Painting with Music

xxx
XXX

XXX
XXX

***Abstract*—The abstract goes here.**

I. Introduction

Accuracte indoor locazation allows creation of noval applications with surrounding awareness that uses position and movment information as input. One application is to allow users to draw with music, without physically touching the computer. Another example is to build AI games with physical pieces such as toy car racing where the computer controls some of the pieces. In this work, we aim to build such a source localization system that is portable, inexpensive, yet accuracte for localization in a small area.

Global Positioning System (GPS) is the prevailing technology used for outdoor localization. Commercial grade GPS has an average error of a few meters, depending on the size and quality of the receiver [1]. While accuracy in this range is good for many applications including driving navigation and vehicle tracking, it does not provide enough precision for local movement tracking. Ultrasound based indoor localization has achieved sub-centimeter accuracy [4]. However, ultrasound systems require use of expensive transducers.

Bluetooth and Wi-Fi based technologies gained popularity in indoor positioning recently, mainly due to the widespread deployment of bluetooth tags and Wi-Fi stations in public spaces. In these systems, signal strength received from different base stations are used for the estimation of the device location. However, their reported accuracy are in the range of 1 to 5 meters [2], [3], which is not enough for local movement tracking.

In this project, we have built a local area localization system using microphone array that localizes normal audio source. The system is built with inexpensive electret microphones. People can interact with the system using any device that has audio output such as a mobile phone.

In this paper, we discuss prior relevant research and the theory in sound localization in Section II. In Section III, we carried out a simulation to evaluate different array achitectures and their impact on accuracy. In Section IV, we present the chosen architecture along with hardware details. Finally, we provide the experiment details and results in Section V.

II. Background

Acoustic localization has been researched extensively in the literature. Localization techniques can be broadly categorized into Location Template Matching (LTM) based approaches and Time Difference of Arrival (TDOA) based approaches.

A. LTM

In LTM based approaches, acoustic templates of locations are first stored in the system during a "training" phase. Then, incoming acoustic waveform is compared with stored templates. Localization result is the location that has the best matching template. Different ways of extracting templates from raw acoustic source and different similarity measures have been investigated in the past.

[5] and [6] investigated using max value from corss-correlation as similarity measure to localize user tap on interactive surface. [7] used L2 distance in Linear Predictive Coding coeeficient space as similarity measure to localize taps on surfaces. [8] further explored accuracy improvement by using multiple templates for each location and speed improvement by merging multiple templates into one representative templates.

The requirement of having a template for each location to be detected makes this approach too restrictive for our project, since we want the drawing to be continuous in the 2D region. Moreover, the need to recalibrate all locations during setup is too curbursome for end users in a portable drawing application. Therefore, our main focus is on TDOA approaches.

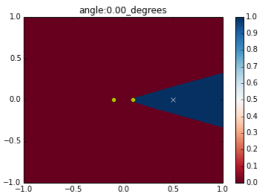
B. TDOA

Points with the same TDOA to two fixed points on a plane form a hyperbola. With more than two microphones, intersection of hyperbola curves generated by each pair of microphoens give the source location. Localization relies on accurate estimate of delay differences between microphones.

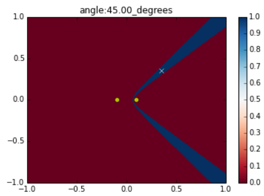
In [9], authors used eight microphnes mounted on corners of a ping pong table to localize the point where the ping pong ball hits the table. They used a threshold to determine the arrival time of acoustic signal. This approach works well in noise free environment but performance degrades with background noise. It also suffers from dispersive deflections that arrives before the main wavefront. To make it more robust, authors in [12] and [13] extracted descriptive parameters for each significant peak(e.g., peak height, width, mean arrival time). The algorithm then used extracted parameters to predict arrival time with a second order polynomial, the parameters of which were fitted during calibration at fixed locations.

Cross-correlation has been used extensively in measuring signal delays[10], [11]. Cross-correlation with different prefiterings (known as *generalized cross correlation (GCC)*) have also been investigated to improve delay estimation [14], [15], [16].

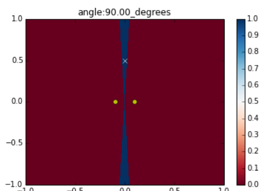
GCC provides a framework to estimate delay differences



(a) source at ($r = 50$ cm,$\theta = 0$ degrees)



(b) source at ($r = 50$ cm,$\theta = 45$ degrees)



(c) source at ($r = 50$ cm,$\theta = 90$ degrees)

Fig. 1: Uncertainty region

t_0 between two signals $x_1(t)$ and $x_2(t)$:

$$t_0 = \arg\max_{\tau} R_{x_1x_2}(\tau) \quad (1)$$

$$R_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} W(\omega)X_1(\omega)X_2^*(\omega)e^{j\omega\tau}d\omega \quad (2)$$

, where $X_1(\omega)$ and $X_2(\omega)$ are Fourier Transform of $x_1(t)$ and $x_2(t)$. $W(\omega)$ provides a way to prefilter signals passed to cross correlation estimator. We focused with three ways of prefiltering the signal:

- GCC: $W(\omega) = 1$. No prefiltering is done. This is normal cross correlation.
- GCC_PHAT: $W(\omega) = \frac{1}{|X_1(\omega)X_2^*(\omega)|}$. Each frequency is divideded by its magnitude. Only phase information contributes to delay estimation.
- GCC_PHAT_SQRT: $W(\omega) = \frac{1}{|X_1(\omega)X_2^*(\omega)|^{0.5}}$. This is somewhere between GCC and GCC_PHAT. part of magnitude information is included in delay estimation.

III. Array Architecture

As was mentioned in the previous section, points with the same TDOA to two fixed locations form a hyperbola in a 2D plane. In practical systems, we can only measure TDOA up to a precision. Therefore we look at all points with difference of distance close to some target value within measurement error ϵ. This ϵ prepresents accuracy on measurement of difference of distances, and in practice it is related to sampling rate and TDOA methods. In this section we evaluate delay estimation's impact on localization accuracy.

To see how precision affects localization accuracy, we simulated two microphones placed at: $M_1 : (x = -10$ cm, $y = 0$ cm$)$ and $M_2 : (x = 10$ cm, $y = 0$ cm$)$. A test sound source is emitted at point P which is 50 centimeters away from $(0, 0)$. Fig 1 shows the region where all points $\hat{P}$ satisfy:

$$(\hat{P}M_1 - \hat{P}M_2) - (PM_1 - PM_2) < 1 \text{ cm}$$

Intuitively, points in the region have difference of distance very similar to each other. From fig 1, the region still has the shape of a hyperbola, but with an uncertainty region around the curvce. The uncertainty region is not uniform around the curve, the farther away the point is, the larger the uncertainty region becomes. It indicates that the same delta distance movement will generate smaller difference of distance when the source is farther away from the array. The size of the uncertianty region is also angle dependent: points closer to the line of microphones have larger region compared to points close to the line perpenticular to microphones.

With more than two multiple microphones, each pair of microphones generates a hyperbolic region and localization becomes finding the intersection of hyperbolic regions. The smaller the intersection region, the better the localization accuracy. To see how accuracy changes with array placement and sound source location, three microphones are placed at three vertices of an 20 cm equilateral triangle. An audio source is placed at 20 cm away from the center of the array. Fig 2 shows the intersection of regions for 5 different placement of the sound source. It can be seen that accuracy is worse when sound source is close to the line of any two microphones. This observation is consistent with two microphone case, since points close to line of microphones have a larger uncertainty region.

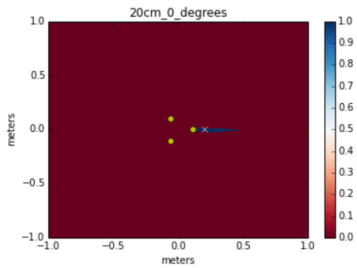
To see how sound source distance affects localization accuracy, the same simulation is carried out with sound source moved from 20 cm to 80 cm away from the center of the array. Results are presetned in fig 3. Comparing with fig 2, accuracy decreases with distance to the array. This is also consistent with our observation in 2 microphone case where source farther away would result in larger uncertainty region.

Intersection area is a measure of the localization accuracy. To evaluate an array's accuracy in a region, we can place sound source at predetermined grid points in the region and look at the intersection area for each tested point in the grid. The center location of intersection region can be used as localization estimate to calculate localization error. Results for a few different microphone array configurations are presented in fig 4.

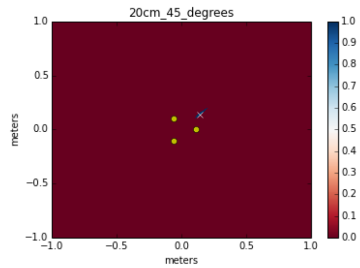
Fig 4a shows the accuracy when microphones are placed at three vertices of a 20 cm equilateral triangle. The region inside the array has good accuracy. However, for regions along line of any two microphones, the accuracy drops significantly. Average error across the region is 18.6 cm.

To evaluate how adding one microphone(without increasing array size) improves accuracy, another microphone is added to the array at $(0, 0)$. Result is presented in fig 4b. Addition of the new microphone only slightly improved the accuracy around the array region. Average error dropped from 18.6 cm to 17.1 cm. Regions near lines of microphones still have significantly larger uncertainty region.

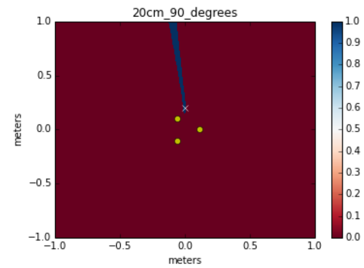
To evaluate array size's impact on accuracy, the size of original array from fig 4a is increased by a factor of 2. The



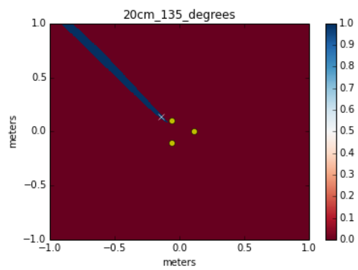
(a) 0 degrees



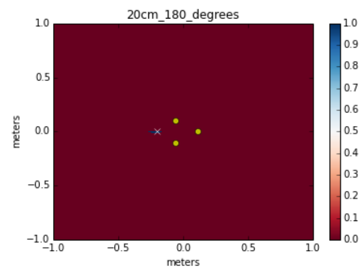
(b) 45 degrees



(c) 90 degrees

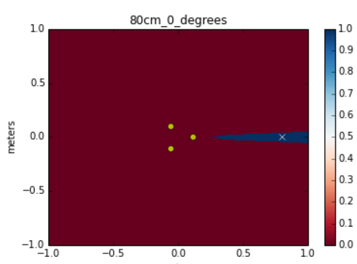


(d) 135 degrees

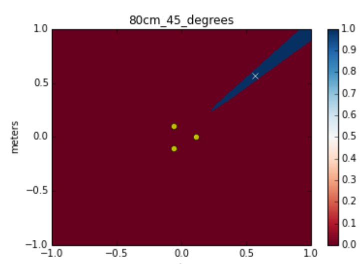


(e) 180 degrees

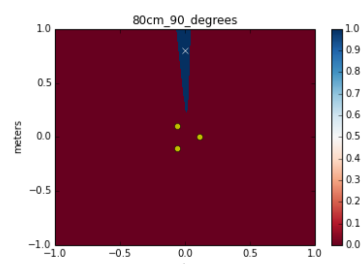
Fig. 2: 20cm equilateral triangle array. Source is 20cm away from the array



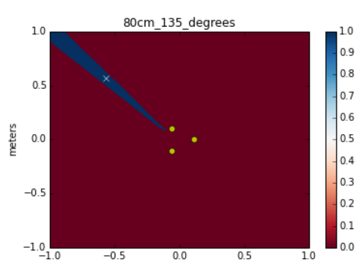
(a) 0 degrees



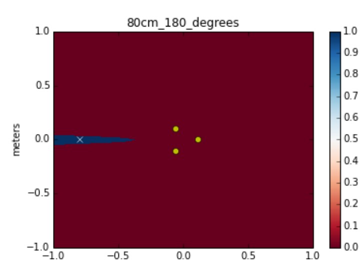
(b) 45 degrees



(c) 90 degrees

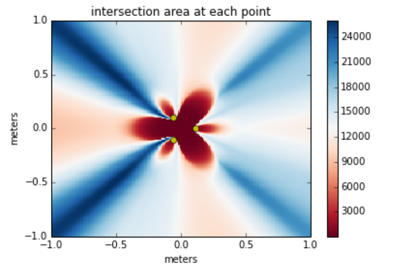


(d) 135 degrees

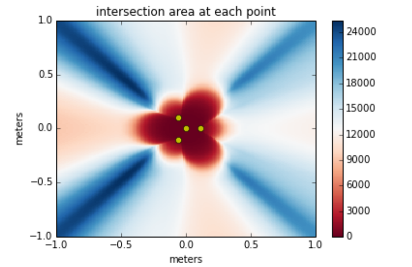


(e) 180 degrees

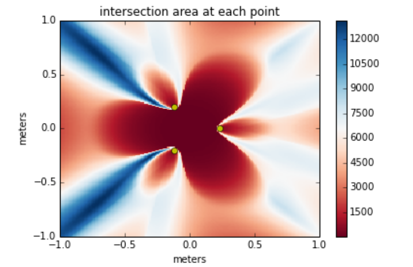
Fig. 3: 20 cm equilateral triangle array. Source is 80 cm away from the array



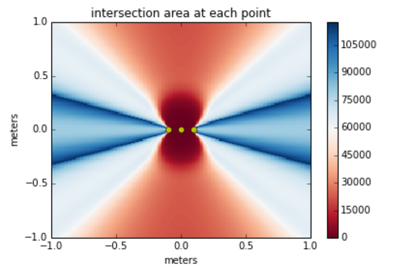
(a) 3 microphones placed at vertices of an 20cm equilateral triangle. Average error is 18.6 cm



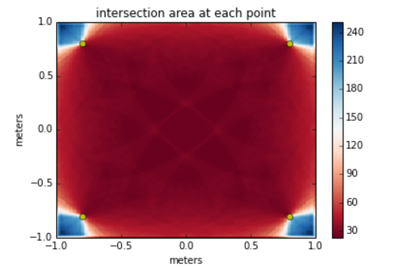
(b) another microphone added at origin. Average error is 17.1 cm



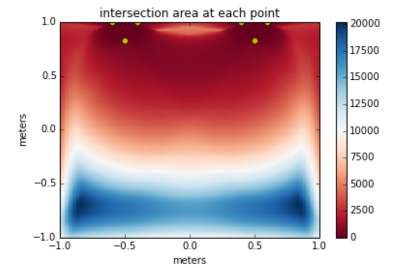
(c) 3 microphones placed at vertices of an 40cm equilateral triangle. Average error is 10.04 cm



(d) 3 microphones placed in a line. Total length is 20 cm. Average error is 55.05 cm



(e) 4 microphones placed at 4 corners of the region. Average error is 0.05 cm



(f) two 3 microphone arrays placed 1 meter apart. Average error is 2.60 cm

Fig. 4: Accuracy for different array configurations

result is presented in fig 4c. The overall uncertainty area decreased across the region. Average error improved to 10.04 cm.

In fig 4d, three microphones are placed 10 cm apart from each other on x-axis. Error heatmap showed high uncertainty on the x axis, and the overall accuracy is not as good as that with three microphones placed in a triangle. The average error is 55.05 cm.

To further increase the distance between microphones, we placed four microphones at four corners of the region. Fig 4e showed the result. With this configuration, accuracy is consistently good across the region. The average error is 0.05 cm. However, placing microphones far apart at corners of the region requires accurate placement of all four individual microphones. The system is less portable compared to small arrays with microphones near each other.

To avoid the need to accurate place four microphones at far distance(as required by fig 4e), we explored configuration with two arrays. Two 3 microphone array are placed 1 meter apart and the result is presented in fig 4f. The result indicates that this configuration has good accuracy when source is close to the arrays. Accuracy decreases as sound source moves outside the one meter by one meter region. The average error is 2.60 cm.

With simulation results, we decided to build the two array system as described in fig 4f. The setup is reasonably portable (compared to fig 4e), while at the same time having significantly better accuracy compared to one array systems.

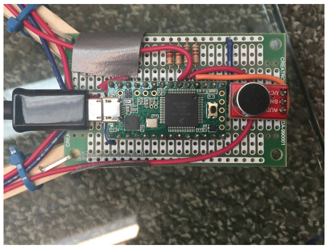
IV. System Setup

The end system has two arrays, each with three microphones mounted on vetices of a 20 cm equilateral triangle. A micro-controller is also attached to one of the vertices. Fig 5 shows a picture of the array setup. Micro-controller used in this project is *teensy 3.1*. It has 64k RAM memory and the ADC is capable of sampling at 500kHz. In this project, the micro-controller collects microphone data on all three channels for 12 millisecond and then send the recorded data to a computer through USB port for localization.

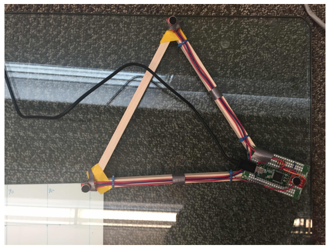
To speed up computation for real time localization, instead of searching for t_0 that maximizes equation 1, a grid search in 2D grid is performed. For each point in the grid, the theoretical TDOA to each microphone pair can be precomputed. Then localization resolves to calculating GCC for each microphone pair and performing a lookup for each point in the grid. To further improve localization accuracy, instead of using point estimate for TDOA, a liklihoood map is built. Each entry in GCC output is used as the liklihood for that delay. With three microphones m_1, m_2, m_3, there are three microphones pairs: m_1m_2, m_1m_3, m_2m_3. Theoretical TDOA from each location (x, y) to each microphone pair is precomputed and stored in $D_{m_1,m_2}(x, y)$, $D_{m_1,m_3}(x, y)$, and $D_{m_2,m_3}(x, y)$. Then the liklihood map can be built as:

$$\begin{aligned} L(x, y) &= R_{m_1,m_2}(D_{m_1,m_2}(x, y)) + R_{m_1,m_3}(D_{m_1,m_3}(x, y)) \\ &\quad + R_{m_2,m_3}(D_{m_2,m_3}(x, y)) \end{aligned}$$

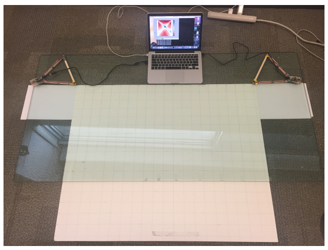
,where $R_{m_1,m_2}(\tau)$,$R_{m_1,m_3}(\tau)$, and $R_{m_2,m_3}(\tau)$ denote GCC output from microphone pairs m_1m_2, m_1m_3, and m_2m_3.



(a) micro-controller

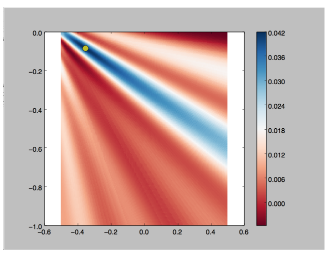


(b) array

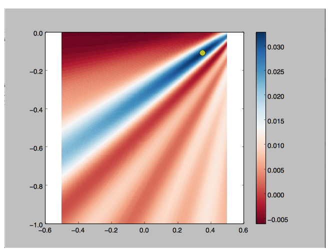


(c) two arrays

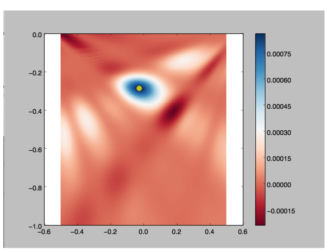
Fig. 5: Localization arrays



(a) localization with only array 1



(b) localization with only array 2



(c) localization with both arrays

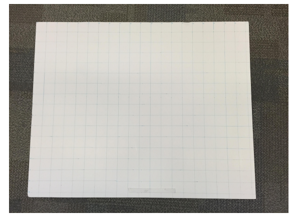
Fig. 6: Liklihood maps for localizating point at $(0.0, -0.3)$ m

Liklihood maps from two arrays can be combined into final liklihood map:

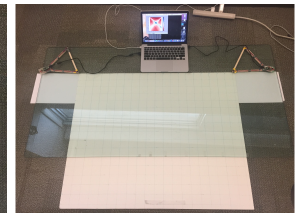
$$L(x, y) = L_1(x, y)L_2(x, y) \tag{3}$$

, where $L_1(x, y)$ and $L_2(x, y)$ represents the liklihood map from array 1 and array 2.

To see the effect of using multiple arrays, fig 6 shows the individual liklihood map from each array and also the combined liklihood map according to equation 3. Individual array gives accurate angle estimate, but has high uncertainty in distance estimate. By combining estimates from two arrays, the angle estimate can be effectively combined to estimate



(a) 1 meter by 1 meter grid



(b) array placement

Fig. 7: Setup for localization accuracy testing

distance.

From a timing point of view, the micro-controller spends 12 millisecond on sampling microphone data before sending the data to the computer for processing. Sending data through USB port takes another 18 millisecond, and processing on computer takes around 50 millisecond. Therefore, the totoal time lag between sound source and localization is around 80 millisecond.

V. Experiment

A. Setup

1) Point localization: To test localization accuracy, an one meter by one meter grid was set up and the arrays are placed at the top left and top right corners of the grid. Fig 7 shows a picture of the setup. A total of 32 positions are chosen uniformly in this region where microphone data is recorded.

2) Movement tracking: To test how well the arrays track movement, we mounted a rotating disk 40 centimeter in diameter onto the grid at $(x = 0, y = -0.3)$. Fig 11 shows a picture of the sestup. A sound source is placed on the edge of rotating disk and the arrays localizes the sound source as it rotates in a circle. In this experiment, we tested how accuracy changes with:

- different window sizes
- different audio sources
- different movement tracking filters
- different movement speeds

To test how different sound sources impact localization quality, we conducted three experiments on the same movement track with three different sound sources:

White Noise	A recording of white noise.
Music A	A music that has normal audio amplitude throughout experiment period was chosen. ”Honest Eyes” by Black Tide was the music used.
Music B	A music with intermittent low amplitude sections was chosen. ”Canon” was the music used.

To test how sound source movement speed affects localization quality, each of three experiments were conducted at two different speeds:

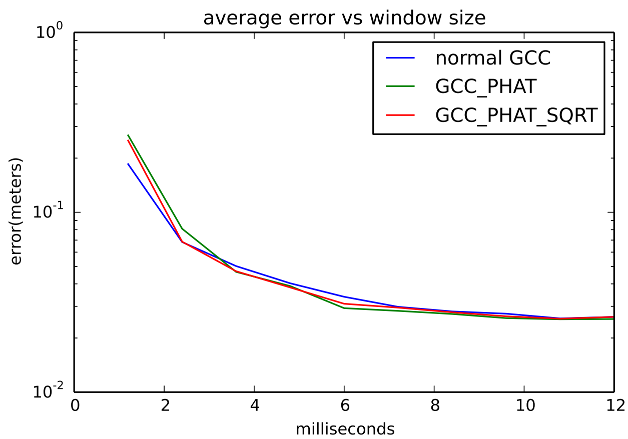


Fig. 8: accuracy versus window size

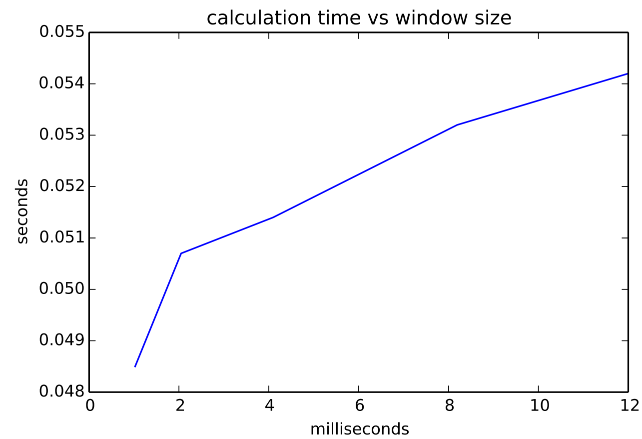


Fig. 9: speed versus window size

Normal	An angular speed of 0.5 rad/s was maintained, which translates to linear speed of 10 cm/s.
Fast	An angular speed of 1.0 rad/s was maintained, which translates to linear speed of 20 cm/s.

For each experiment conducted, two different movement filters are tests:

Averaging filter	localization for past 0.5 seconds are averaged and outputed as current estimate.
Kalman filter	A 2nd order Kalman filtering is used.

B. *Results*

1) Point localization: To test how accuracy varies with window size, the algorithm is fed with recorded microphone data with different segment length. Fig 8 shows how accuracy changes with window length for three GCC algorithms. The error lowers as window size increases and plateaus after window size exceeds around 10 milli-second. The lowest error achieved is 2.53 centimeters. It is achieved when window size is set to 12 millisecond and GCC_PHAT is used for TDOA estimation.

Although accuracy improves with window length, the calculation time also increases with window length. The part of calculation that depends on window length is using cross correlation to estimate TDOA. cross correlation can be calculated with FFT and the runtime is of order $O(N \log N)$. We measured how the computation time varies with window length and Figure 9 shows the result. The runtime increases approximately linearly in the window size region of interest.

We also calculated the localization error for each tested point in the region. Figure 10 shows a heatmap of the error distribution inside the grid. The error is below 3 cm for most areas inside the region. There is one error spike in the mid-left region and we contribute this to audio source placecment error becuase the error is fairly low and consistent around that spike region.

2) Movement tracking: Fig 13 gives an intuitive representation of how accuracy changes with window size. When window size is small(1.02 millisecond), the audio does not contain enough information to reliably estimate TDOA. The localization is noisy. As window size increases, the localization converges to the shape of ground truth circle. Fig 13 shows how the error changes with window size. The general trend is similar to that in point localization case. The error decreases as window size increases and plateaus after window size exceeds around 10 milliseconds.

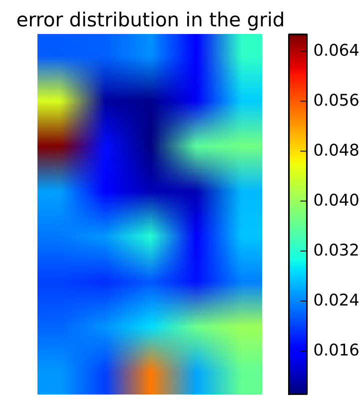


Fig. 10: error distribution in the grid

Fig 14 shows results for experiments with at normal speed, and Fig 15 shows results at fast speed. By comparing localization error for each audio source between normal movement

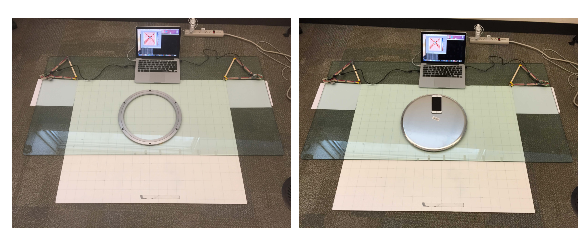


Fig. 11: Setup for circle movement localization

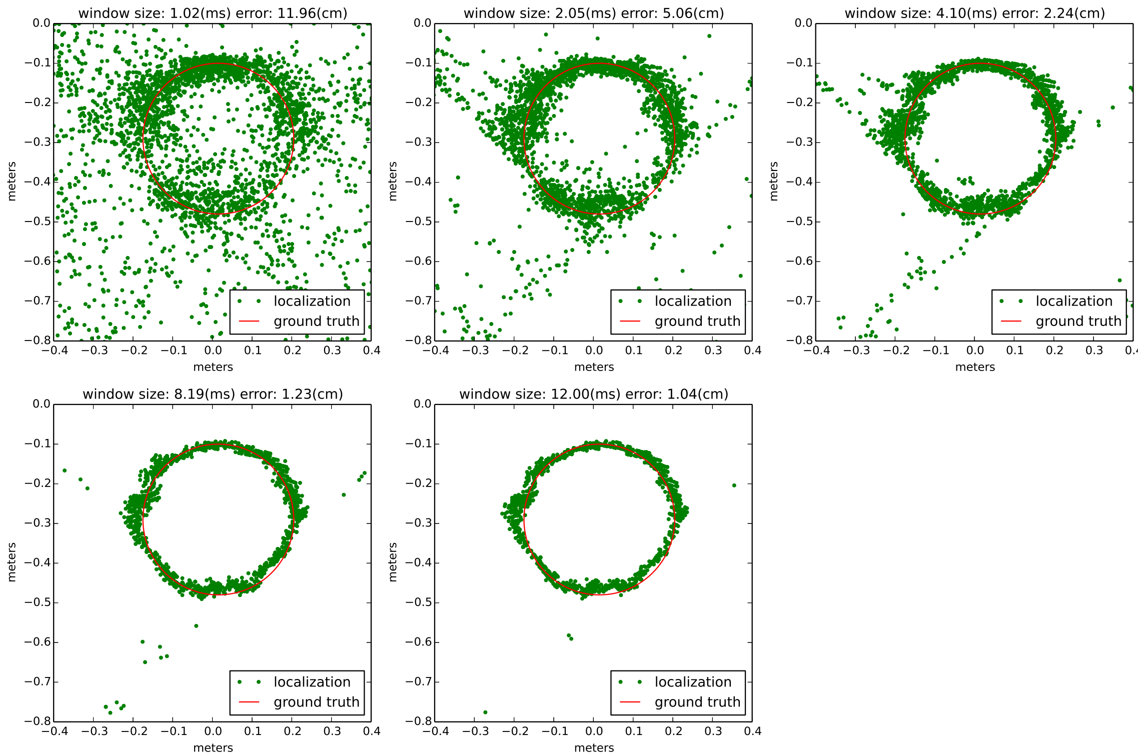


Fig. 12: Localization quality versus window size

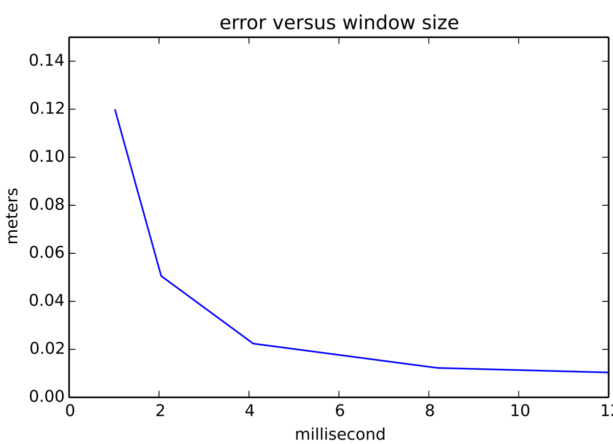


Fig. 13: Localization error versus window size

speed and fast movement speed, we find the localization error does not depend on how fast the sound sorce is moving. For example fig 14b and fig 15b shows that localization error is 1.289 cm at normal movement speed and 1.291 cm at fast movement speed.

From fig 14, localization error is 0.9 cm for white noise, 1.29 cm for Music A, and 2.9 cm for Music B. Localization accuracy is the best for white noise source, and worst for Music B. This is consistent with our expectation because low amplitude regions in Music B would cause arrays to lose track of where the source is. This can also be seen from the ”blank” regions in fig 14c.

Fig 14 also shows that raw detection has the most amount of jiggling. Kalman filter reduces the amount of jiggling from raw detection. Averaging filter has the least amount of jiggling. However, averaging filter averages detection outputs from past 0.5 seconds, which makes the filtered output lag the real movement.

VI. Conclusion

The conclusion goes here.

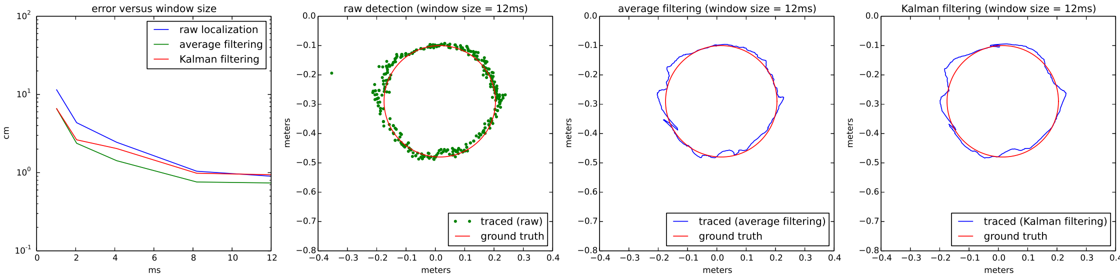
Acknowledgment

The authors would like to thank...

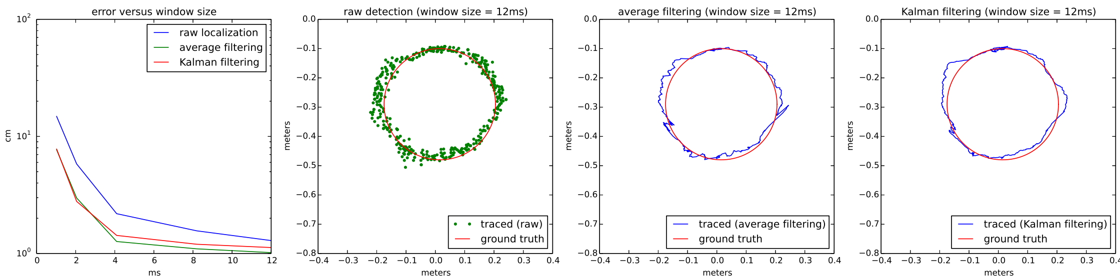
References

[1] Wing, Michael G. and Eklund, Aaron and Kellogg, Loren D. “Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability,” *Journal of Forestry*, 2005.

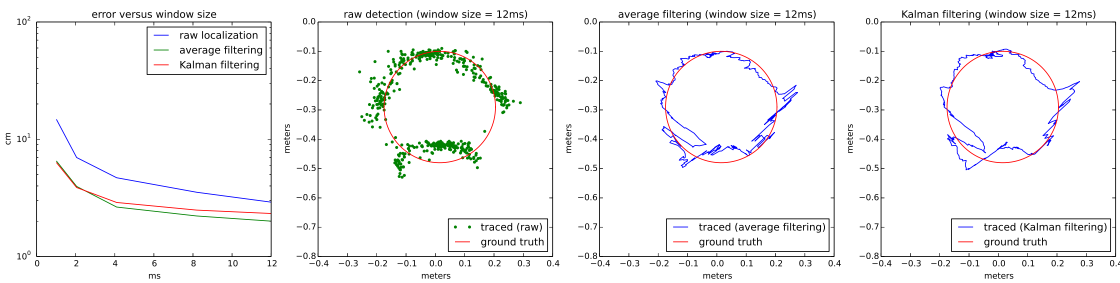
[2] Bekkelien, Anja, Michel Deriaz, and Stphane Marchand-Maillet. “Bluetooth indoor positioning,” *Master’s thesis, University of Geneva*, 2012.



(a) white noise



(b) music A



(c) music B

Fig. 14: Localization of circle movement with different sound sources. Sound source is moving at 10 cm per second

[3] Hui Liu; Darabi, H.; Banerjee, P.; Jing Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2007

[4] Yazici, A.; Yayan, U.; Yucel, H., "An ultrasonic based indoor positioning system," *2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2011

[5] Chowdhury, T., Aarabi, P., Weijian Zhou, Yuan Zhonglin, and Kai Zou, "Extended touch user interfaces," *IEEE International Conference on Multimedia and Expo (ICME)*, 2013.

[6] Pham, D. T., et al. "Tangible acoustic interface approaches," *Proceedings of IPROMS 2005 Virtual Conference*, 2005.

[7] XueXin Yap, Khong, A.W.H., and Woon-Seng Gan, "Localization of acoustic source on solids: A linear predictive coding based algorithm for location template matching," *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010

[8] Chowdhury, Tusi. "Single Microphone Tap Localization," *University of Toronto*, 2013.

[9] Ishii, Hiroshi, et al. "PingPongPlus: design of an athletic-tangible interface for computer-supported cooperative play," *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 1999.

[10] Paradiso, Joseph A., et al. "Passive acoustic sensing for tracking knocks atop large interactive displays," *Proceedings of IEEE. Vol. 1. IEEE*, 2002.

[11] Polotti, Pietro, et al. "Acoustic Localization of Tactile Interactions for the Development of Novel Tangible Interfaces," *Proc. of the 8th Int. Conference on Digital Audio Effects (DAFX-05)*, 2005.

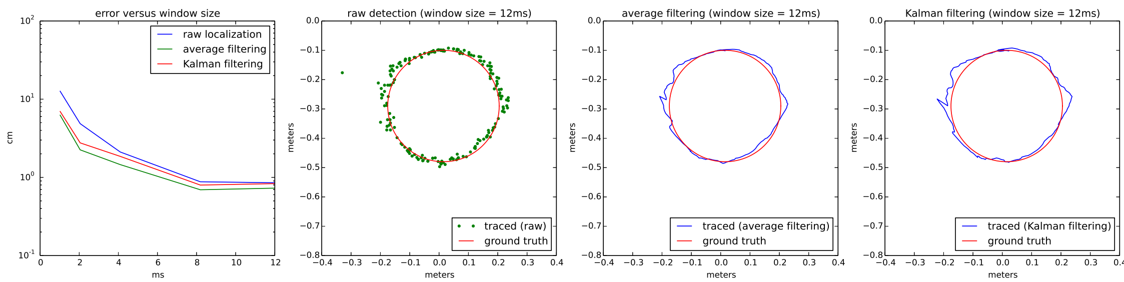
[12] Paradiso, Joseph A., et al. "Sensor systems for interactive surfaces," *IBM Systems Journal*, 2000.

[13] Checka, Nisha "A system for tracking and characterizing acoustic impacts on large interactive surfaces," *Diss. Massachusetts Institute of Technology*, 2001.

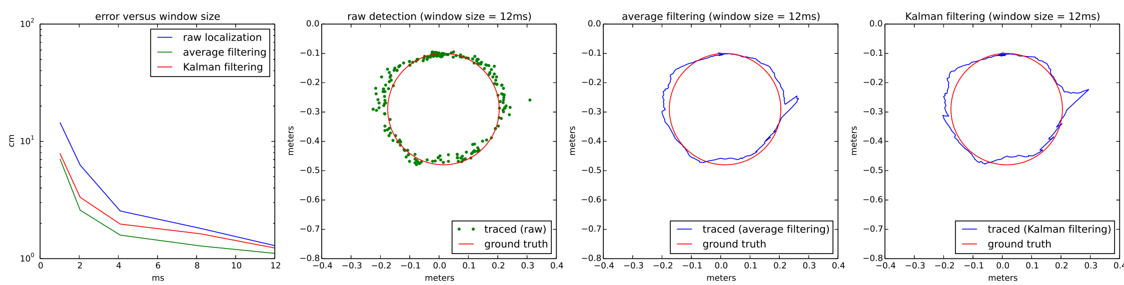
[14] Brandstein, M.S.; Silverman, H.F., "A robust method for speech signal time-delay estimation in reverberant rooms," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997.

[15] Knapp, C. and Carter, G.Clifford, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1976.

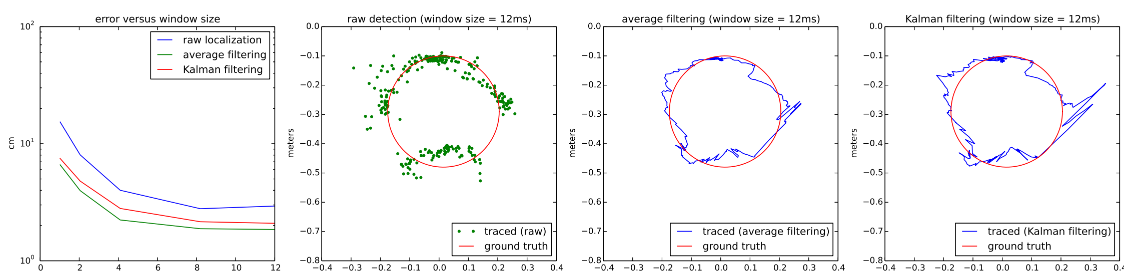
[16] Aarabi, P. and Guangji Shi, "Phase-based dual-microphone robust speech enhancement," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2004.



(a) white noise



(b) music A



(c) music B

Fig. 15: Localization of circle movement with different sound sources. Sound source is moving at 20 cm per second