

SmartPhone based Sound Direction Detection

Amit Sharma, Youngki Lee, Rajesh K Balan, Archan Misra
Singapore Management University
Singapore
{amit.2015, youngkilee, rajesh, archanm}@smu.edu.sg

ABSTRACT

Information about sound direction can be quite useful in certain situations. For example, imagine a lecture hall with video recording facility. We want to video record the speaker along with attendants asking questions, if any. Using a smartphone, we can detect direction of current speaker and accordingly automatically rotate the camera to capture the speaker. Another application could be assistance to the deaf people in a meeting. Since they rely on lip/gesture reading of the speaker to understand the ongoing discussion, so a smartphone can be used to detect the speaker direction and help the deaf person look in appropriate direction.

In this work, we present design, implementation and experiment results of a smartphone based sound direction detection system. We utilized difference in amplitude and Time Difference of Arrival (TDOA) of two signals at two microphones of the smartphone. We used only one smartphone with two microphones to detect directions from 0 to 180 degrees. Experiment results for using *white noise* as a sound source are discussed in this work.

Keywords

SmartPhone, Time-difference of Arrival (TDOA), Angle of Arrival

1. INTRODUCTION

Detecting direction of speaker can be quite useful in small gatherings e.g. meetings and seminars. Since smartphones are getting more and more pervasive day by day, so smartphone based sensing applications are quite useful. Consider a scenario wherein a deaf person is attending the meeting. He relies on lip/gesture reading of the speaker to understand what is being discussed. In order to be able to lip-read the speaker, it is important for him to know where to look for the speaker. So, real-time direction detection becomes quite critical here. If the person happen to have a smartphone, then it can be used to estimate the direction of the sound

and the person can just look at the smartphone to find out where to look for the speaker.

Another possible application could be in camera automation e.g. a video camera in a conference room can be automatically rotated to capture the speaker as well as audience if they also participate in discussion. For this, we need to install actuators on the camera and attach a smartphone with the camera. Depending on the angle detected by the Smartphone, actuators can rotate the camera accordingly.

Direction of the speaker can be detected using multiple microphones or microphone array as described in [1] [cite papers here]. But, this requires a dedicated setup of microphone hardware which is not possible to carry around all the time especially in above described scenarios.

Smartphones on the other hand are always carried by the people. Many of the smartphones have more than one microphones available on-board which potentially can be used for sound direction detection. Since these microphones are located at different positions and are separated by some distance, so the signals received at both the microphones can differ in amplitude and time of arrival depending on the position of source. This difference in amplitude and time of arrival can be used to estimate the direction of incoming sound. We have developed an Android application which leverage two microphones available on a smartphone to provide a real-time estimate of sound direction. Two microphones on the smartphone can capture two separate audio signals in stereo recording mode. We compute difference in time of arrival of these two signals at the two microphones to estimate angle of arrival of the sound source. Using signal convolution, we find lag between the two received sound sample arrays, then simply using cosine law, angle of arrival is estimated. All processing is done on smartphone itself and hence no cloud connectivity is required. Also, our method doesn't require any sort of signal processing and we don't need to store the incoming sound stream to phone memory, the analysis is done directly on the incoming data stream. Since the sound source can be either white noise or human voice etc., so in this paper, we discuss the results of our experiment with white noise as the sound source.

2. SYSTEM DESIGN

We have developed an Android application, DirectionDetector, which provides an interface for starting and stopping the detection process. The application interface displays the estimated angle and also provides a visual representation [have to add a speedometer kind of interface] of the estimated angle. Figure 1 [insert a figure number here] shows

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

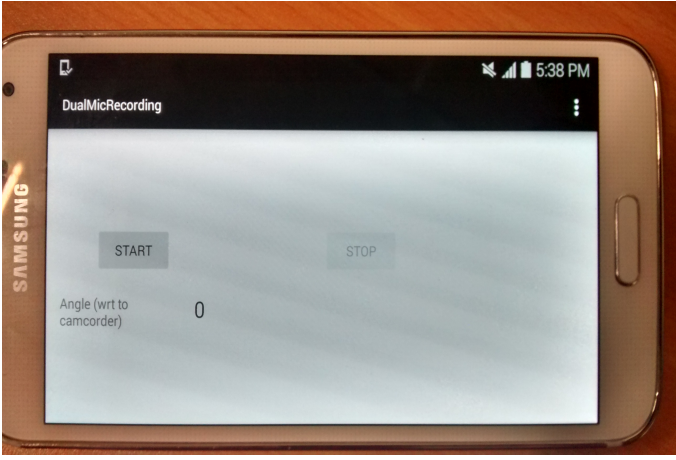


Figure 1: Android Application Screenshot

the screenshot of the application.

The flow chart in Figure ?? represents the steps of angle estimation. Once the user starts the detection process, by clicking on start button on application interface, the application start capturing sound samples from both the microphones i.e. camcorder and mic. The sampling rate can be varied from 8000 HZ to 44100 HZ. For our experiments, we used 44100 HZ because Android guarantees to support this sampling rate at all devices and also using higher frequency provides a better estimate. Each step of flow chart is explained below.

2.1 Capture & Separate Samples

When recording audio in stereo mode, Android puts captured audio frames from both the microphones in one single buffer. For processing, the application can read data from this system buffer. In our analysis we read 4096 samples from this buffer. The samples at even indexes represents a different channel than the samples at odd indexes. Before proceeding further, we split this array into two different arrays, one for camcorder data and other for the mic data. Elements of these arrays are called frames of audio data. Individual frames are processed for angle estimation.

2.2 Noise filter

Since the environment may have some noises, so before estimating angle for captured samples, we filter the possible noise. We compute root mean square value of sound magnitude. If this magnitude is lower than a certain threshold, 680 [reasons for selecting this]for our experiments, then these samples are not used for angle estimation and rather discarded. Also, some smartphones might have an inbuilt noise suppressor module, so on devices with that capability, we use noise suppressor module together with our noise filter.

2.3 TDOA estimation

The objective of this step is to find difference in time-of-arrival (TDOA) of both signals. The time difference might arise because microphones might be at different distances from the sound source. We use signal convolution for lag estimation. The idea behind this approach is that given two audio signal arrays, sum of their products will be maximum

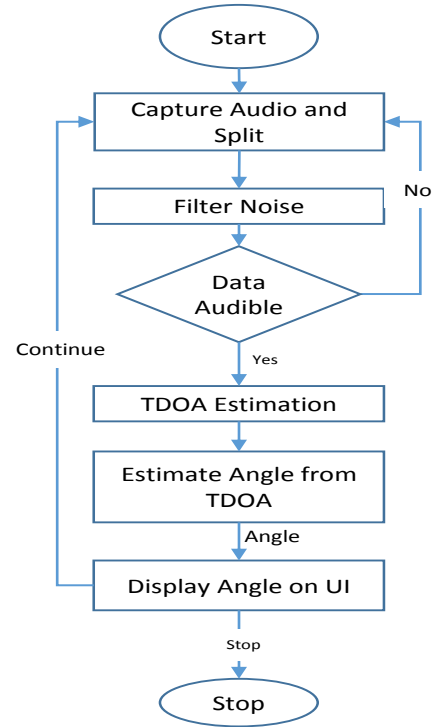


Figure 2: Flow Chart of Angle Estimation

when both signals are similar. So, given camcorder and mic data array, we shift the frames of mic array and compute sum of products of frames of both arrays. Each shift represents a lag. For example, we start by left shifting elements of mic array by N . For this shift, sum of products of elements of camcorder and mic array is computed and stored. This represents convolution value corresponding to lag equal to $-N$. This negative lag means that mic samples lag behind camcorder samples by N number of frames. Next, we compute convolution value for left shift equal to $N-1$ and store it. This process is carried on for N times for left shift followed by N times for right shift. We also compute sum of products for 0 lag. From all these convolution values for different lags, we pick the maximum value and record the lag corresponding to that value. This lag value is actually lag in terms of number of frames, so we divide it with sampling frequency to calculate TDOA which represents difference in time-of-arrival in seconds.

2.4 Compute angle from lag

Once we have TDOA, we estimate angle of arrival from it. The angle of arrival derivation looks like as shown in Figure 3. The sound source lies somewhere in first quadrant and the inclined straight lines represents sound waves coming from the source. Same technique is used for estimation when the sound source is in second quadrant as well. Here, d represents the distance between camcorder and mic of the smartphone, d_T represents difference in distance traveled by both signals and θ represents the angle of arrival with respect to camcorder of the smartphone. Since, we already know TDOA, we can d_T using velocity of sound in air (343m/s). Since position of microphones may vary device

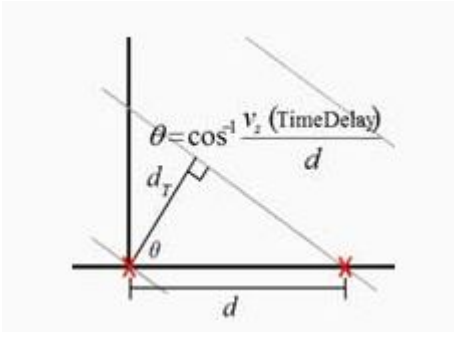


Figure 3: Angle of Arrival derivation [3]

to device, so we need to manually measure the distance between microphones. For our experiments, we used Samsung Galaxy S5 with value of d equals to 0.18 meters.

This approach can be used for angle estimation from 0 to 180 degrees only and hence the application wouldn't be able to differentiate sound sources in first and third quadrants with same TDOA values. Similarly, it is not feasible to differentiate between sources in second and fourth quadrants with same values of TDOA. We believe that if a smartphone has a third microphone (many high end smartphones have 3 microphones), then we can remove this limitation and then this same approach can be used to detect angles from 0 to 360 degrees.

2.5 Display angle on UI

Once the angle is estimated, we display that on UI thread and continue estimating for next incoming samples. Angle computation and display occurs on two separate threads. The interface also displays a pointer corresponding to the angle for easier visualization of angle.

3. EXPERIMENT SETUP

For computing the accuracy of our system, we collected audio samples from sound source at different angles (0, 30, 45, 60, 90, 120, 135, 150, 180). Multiple samples [how many?????????] were collected from each position. Then we compared the estimated angle with the ground truth giving us the accuracy of angle estimation by our system. It is interesting to study if distance between sound source and the smartphone has any impact on the accuracy of angle estimation. So, we also varied distance between the source and smartphone from 1 meter to 2 meters and 3 meters. For each value of distance, we collected multiple samples from each angle and estimated angle for each sample.

Experiments were done inside a meeting room [dimensions please?????????] with very low noise (low ambience noise). The sound used for the experiment is a white noise [cite it please ??????????]. The selection of specific music is arbitrary and angle estimation is independent of type of white noise chosen.

The smartphone used for experiments is Samsung S5 running Android v4.4.2 (kitkat) with 2 GB of RAM and quad-core 2.5 GHz of CPU clock. This smartphone has one microphone at the top and the other at the bottom.

4. RESULTS

5. FUTURE WORK

Our current approach uses only two microphones for angle estimation. If one choose two symmetric points in first and third quadrants with same value of TDOA, then the system although will determine the angle correctly but would not be able to differentiate between first and third quadrant. So future work would focus on using a smartphone with three microphones and use received sound amplitude to differentiate between sources in first-third and second-fourth quadrants. Using a third microphone, we can extend our technique to estimate angles from 0 to 360 degrees.

Secondly, the current implementation doesn't consider the height of the sound source and all the estimation assumes the sound source to be on same level as the smartphone. If smartphone and sound source are at different heights then the estimated angle might have an error. In our future work, we plan to introduce height as another parameter for angle estimation.

Lastly, we believe it would be interesting to see a comparison of accuracy between different sources of sound. Current implementation uses white noise as sound source, the future implementation would also consider other sound sources e.g. human voice.

6. REFERENCES

- [1] Y. Guo and M. Hazas. Acoustic source localization of everyday sounds using wireless sensor networks. In *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing - Adjunct*, UbiComp '10 Adjunct, pages 411–412, New York, NY, USA, 2010. ACM.
- [2] J. C. Murray, H. Erwin, and S. Wermter. Robotics sound-source localization and tracking using interaural time difference and cross-correlation. In *Proceedings of NeuroBotics Workshop*, pages 89–97, 2004.
- [3] W. University. Acoustic source localization. <http://www.ease.wustl.edu/~nehorai/josh/students.cec.wustl.edu/~jly1/angle.html>.
- [4] C. Zhang, D. Florencio, D. E. Ba, and Z. Zhang. Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings. *IEEE Transactions on Multimedia*, 10(3):538–548, April 2008.