos \frac{BC - AC}{AB},$$

where $BC + AC$ is approximated as twice OC. This approximation is accurate when AB is small relative to OC, which is typically the case as the distance to the remote phone is much greater than between the mics on the local phone.

The algorithm uses a similar computation for triangle DEF, resulting in an estimate for an angle ϕ, albeit in a potentially different 2D plane.

4.1.2 *Power Difference Cues*

The idea of using power difference as an additional cue stems from the observation that the received acoustic signal of the contralateral mic experiences greater attenuation than that of the ipsilateral mic. We use this difference in attenuation as an additional indicator for the relative angle between the two phones.

The principle of acoustic attenuation is complex and we did not find it straightforward to model it analytically. For example, Figure 5 shows that neither mic's attenuation follows a simple relationship with relative azimuth angle. For example, mic 1's position at the back of the phone suggests a power peak around 0 degrees. Instead, a power valley occurs between -30 and +15 degrees at 1m. Similarly, mic 2's position at the base of the phone would suggest a monotonic decrease as the azimuth rotates away from mic 2. This relationship is evident at 1m, but at 2m there is non-monotonic behavior at -75, -15, and above +45 degrees.

Instead of an analytical model, we therefore use empirical measurements to construct a lookup table with the following attributes.

- *distance*: measured distance.
- *power_ratio*: measured ratio of mic 1's power to mic 2's power.
- *angle_mean*: expected angle at the measured distance and power ratio.
- *angle_stddev*: expected standard deviation of the expected angle.

As part of each position estimation, the power difference and distance are mapped to estimates of angle and standard deviation. These constitute the power difference cue.

Figure 6 shows the power ratios corresponding to Figure 5. The non-injective relationship between angle and power ratio means a given power ratio is not definitively mapped to a unique angle. In order to resolve the ambiguity, each curve is partitioned into piecewise injective relations. For example, the power ratio at 1m is represented by two relations: one from [-90,+15), and another from [+15,+75]. To use the appropriate piecewise relation, an angle estimate from the TOA cue is first calculated by the technique described in §4.1.1. The TOA cue's accuracy is not a concern here because it is only necessary to select among relatively large partitions in the angle space. Finally, the power information is mapped to an angle estimate.

Undesirable attenuation may occur even with the assumption that two users maintain line of sight. For example, a user's hand grip may block her phone's speaker or mics. This specific situation can be corrected by comparing the captured signal from the local speaker to the local mic with its known signature, and upon detecting a mismatch, reminding

the user to unblock the speaker and mics.

4.1.3 Alternative Cues

Note that in principle, additional cues for position estimation could be used. We did consider several of these, but for various reasons decided to employ only TOA and power difference. For example, one tempting cue is the *spectral transformation information* which is an important cue for human hearing localization [6]. The high frequency component of an audio signal can be blocked by a human's head, resulting in a frequency difference of the received signal between the human's two ears. The spectrum difference is obvious when the wavelength of the high frequency component in the signal is much smaller than the size of the human's head. However, the limited frequency response of microphones in current phones (the receiving bandwidth commonly has an upper limit of 8k Hz) prevents us from using a broadband audio signal, and the wavelength of currently used signals is comparable to the size of the phone. Therefore, we found in our tests that due to the diffraction effect, the received signals at the two mics are insufficiently different in the frequency domain for our purposes.

4.2 Static 3D Localization

The TOA and power cues discussed in the previous section yield for each phone 2D angle information relative to this phone's own coordinate system. The next step is now to combine these cues from both phones, and calculate each phone's 3D position. To do so, we must translate the angle calculated in one phone's (phone Y) coordinate system to an angle in the other phone's (phone X) coordinate system.

Let the two mics of phone X lie along a phone's x-axis, and let the z-axis be normal to the face of the phone. The 2D angle θ calculated by this phone and the measured distance d define a circle whose center is on the x-axis as illustrated by *Circle 1* in Figure 7. This circle is the well-known *cone of confusion* [3]; with one phone's measurements alone, the location of the remote phone is underconstrained and can lie at any point along the rim of the circle. Fortunately, with both phone's measurements and with additional sensor information, the ambiguity can be resolved.

The coordinate of the center of the circle can be represented by a vector $\vec{v_1} = (d\cos\theta, 0, 0)$. The radius of the circle is $d\sin\theta$. Thus the circle is determined by its center's coordinate and radius. Similarly, let the remote phone also define its own circle $\vec{v_2} = (d\cos\phi, 0, 0)$ relative to its coordinate system, where ϕ is an angle in the remote phone's coordinate system.

The translation – which must map the mirror of $\vec{v_2}$ which is $\vec{v_2^-} = (-d\cos\phi, 0, 0)$ in the local phone's coordinate system – is computed as follows. First, the *rotation matrix* R is used to map a vector in a phone's coordinates to Earth coordinates. R is calculated by using data from gravity (accelerometer) and geomagnetic (digital compass) sensors. On the Android operating systems, R is provided as a library function. The vector to represent the same circle on another phone is:

$$\vec{v_2^*} = R_1^{-1} R_2 \vec{v_2^-} \tag{3}$$

The vector is translated from the local phone's coordinate system to Earth coordinate system using rotation matrix R_2, and then translated to the other phone's coordinate system using rotation matrix R_1. This gives us two circles on the local phone's coordinate system, and their intersection point $\vec{p} = (p_x, p_y, p_z)$ is the coordinate of the remote phone. Figure 7 shows two circles: *Circle 1* is given from $\vec{v_1}$ calculated with the local angle measurement, and *Circle 2* is transferred from the remote phone.

We now have three equations:

$$|\vec{p}| = d \tag{4}$$

$$\frac{\vec{p}\cdot\vec{v_1}}{|\vec{x}||\vec{v_1}|} = \cos\theta \tag{5}$$

$$\frac{\vec{p}\cdot\vec{v_2^*}}{|\vec{x}||\vec{v_2^*}|} = \cos\phi \tag{6}$$

We illustrate how the coordinate $\vec{p}$ of the remote phone is obtained using a closed form method. Equation (5) is solved first and we obtain $p_x = d\cos\theta$. p_x is then substituted into equation (4) and (6). As long as the two phones' x-axis are not parallel to each other, a pair of p_y and p_z values can be obtained by solving these two equations. Since there are actually two intersection points of the two circles, we eliminate the point with negative p_z on one of the two phones' coordinate system based on each phone's z-axis accelerometer. As shown in Figure 7, p_z of $\vec{p}$ and p_z of $\vec{p'}$ are both positive in the local phone's coordinate system. However, p_z of $\vec{p'}$ is negative in the remote phone's coordinate system, hence $\vec{p'}$ is eliminated. As an alternative to the closed form method, the least squares method can be applied for obtaining a single coordinate. In addition, the next section achieves continuous localization by employing Equations (4)-(6) in a linear estimator model.

5 Continuous Localization

The basic procedure for continuous localization simply takes sequential static position estimates as fast as possible. We are able to cope with very fast changes in relative position by tackling the following additional challenges.

5.1 Motion-Induced Measurement Error

Our static acoustic localization algorithm assumes that both phones are stationary during the exchange of audio tones. Figure 8 illustrates the measurement error caused by translational and rotational movement during one-way audio tone exchange. The error s is limited to the displacement of the phone in the intervening period between the reception of the tone at the ipsilateral mic B and at the contralateral mic A.

For translation motion shown in Figure 8(a), consider a tone that reaches B and A at time t_1 and t_2 respectively. Translation motion away from the tone between the time when the tone reaches B and when it reaches A causes the contralateral mic to receive the tone at position A' at time t_3. We assume that the distance to the remote phone dominates the distance between the mics AB, so $\theta' \approx \theta$, and that the line k tangent to t_2 at x_3 intersects A. Therefore, s can be

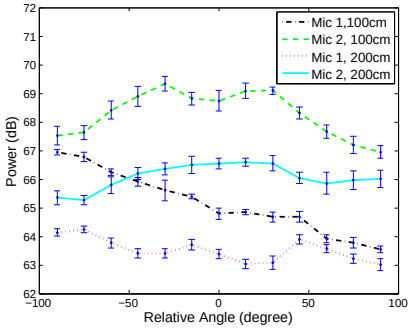


Figure 5. Signal power readings measured at various azimuth angles

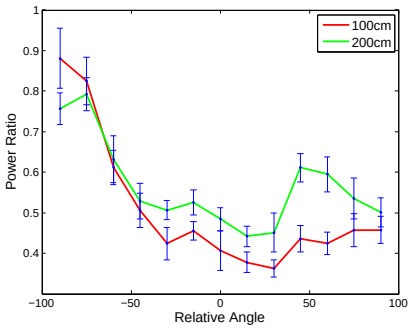


Figure 6. Power ratios at various azimuth angles

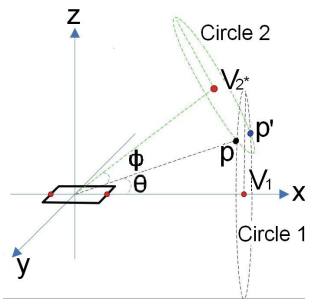
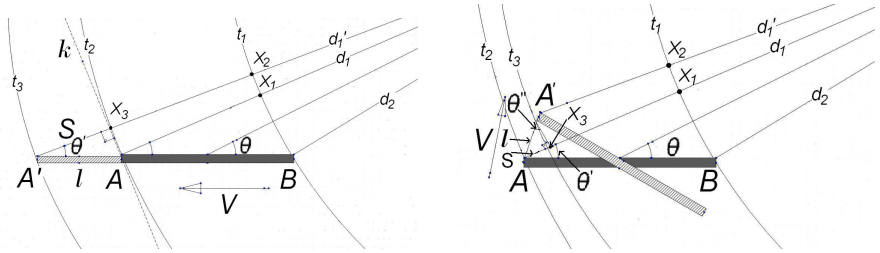


Figure 7. Mapping remote phone's angle to local phone's coordinate system



(a) Translation Error (b) Rotation Error

Figure 8. Motion-induced measurement error

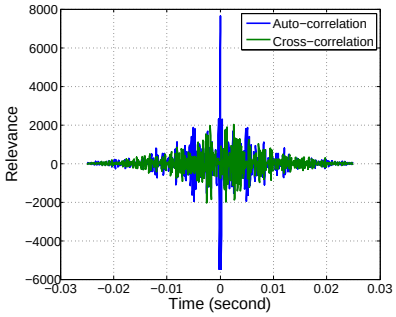


Figure 9. Overlapped tones detected

calculated as:

$$s = d_1' - d_1 \approx l\cos\theta$$

where l is found by comparing the distance sound travels to the distance the phone travels in the interval $[t_1, t_3]$.

$$l = (d_1' - d_2)\frac{v_{translation}}{v_{sound}}$$

For rotation motion shown in Figure 8(b), we continue our assumptions from above. Specifically, we assume that $\angle A'AB$ is a right angle because the absolute separation between A and A' is small, providing $\theta'' \approx \theta' \approx \theta$. Also, we assume that $A'x_3$ is a reasonable approximation of the tone's arc at t_3. Therefore:

$$s = d_1' - d_1 \approx l\sin\theta$$

Fortunately, the practical consequences of movement error are limited due to slow human movement velocity relative to sound velocity. Table 1 shows the degree of angle measurement error resulting from various movement velocities and phone orientations.

velocity (m/s)	orientation (degree) 20°	45°	70°
Translation			
1m/s	0.416	0.113	0.020
5m/s	2.200	0.574	0.100
10m/s	4.779	1.169	0.204
20m/s	12.58	2.436	0.420
Rotation			
1m/s	0.149	0.112	0.054
5m/s	0.731	0.555	0.268
10m/s	1.432	1.093	0.530
20m/s	2.755	2.127	1.031

Table 1. Impact of phone velocity and orientation on angle error (degree)

5.2 Time Synchronization and Tone Overlap

The preceding analysis only considers error introduced by one-way error tone transmission. BeepBeep acoustic ranging and our usage of it described thus far performs an unsynchronized two-way tone exchange. During the arbitrarily long round trip time, phone displacement may be significant. Therefore, we use two techniques to minimize round trip time. First, the phones perform loose time synchronization once at initialization. Second, the phones exchange tones in close succession with partial overlap. The net result is a decrease in two-way tone exchange time to nearly that of one-way tone exchange time plus clock sync error.

Loose time sync is performed as follows. The phones first exchange several pings with CSMA backoff disabled to determine stack traversal plus WiFi round trip time. The phone with the lower device id then sends its local clock value (again without CSMA backoff) to the phone with the higher device id which adjusts its local clock by the appropriate offset minus estimated WiFi round trip time. This suffices to synchronize clocks within 10 milliseconds.

Once synchronized, the phones exchange tones at a fixed period and phase. Because the two directions of the audio exchange may overlap with one another, the tones must be easy to detect despite interference. We choose tones with low cross-correlation (interference rejection) and high

auto-correlation (pulse detection) following the techniques in [7]. Figure 9 shows strong relevance impulse for the auto-correlated tone but not for the cross-correlated tones, indicating that we can robustly distinguish tones.

Tone overlap does introduce the issue that the local phone's signal may be strong enough to mask the envelope of the remote phone, especially when the distance between the two phones is large. It will increase error of correlation measurement even though special tones are used. Such overlapping impacts both power cue extraction and TOA cue extraction. Therefore, our approach is an adaptive one in which we insert a gap period between tone starting times and adjust the gap based on the distance of the two phones and motion speed. When the distance of the two phones is small and the motion speed is fast, the gap is decreased as long as two tones can still be successfully detected. When the distance of the two phones gets longer, the gap is increased until there is no overlap. This pleasantly trades off static accuracy for motion error sensitivity to achieve higher overall continuous localization accuracy.

5.3 Extended Kalman Filtering

We use an Extended Kalman Filter to smooth the measurements from TOA and power cues, and track the motion of the phone. The filter recursively updates its state of the phone's position, and adjusts the state based on the noisy measurements of angle and distance. The filter model is similar to the one proposed in [2] which is developed to track human motion based on acoustic signals received by a passive microphone array. We adapt it to use our TOA and power cues.

In our model, the state contains remote phone's position, velocity and acceleration on each axis of the local phone's coordinate system. The state vector at time k is: $s_k = [\ p_x \quad p_y \quad p_z \quad v_x \quad v_y \quad v_z \quad a_x \quad a_y \quad a_z\ ]^T$, and the state estimation equation is:

$$s_k = As_{k-1} + w_{k-1}$$

where A is state transition matrix, and w_k is a random variable representing the state update error.

The measurement vector z contains the distance between two phones d and two angles (one angle θ measured by the local phone and another angle ϕ measured by the remote phone). The measurement at time k is $z_k = [d, \theta, \phi]^T$. The relationship between state vector and measurement vector, which can be obtained using 3D coordinate transfer as shown in Equations (4)-(6), is nonlinear. Hence the measurement equation at time k must be linearized as shown below to fit the Kalman filter model.

$$z_k = h(s_k^-) + H_k(s_k - s_k^-) + v_k$$

where the linearized matrix is

$$H_k = [\frac{\partial h(s_k)}{\partial s_k}]_{s_k^-}$$

s_k^- denotes the estimated state before it is corrected by the measurement. The random variable v_k represents measurement error, which in our problem gives error of distance and angle measurement. w_k and v_k are assumed to be Gaussian white noise and known a priori. Their covariance matrices are denoted as Q_k and R_k, respectively. Q_k is determined by standard deviation of updated state and state update speed. To obtain Q_k, we assume that the estimation noise comes from acceleration estimation error which then leads to velocity and distance estimation error.

$$Q_k = GG^T\sigma_a^2$$

where $G = [\ \frac{\triangle t^2}{2} \quad \frac{\triangle t^2}{2} \quad \frac{\triangle t^2}{2} \quad \triangle t \quad \triangle t \quad \triangle t \quad 1 \quad 1 \quad 1\ ]^T$. In the equations, σ_a denotes the standard deviation of estimated acceleration, and $\triangle t$ denotes the time difference between two consecutive measurements.

R_k can be obtained if the standard deviation of distance and angles measurements are known. Let the standard deviation of distance and two angles be σ_d, σ_θ and σ_ϕ, respectively.

$$R_k = [\ \sigma_d \quad \sigma_\theta \quad \sigma_\phi\ ]^T[\ \sigma_d \quad \sigma_\theta \quad \sigma_\phi\ ]$$

To simplify the calculation in our problem, we assume that the angle and distance measurement errors and estimation error are independent Gaussian distributions.

Based on the two equations above, the Kalman Filter algorithm falls into two steps running recursively: *predict* step and *correct* step.

1. In *predict* step, a new state is updated based on state estimation equation, and an estimated error covariance matrix P_k^- is obtained.

$$s_k^- = As_{k-1}$$
$$P_k^- = AP_{k-1}A^T + Q_k$$

2. In *correct* step, blending factor K_k is calculated at first, then the state is corrected based on the measurement residual. Finally, the estimate error covariance matrix is updated.

$$K_k = P_k^- H^T(HP_k^- H^T + R_k)^{-1}$$
$$s_k = s_k^- + K_k(z_k - h(s_k^-))$$
$$P_k = (I - K_k H)P_k^-$$

Since we have multiple cues (TOA and power), we can get multiple angles measurements at one time. There are two approaches to combine these multiple measurements. One approach calculates a weighted angle at time k which then is fed into the Kalman filter. The weight of each cue is determined by the standard deviation of the angle calculated by the cue. Let α_P and σ_P denote the angle measured by power and its standard deviation. Let α_T and σ_T denote angle measured by TOA and its standard deviation. The weighting equation is shown as below.

$$\alpha_C = \frac{\sigma_T^2}{\sigma_T^2 + \sigma_P^2}\alpha_T + \frac{\sigma_P^2}{\sigma_P^2 + \sigma_T^2}\alpha_P \tag{7}$$

Another approach assumes that the angle measurement of each cue is obtained at different time point ($k + \triangle$ and $k - \triangle$), then angles from different cues are all used during filtering process. The measurement error covariance matrix

R should be recalculated at each time an angle measured by different cue is used. We have used the first approach to combine the measured angles by different cues which has lower time complexity.

5.4 Fallback Mode

When localization detects that the phones' relative positions are near the alignment region, we switch from primary mode to fallback mode. Fallback mode has two objectives: (1) to provide location estimates while the two phones are not in the alignment region, and (2) to detect when the two phones have re-entered each other's alignment region. Both of these objectives are accomplished by continuously calculating the displacement of each phone with accelerometer and compass readings. Fallback mode operates as follows.

- Initialization establishes initial coordinates based on the last primary mode coordinates.
- Phones continuously exchange accelerometer and compass readings.
- Based on local and remote accelerometer and compass updates, each phone calculates its relative displacement, and updates its position estimate.
- If it is detected that the phones are back within the alignment region, primary mode is restarted.

The accelerometer is sampled very frequently at 20-50Hz. A key requirement of fallback mode is to quickly update position estimates based on initial position estimates of the local phone $p_1 = (0,0,0)$ and remote phone $p_2 = (x_2, y_2, z_2)$, and the accelerometer and compass updates $u_1 = (a_1, c_1)$ and $u_2 = (a_2, c_2)$ where a and c each consist of triples (x,y,z). Our phone-relative localization scheme assumes that $p'_1 = (0,0,0)$, and therefore:

$$p'_2 = p_2 + R_2^{-1} R_1 d(a_2) - d(a_1)$$

where R_1 and R_2 are Rotation Matrices as described in §4 and $d(a) = v\Delta T + \frac{1}{2} a \Delta T^2$. The initial velocity v is estimated by the state of Kalman filter at the initial position. With updated position p'_2, detection of the alignment region is performed. Let ρ be the empirically determined angle that defines the boundary of the alignment region. Let θ be the angle of p'_2 relative to the z-axis (recall the axes defined in Figure 7) in the plane defined by p'_2, mic A and mic B. Then we have:

$$\theta = \arcsin \frac{x'_2}{|p'_2|}$$

A comparison of $\theta > \rho$ serves to indicate whether the remote phone is outside the alignment region.

5.4.1 *Identifying the Alignment Region*

We propose a systematic method to identify the optimal alignment region. The idea is to select a fallback mode or primary mode based on the expected error. The primary mode error is angle-dependent, whereas the fallback mode error is time-dependent. Specifically, the fallback mode error increases over time because of displacement measurement error accumulation. Fortunately, primary mode measurement is not subject to time-dependent error, and therefore any primary mode measurement acts as a new initial position for fallback mode. Therefore, we know the primary mode error from Equation (7), and we know the fallback mode error because we bound the time between measurements according to §5.2. For every angle θ, we precompute both expected primary and fallback error. Past a certain angle ρ, the expected primary error will start to exceed the expected fallback error. At this point we switch to fallback mode.

We define an indicator $I = e_{fallback}(n\Delta T) - \varepsilon e_{primary}(\theta)$. $e_{fallback}(n\Delta T)$ denotes the position error after running fallback mode $n\Delta T$ time. It can be approximated by the standard deviation of acceleration σ_a as $e_{fallback} \approx |\sigma_a^2 (n\Delta T)^2/2|$. $e_{primary}(\theta)$ denotes the position error of primary mode at angle θ. It can also be obtained if the standard deviation σ_θ is known. If the indicator I has positive value, the phone is changed to fallback mode until I changes to positive.

While the above method identifies the optimal alignment region, we used a heuristic for simplicity. Namely, we operate in primary mode whenever we are between the angles $(-90, 90)$. This heuristic is based on prior work [7] that indicates acoustic localization operates poorly when mics and speakers are turned away from each other.

6 Implementation

We implemented our primary acoustic localization scheme on the Android v2.2 operating system. We have implemented fallback mode in a simulator, and are currently incorporating it into the phone implementation. Below, we discuss implementation details.

Audio Signal Selection: Audio signal selection is important for angle and distance measurement by the TOA cue. Due to the fact that speakers and mics on smartphones are not ideal devices, audio signals will be distorted when sent and received. These distortions are the main cause for angle and distance measurement errors. To reduce this error, the signal should be carefully selected based on the frequency response of the phone's speaker and mics. We have tested with different parts of the audio spectrum (2-6 kHz, 2-4 kHz, and 4-6 kHz) and duration time (25ms, 50ms and 100ms). We found that mic B (the mic near the camera on the Nexus One) cannot receive signals with frequency higher than 4 kHz, because its intended purpose is to sample ambient noise for noise cancellation. Hence we chose a signal frequency of 2-4 kHz and a duration of 25ms because long duration signals are more susceptible to error when the phones are moving. We also found that the angle and distance accuracy did not increase with longer duration signals. The selected 2-4 kHz, 25ms signal is then modulated with a pair of 20-bit pseudo random codes using Binary Phase Shift Keying. The pseudo random code is generated using the same method as in [7]. The resulting two signals exhibit low cross correlation and high auto correlation.

Position Estimation Calculation: It is critical to keep the time for computing one localization point to a minimum, in order to reduce lag. In our prototype, we have implemented two approaches to enable efficient position estimation calculation. The first approach is *phone-only*: phones exchange angle and distance information with one another as described in §3, and each phone calculates the position based on the exchanged data. The second approach is *server-*

assisted: phones send their received and timestamped audio tones to a cloud server, the server performs the position calculation, and returns the results to the clients. The benefits of this approach is that the server can perform the necessary calculations significantly faster. This is particularly true because the bulk of the position calculation time is spent on correlation calculation (see §7), which is readily parallelizable. For example, the recorded tone can be split into multiple pieces, and the correlation of each piece can then be calculated in parallel, utilizing multiple (phone or server) cores. Upon completion of calculating individual pieces, the results can be recombined in a straightforward manner to find the peak correlation. Another strategy is to recode the correlation algorithm for parallel GPU hardware already available on the phone or server. It should be noted that dedicated DSP hardware performs correlation calculations at line speeds for all variety of radio signals on the modern phone. Investigation of these options is left for future work. Overall, we believe there are no inherent barriers to lowering latencies substantially.

Android Issues: The Android OS does not currently provide an API to turn on two microphones and fetch recording data from them simultaneously. Our current implementation simulates simultaneous activation of two microphone by first turning on one microphone to record one tone, then switching to the other microphone to record the same tone sent a second time. Mic driver and operating system modifications may permit both microphones to record at the same time.

7 Evaluation

We have evaluated our two phone localization scheme, and found it is capable of the following.

- Achieves continuous 3D localization accuracy within 13.9cm position error over 90% of the time, and within 4.9cm position error over 50% of the time.
- Incorporates both TOA and power cues in a complementary manner to estimate angles and distances.
- Operates smoothly with simulated motion paths up to distances of 2m.

After detailing our methodology, we first look at microbenchmarks on individual protocol elements, before looking at macrobenchmarks on full motion paths.

7.1 Methodology

We set up our experiments in the following manner.

7.1.1 Devices

We tested on two off-the-shelf Nexus One phones. Both devices run Android operating system v2.2. The Nexus One contains the minimum set of sensors necessary for our localization scheme. These are: two microphones, one speaker, three-axis accelerometer, and three-axis digital compass. It has a single core 1GHz Qualcomm Snapdragon Scorpion processor and GPU. It is capable of communication in WiFi 802.11 a/b/g, 3G, EDGE and GPRS.

7.1.2 Metrics

We are interested in the following metrics.

Position Error. Our primary target is small position mean error and small standard deviation. Error is defined as the distance between measured position and ground truth position. Results for mean and standard deviation are conducted over 30 trials.

Figure 10. Experimental Setup

Angle Error. The overall position error is a combination of ranging (distance) error and angle error. However, several of the evaluations we present are based on angle error, rather than position error. This is because position error is more sensitive to angle error than distance error. Ranging error was extensively studied in [12] and was found to be minimal. In addition, the Kalman filter position estimate requires as input empirically measured angle standard deviations derived from both TOA and Power.

Operational Range. As the distance separating the two phones increases, distance error (and hence position error) accumulates. We also measure the position error as a function of distance.

Sync Time. Short sync time permits us to tolerate phone motion and still accurately estimate position.

Position Update Lag. Delay between tone exchange and position computation results in position update lag.

7.1.3 Experimental Setup

The test environment consisted of a 4m x 3m office room. The office was not anechoic, and therefore was subject to echo and multi-path effects. Furthermore, there was regular office furniture consisting of a bookshelf, desk and chair in the room, though nothing occluded the direct path between the two phones. The office was quiet most of the time, with only very occasional disturbances.

In order to precisely control ground truth position, we affixed each phone to a standard tripod as shown in Figure 10. We taped each phone securely to the tripod head in such a fashion as to not obstruct speaker nor microphones. The tripods were capable of angular rotation in both azimuth and

elevation, and had a height adjustment range between 100cm to 200cm. In order to precisely control ground truth angle, we implemented a helper GUI on the phone that displayed the angular orientation of the phone in real time.

For continuous localization experiments, we had the additional challenge of determining ground truth while potentially both phones were in motion. Our initial efforts with various physically constrained, motion generating gadgets proved unsuccessful. We settled on a method in which we simulated continuous motion with discrete time steps. We chose one phone to be fixed, and the other phone to be mobile. Between estimations, we manually moved the mobile phone and tripod to fixed points along the motion path. Figure 10 shows the Z motion path with the mobile phone and tripod. Establishing ground truth was subject to the error of aligning the center of the tripod with the motion path, which we estimate to be within 2cm. An alternative approach to acquire ground truth is to use vision-based tracking [2], which we leave for future work.

7.2 Microbenchmarks

7.2.1 TOA Cue Performance

We first look at TOA Cue performance independently. We fixed the position of one phone X and the distance between two phones. We then placed the other phone Y at various azimuth and elevation angles relative to X. Figure 11(a) and Figure 11(c) show the mean elevation and azimuth angle error respectively, as calculated by TOA cues alone. Similarly, Figure 11(b) and Figure 11(d) show the corresponding standard deviations. Overall, there is noticeable asymmetry in both the mean and standard deviations, across the phone, which is less surprising upon considering the asymmetrical placement of the sensors on the phone shown in Figure 3. Of note, the azimuth mean error is very low from $(azimuth = -15, elevation = -45)$ to $(-15,-15)$, and from $(15,0)$ to $(45,30)$. On the other hand, the elevation mean error is best from $(azimuth = -30, elevation = -45)$ to $(0,-15)$. Low standard deviation for both azimuth and elevation is clustered around the center $(0,0)$.

These results suggest two conclusions. First, it is important to obtain mean error and standard deviation values empirically because it is nontrivial to model them from first principal, even though our phones are ostensibly simple geometric structures amenable to modeling. Standard deviation plays an especially important role in the weighting of our Kalman filtering in Equation (7).

Second, mean error and standard deviation exhibit different trends: whereas the mean error is lowest toward the periphery, standard deviation is lowest toward $(0,0)$. This can be attributed to two factors. First, differences between ranging distances are larger toward the periphery, making it easier to estimate the angle. On the other hand, variability of the ranging distance also has a higher impact on the angle estimation toward the periphery, causing higher standard deviation at the periphery as well.

7.2.2 Power Cue Performance

Based on empirical measurements collected simultaneously with those in the preceding section, we built a mapping from power ratio to angles as described in §4.1.2. It is shown in Figure 12. It shows that for a power ratio of 0.6, the angle should be mapped to -60 degrees. We currently use a two piecewise injective functions split at 0 degrees. Therefore, in order to map power of 0.5 or below, a determination must first be made as to whether the angle is positive or negative. Above power ratio 0.5, the standard deviation on the positive angle side is too large to be of much utility, so we completely ignore the power cue in this region.

Figure 15 shows the resulting angle mean error and standard deviation when using power cues and map alone to estimate the true angle. Angles in the range $(-90,-60)$ perform extremely well in mean error and standard deviation, whereas angles in the mirror positive range perform poorly. This can be attributed to the specific phone geometry, which has one mic positioned on the lower lip of the phone, and another mic positioned on the back face of the phone (see Figure 3). Power cue's error properties actually complement TOA cue's properties nicely: Power cue's standard deviations are best past -45, whereas TOA's cue's standard deviations start to degrade at -45. The Kalman filter accounts for these empirically measured weightings.

7.2.3 Time Sync and Tone Overlap

We next evaluate the minimal sync time obtainable at a reference distance of 1.5m distance between phones. We adjusted the interval between tone exchange until the tones were just distinguishable from one another. Figure 14 shows the operation where a tone from the remote phone is received at 15ms, and a tone from the local phone is received at 50ms. Therefore, since each tone is 25ms long, we have achieved an overlap of 15ms, yet still maintained the ability to distinguish tones and power levels from one another. For shorter reference distances (and hence louder remote tones), we may obtain even tighter overlap. For longer reference distances (and hence quieter remote tones), the gap time is capped by the time to exchange both tones non-overlapped, which is 50ms.

7.2.4 Position Estimation Lag

As noted in §6, we have implemented both phone-only, and server-assisted position estimation. We tested their relative performance. Our server implementation is on a vanilla workstation PC, with a 2 GHz single core processor and 512MB of main memory.

The entire estimation procedure consists of (1) correlation calculation, (2) distance and angle calculation for TOA cues, (3) angle calculation for power cues, and (4) Kalman filter position estimation. From Figure 13, it is clear that the vast majority of the time is spent on the first step: calculating correlation between the reference signal and the audio signal recorded by each of the two microphones. The computation time of other subprocedures in position estimation are all less than 1ms. §6 mentions several available techniques for reducing correlation calculation time.

7.3 Macrobenchmarks

7.3.1 Motion Path Performance

We tested a range of different motion patterns by fixing one phone X, and moving the other phone Y according to a predefined motion pattern. We selected three basic patterns: a straight line, a staircase Z, and a sinusoidal S. Phone Y's

Elevation \ Azimuth	-45	-30	-15	0	15	30	45
-45	5.8	4.35	0.56	2.35	1.63	1.76	2.31
-30	3.36	0.91	0.05	2.19	2.68	1.06	1.96
-15	0.16	3.97	0.47	3.73	3.53	2.68	3.32
0	1.1	1.8	1.7	1.8	1.7	0.9	0.7
15	2.31	1.89	3.85	3.73	2.69	0.95	0.79
30	0.9	2.86	3.84	3.73	0.69	0.89	1.05
45	3.17	0.87	2.98	3.96	1.62	2	3.53

(a) Azimuth mean error (degrees)

Elevation \ Azimuth	-45	-30	-15	0	15	30	45
-45	8.83	6.74	7.49	4.68	6.35	6.94	7.05
-30	10.25	6.01	4.45	4.1	3.76	6.31	8.42
-15	8.31	7.2	4.83	5.79	5.68	6.75	5.51
0	4.1	3.7	5.9	4.5	3.1	6.8	3.5
15	5.85	7.18	5.53	5.39	3.1	9.32	5.29
30	9.74	8.26	3.15	4.97	6.71	8.18	10.43
45	7.87	6.92	2.26	5.05	6.32	5.16	10.61

(b) Azimuth error standard deviation (degrees)

Elevation \ Azimuth	-45	-30	-15	0	15	30	45
-45	3.94	0.32	0.95	0.22	5.14	4.32	3.5
-30	5.52	2.42	1.91	2.39	1.46	4.85	1.55
-15	3.3	3.68	0.82	1.48	3.25	1.64	3.79
0	3.22	1.02	5.15	2.33	4.65	1.02	3.77
15	4.31	4.36	2.91	3.36	3.5	2.94	5.5
30	5.62	1.79	2.53	3.86	2.97	3.6	4.78
45	3.32	0.56	5.56	2.59	3	1.51	3.07

(c) Elevation mean error (degrees)

Elevation \ Azimuth	-45	-30	-15	0	15	30	45
-45	9.84	8.28	8.05	6.07	7.77	6.32	9.82
-30	6.13	7.67	8.03	7.21	8.62	6.37	7.45
-15	5.32	8.72	5.64	4.73	3.56	5.29	6.27
0	6.42	6.69	5.05	3.69	5.71	7.26	6.03
15	4.58	4.53	5.65	3.83	3.07	6.41	6.32
30	8.58	7.8	5.31	5.1	6.74	4.45	7.79
45	6.28	8.5	7.21	5.06	6.95	5.04	9.73

(d) Elevation error standard deviation (degrees)

Figure 11. Angle estimation error with TOA cues

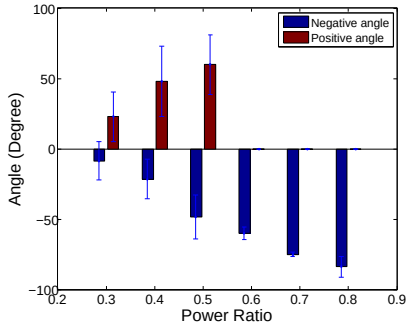


Figure 12. Power cues map to angle estimates

	Phone-Only	*Server-Assist*
Tone exchange[1] RTT	35ms	35ms
Computation		
Correlation	3200ms	85ms
TOA	< 1ms	< 1ms
Power	< 1ms	< 1ms
Kalman filtering	< 1ms	< 1ms
Communication		
Distance	Direct	Multi-hop
Data exchange	One-way	Round-trip
Data volume	44B	4KB↑/24B↓

Figure 13. Position estimation overhead

[a]Assuming phone separation of 1.5m

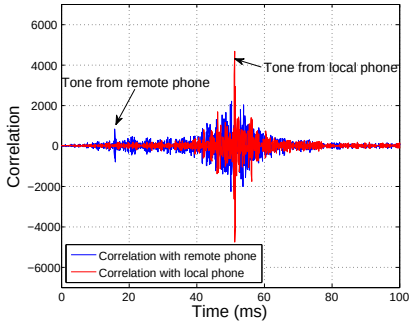


Figure 14. Distinguishing time sync overlapped signals

Angle, Ground Truth	-90	-75	-60	-45	-30	-15	0	15	30	45	60	75	90
	5	5.1	6	7.6	10.4	6.6	4	11.7	4.3	3.8	17.8	25.7	34.3

(a) Mean error (degrees)

Angle, Ground Truth	-90	-75	-60	-45	-30	-15	0	15	30	45	60	75	90
	1.44	4.23	5.98	12.52	15.73	14.1	20.75	10.79	7.69	3.82	12.94	12.21	17.55

(b) Standard deviation (degrees)

Figure 15. Angle estimation error with Power cues in 2D

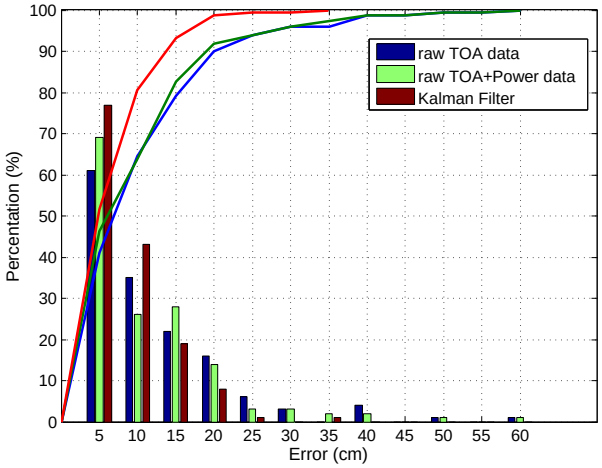


Figure 16. Position error CDF & PDF across all motion paths

motion path ranges from between 80cm to 160cm away from the phone X in the horizontal plane. We investigate three relative height positions of the phones in the vertical plane: High, Mid and Low. In High, Y was 50cm above X. In Mid, the two phones were at the same height. In Low, Y was 50cm below X. In all, we present nine distinct motion paths: one for each pattern and relative height combination. The positions are reported in terms of fixed phone X's coordinate system.

For each motion path, we compare three types of position estimates. First, we look at raw TOA-only estimates. Second, we look at raw TOA+Power estimates. Third, we look at Kalman filtered TOA+Power estimates. In addition, we

also provide an explicit break down of the measurement error versus ground truth for each motion path. These errors can be noisy because individual measurements are noisy.

Figure 16 shows the CDF and PDF for the position estimate errors across all of the tested motion paths. Position estimation with Kalman filtering achieves less than 13.9cm position error over 90% of the time, and less than 4.9cm position error over 50% of the time. Using unfiltered TOA and Power cues results in less than 19cm error 90% of the time, and less than 6.1cm error over 50% of the time. Using TOA cues alone results in less than 20cm error 90% of the time, and less than 6.9cm error over 50% of the time.

Figure 17 and Figure 18 show individual motion paths for High and Low respectively. Mid is omitted as it is substantially similar. A few points are worth highlighting. First, visual inspection indicates that the Kalman filter substantially improves upon both the raw TOA and TOA+Power cues. The error graphs verify that the average position error after Kalman filtering is 5.98cm, compared with 9.49cm by TOA cues only and 8.92cm by TOA and power cues together. Rarely does the Kalman Filter error exceed that of the other two for any point estimate. On the other hand, while TOA+Power cues are superior on average to TOA power cues alone, a minority of the time, TOA+Power cues actually perform worse than TOA power cues. This suggests that we should potentially give less weight to Power cues.

Second, localization of Line path is the best (0-5cm error); S path localization is fairly good (5-10cm error); Z paths localization is performs least well (10-15cm error). This suggests that smooth continuous motion is fairly well supported by the Kalman filter, and that sharp turns are difficult to compensate. Future work is to detect sudden acceleration changes and adapt the Kalman filter accordingly.

Third, our localization scheme works well despite changes in relative phone height: High, Mid, Low. There was no discernible relationship between the Heights. It is possible that differences would emerge had we the ability to further increase the height differential. In this regard, we were limited by our tripod's adjustment capability.

7.3.2 Operational Range

We tested the position error as a function of distance. Two phones were set up at various distances (1m, 2m and 4m) and angles $(-45, 45)$ relative to one another. For this experiment, we used a larger 6.5m x 5.5m test environment with otherwise similar properties to the first environment. Figure 19 shows that position error standard deviation is quite small at 1m and at 2m, but exhibits increased sensitivity to distance at 4m. Of note, the y-axis shows much smaller standard deviation than the x-axis in the tighter $(-15, 15)$ range because y-axis position error here is mostly a function of distance error, which is much less than than angle error.

8 Related Work

Positioning and localization are key primitives in today's mobile phone apps, and it is therefore not surprising that there exists significant research on various variants of these problems (e.g. [9]). Of most interest with regard to our work are infrastructure-based approaches for fine-grained localization, as well as other acoustic-based solutions. Finally, it is insightful to compare our proposed solution with work on human hearing.

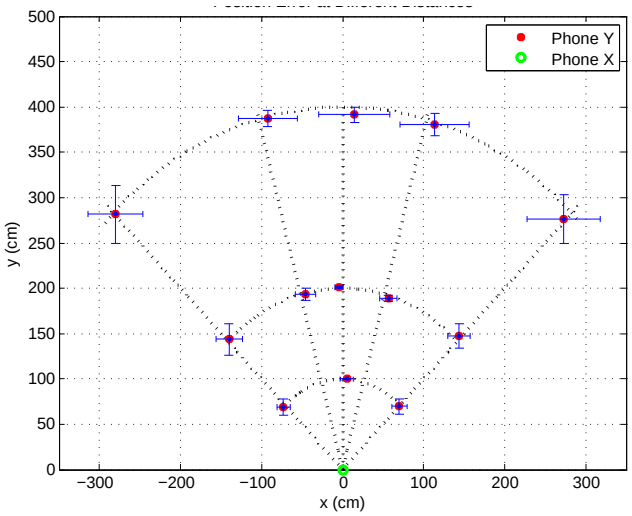


Figure 19. Position error as a function of distance

Infrastructure-based Solutions: Systems such as ActiveBat [8] and Cricket [13] are TOA-based and can achieve centimeter resolution localization, but require dense infrastructure deployment, and special end user ultrasonic devices, which renders them unsuitable for our purposes. A common technique used in these and other TOA-based localization algorithms is to take multiple distance measurements, generating a nonlinear system of equations and performing gradient descent [9]. Unfortunately, when we tested this approach, we found that it magnified any distance measurement error into considerable 3D position error. Instead, our algorithm takes advantage of the unique TOA properties of multiple mics and speakers on the phones.

Game consoles such as Kinect and Wii also perform high-precision 3D localization. Kinect projects a near-infrared mesh and employs a fixed camera with vision recognition on the near-infrared field to monitor movement of objects in front of the camera. The Wii uses a sensor bar and handheld remote to perform infrared distance measurements to determine location. Both Kinect and Wii thus assume fixed infrastructure support.

Acoustic Localization: The system more relevant to ours is BeepBeep, which performs pairwise acoustic distance measurements. We discuss it in detail in §2. Other acoustic localization techniques include ENSBox [7], which is a distributed and self-calibrating localization system for outdoor environments. It includes a custom-built hardware with each node consisting of a 4 mic array. Nodes first perform acoustic ranging, followed by bearing estimation and position estimation. Several algorithmic differences between our work and ENSBox are notable. First, ENSBox relies on node-relative time synchronization for ranging. Second, ENSBox uses its relatively more abundant mic array to estimate bearing and perform beamforming based on angle of arrival by phase shift cross-correlation. Third, ENSBox position localization relies on least squares estimation and gradient de-

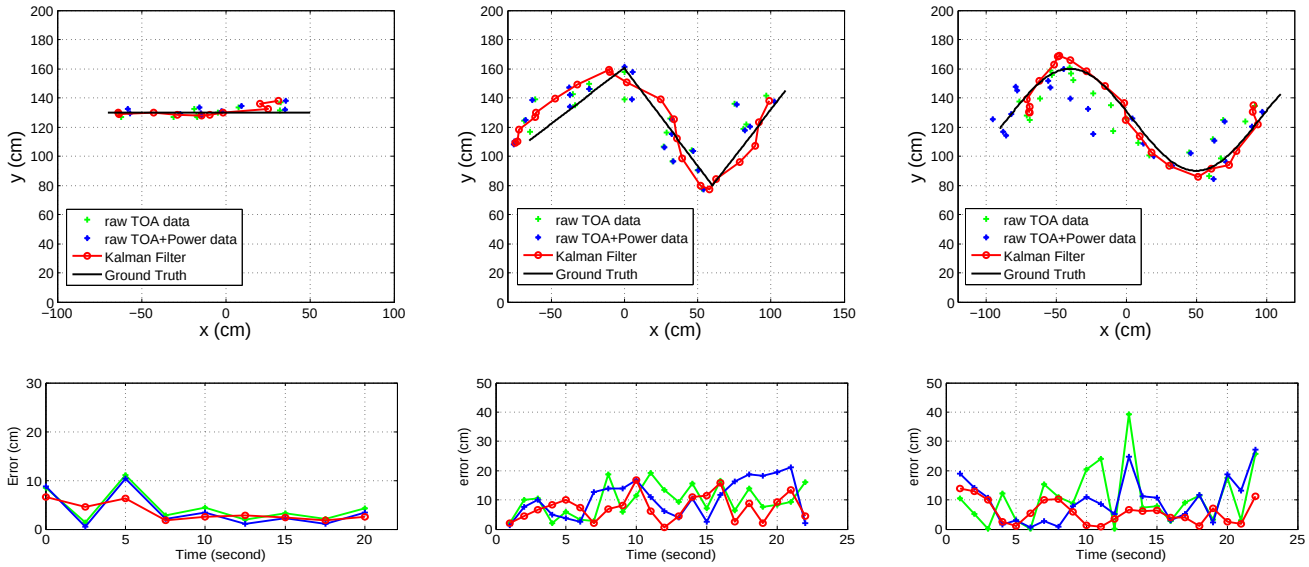


Figure 17. Raw and filtered position estimates for Line, Z and S paths at High height.

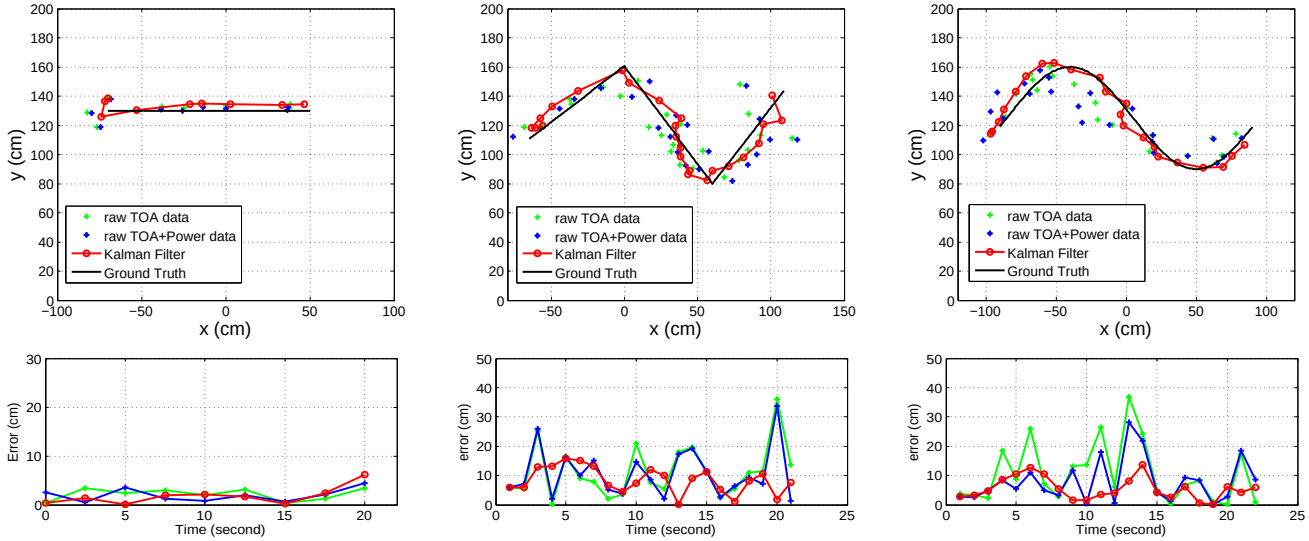


Figure 18. Raw and filtered position estimates for Line, Z and S paths at Low height.

scent, which is subject to poor initialization and higher error rates.

Other work based on acoustic localization includes [10, 14, 4]. These systems achieve resolution in meters, which is insufficient for us, and they all make strong assumptions about infrastructure support and availability of spatially distributed sensors. Similarly, there is significant work on microphone array beamforming [15], which seeks to detect the most probable location of a sound source while minimizing noise from other sources. Unlike our work, this line of research typically assumes more than two mics (often custom hardware), a passive receiver, and an exogenous sound source.

Human Hearing: Interestingly, human hearing naturally performs a form of real-time 3D localization [3]. It is therefore insightful to understand whether techniques used by the human ear could be applicable in our smartphone context. In principle, human hearing 3D localization works as follows: Given an audio source closer to the ipsilateral ear than the contralateral ear, the received signals at each ear may exhibit both amplitude and phase differences, termed the *interaural intensity difference* (IID) and *interaural time difference* (ITD) cues respectively. Human audio processing now translates these IID and ITD cues into distance ranges. However, two distances from two fixed receivers alone are insufficient to disambiguate position in 3D. This resulting ambi-

guity is termed the *cone of confusion*, and is a representation of the points which are equidistant from the ipsilateral and contralateral receivers. The key to resolving the ambiguity is the *Head-Related Transfer Function* (HRTF), a series of acoustic filters that transforms the sound due to the intricate structure of the outer ear, head and torso [6]. Intriguingly, HRTFs have proven elusive to analytical modeling [5], vary considerably per individual [16], and are typically measured empirically [1].

9 Limitations & Future Work

Impact of Externalities: Our work described in previous sections is preliminary in the sense that we have not yet evaluated fully the consequences of externalities. For example, as with all acoustic localization techniques, our system is susceptible to error from solid-body interference, and from the variable propagation speed of sound in more or less humid air. The former source of error indicates that our localization technique is not suitable for environments with many obstructions. The latter source of error may be mitigated with calibration. On the whole, we view these as acceptable tradeoffs for the ability to perform pairwise localization with commodity mobile devices. Another potential source of problems is noise. We have not yet fully evaluated the impact of background noise on the accuracy of the various cues employed by our algorithm, but we are hopeful that the algorithm should have sufficient robustness in practice. This optimism is spurred by our finding in Figure 14 which shows that overlapping tones (which is one source of noise) does not overly affect our algorithm.

More Powerful Devices: Our work is guided by currently available smartphone platforms. We believe that the key observations and principles derived in this work will hold even in next generation phones with additional sensors and more mics/speakers, however, the having additional mics/speakers or sensors may be utilized to achieve even better accuracy. For example, it would be possible to extend our algorithms to the case in which 3 mics/1 speaker, or 2mics/2 speakers (left channel, right channel) are available.

10 Conclusion

In this paper, we ask the question whether it is feasible on today's commodity phones to enable high-speed, locational, phone-to-phone games and applications, in which the two phones determine and maintain each other's relative position in real-time. We have shown that the underlying localization problem is unique in that its requirements are particularly challenging. Nevertheless, our results indicate that by exploiting and suitably combining acoustic signaling mechanisms, two microphones on each phone, and various sensor information, it is possible in principle to achieve real-time phone-to-phone 3D localization with an accuracy sufficient for many applications. However, it is clear that before the potential of HPLL applications can be fully tapped, much more work is required. We view this paper as a first step towards this goal.

11 References

[1] M. Aytekin, E. Grassi, M. Sahota, and C. F. Moss. The bat head-related transfer function reveals binaural cues for sound localization in azimuth and elevation. *J Acoust Soc Am*, 116(6):3594–605, 2004.

[2] D. Bechler, M. S. Schlosser, and K. Kroschel. System for robust 3d speaker tracking using microphone array measurements. In *Intelligent Robots and Systems*, 2004.

[3] D. R. Begault. *3-D sound for virtual reality and multimedia*. Academic Press Professional, Inc., San Diego, CA, USA, 1994.

[4] X. Bian, G. D. Abowd, and J. M. Rehg. Using sound source localization in a home environment. In *Pervasive*, 2005.

[5] J. Chen, B. D. Van Veen, and K. E. Hecox. A spatial feature extraction and regularization model for virtual auditory display. In *ICASSP'93*, 1993.

[6] G. H. Cheng, Corey I.; Wakefield. Introduction to head-related transfer functions (hrtfs): Representations of hrtfs in time, frequency, and space. In *Audio Engineering Society Convention 107*, 9 1999.

[7] L. Girod, M. Lukac, V. Trifa, and D. Estrin. A self-calibrating distributed acoustic sensing platform. In *SenSys*, 2006.

[8] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. *Wireless Networks*, 2002.

[9] A. LaMarca. *Location Systems: An Introduction to the Technology Behind Location (Synthesis Lectures on Mobile and Pervasive Computing)*. 2008.

[10] C. V. Lopes, A. Haghighat, A. Mandal, T. Givargis, and P. Baldi. Localization of off-the-shelf mobile devices using audible sound: architectures, protocols and performance assessment. *Mob. Comput. Commun. Rev.*, 2006.

[11] P. M. Morse and K. Ingard. *Theoretical Acoustics)*. Princeton University Press, 1968.

[12] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *SenSys '07*, 2007.

[13] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *MOBICOM '00*.

[14] J. Scott and B. Dragovic. Audio location: Accurate low-cost location sensing. In *Pervasive*, 2005.

[15] I. Tashev and A. Acero. Microphone array post-processor using instantaneous direction of arrival. In *IWAENC '06*, 2006.

[16] E. M. Wenzel, M. Arruda, D. J. Kistler, and F. L. Wightman. Localization using nonindividualized head-related transfer functions. *The Journal of the Acoustical Society of America*, 94(1):111–123, 1993.